Machine Learning for Humans: Building Models that Adapt to Behavior

HARVARD UNIVERSITY

Graduate School of Arts and Sciences

DISSERTATION ACCEPTANCE CERTIFICATE

The undersigned, appointed by the

Harvard John A. Paulson School of Engineering and Applied Sciences
have examined a dissertation entitled:

"Machine Learning for Humans: Building Models that Adapt to Behavior"

presented by: Anna Sophia Hilgard

Signature David C. Parkes
Typed name: Professor D. Parkes

Signature ______________________________
Typed name: Professor E. Glassman

Signature ______________________________
Typed name: Professor N. Rosenfeld

Signature Sendhil M
Typed name: Professor S. Mullainathan

August 9, 2021

Machine Learning for Humans: Building Models that Adapt to Behavior

A dissertation presented

by

Anna Sophia Hilgard

to

John A. Paulson School of Engineering and Applied Sciences

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Computer Science

Harvard University
Cambridge, Massachusetts
August 2021

Dissertation Advisor:
Prof. David C. Parkes

Author:
Anna Sophia Hilgard

Machine Learning for Humans: Building Models that Adapt to Behavior

Abstract

As machine learning continues to exhibit remarkable performance across a wide range of experimental tasks, there is an increasing enthusiasm to deploy these models in the real world. However, the traditional supervised learning framework optimizes performance without consideration to the use of these models by humans. In nearly all applications, human interaction affects the generation of input data, outcomes, or both. For example, a doctor may choose to either incorporate or override a machine-generated medical risk score. This judgment influences outcomes and invalidates the predicted level of performance in isolation. In the case of movie recommendation, digital records of human viewing behavior are guided by a recommendation engine, such that the distribution of input data is a function of the recommender itself. Human behavior is dynamic and responsive, and failing to account for this leads to suboptimal and even harmful results when machine learning models trained in isolation begin to interact with human stakeholders.

In this thesis, I consider humans in three different roles relative to the machine learning system: humans as model users, humans as model subjects, and humans as model auditors. For the first two configurations, I develop new frameworks that are capable of considering and adapting to relevant human behavior. For the last configuration, I reveal an important vulnerability in popular tools intended to assist human auditors. Specifically, when humans are model users, I design a new model architecture and training procedure that allows machine learning decision aids to directly adapt to how humans use them, optimizing for performance of the entire machine-human pipeline rather than solely machine accuracy. This system is validated in experiments with real human users, confirming its ability to adapt productively to different human behaviors. For humans as model subjects, I introduce a new form of model regularization that considers the motivations of to adopt new behaviors when regarding predictive models as accurate proxies for causal phenomena. This look-ahead regularizer balances model accuracy against ensuring that behavior change motivated in users results in positive outcomes with high probability. Finally, I construct an adversarial model capable of causing popular explainability tools to lead human auditors to incorrect inferences about model behavior. I show that on a variety of real world datasets, predictive models can exhibit discriminatory behavior (e.g. racial or gender disparity of outcomes) while passing proposed tests for such behavior.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Machine learning has shown incredible potential in advancing data-driven modeling and decision making. With increased access to data, flexible architectures, and greater processing power, machine learning algorithms now surpass human experts at tasks across a variety of fields. These systems can ingest large quantities of unstructured data and extract insights, distilling information that human experts would take years to read and process. Additionally, algorithms avoid human cognitive biases and prejudices, interpreting the data with the sole objective of minimizing the specified loss criterion. For these reasons, machine learning has been deployed across a wide range of fields with the goal of reducing human errors and improving performance. However, many worry that in automating decisions that traditionally rely on human judgment, important details may be lost in translation, resulting in unexpected and potentially catastrophic errors.

In law enforcement, machine learning is used to help predict the time and location of future crimes, as well as identify individuals who are more likely to commit crimes. Location-based predictive policing uses past crime data to highlight patterns of recurring crime in time and space. This allows additional resources to be preemptively assigned to those areas, potentially stopping crime before it happens [MW19]. In selected examples, this has been highly effective at reducing crime [Pea10], but with the limited data available to machine algorithms, is it possible for these predictions to incorporate the longer term implications of modified police presence as holistically as experienced human experts? On an individual level, predictive policing is used to identify individuals who are either at high risk of becoming criminals or being victims of criminal behavior [Per13]. Machine learning is also used to help judges determine bail and sentencing. Risk assessment tools predict whether defendants will commit another crime or fail to appear in court. In simulation, statistical tools can reduce the number of defendants held pretrial while also reducing the percentage of defendants who fail to appear, relative to human judges [Jun+20; Kle+18]. The algorithms can also simultaneously achieve better fairness outcomes, equalizing release rates across judges or jailing fewer minorities, compared to human judge data. Does this imply that even the most experienced human judges are unskilled at their jobs, or should we be concerned that the algorithms have missed some other important criterion of human judges?

In medicine, machine learning models are used to build risk models, perform diagnostic imaging, and surface relevant data across patient populations. Neural networks have shown such success in identifying disease in medical images that several high profile computer scientists have questioned whether radiology has a future as a medical specialty [The]. Models can successfully identify and predict the progression of diabetic retinopathy [Nie+19; Arc+19] and identify melanomas and other skin cancers as well as experts [Est+17; Tsc+19]. In radiology, deep learning matches or outperforms radiologists in a number of tasks [Liu+19b], including predicting breast cancer risk [Yal+19] and analyzing chest x-rays for pneumonia [Raj+17]. Patients may wonder, though: do these algorithms rely on the same indicators that human doctors use? Are they robust enough to correctly identify unusual cases or to flag conditions other than those which they are explicitly trained to detect, as a human physician would? Machine learning has been used to generate both standalone risk scores that are more predictive than the prevailing human-developed risk scores [Spa+19] and interpretable risk scores to aid physicians in understanding the interactions between known risk factors where no models previously existed[Str+17]. Advances in content-based image retrieval allow practitioners to search by concept as well as image, quickly identifying labeled examples that relate to a query image (in, for example, tissue biopsies) [Cai+19]. Are these tools proven to improve patient outcomes, or might they interfere with a doctor's carefully refined intuitions?

In financial technology, machine learning is used to generate credit scores and evaluate insurance applications and claims. Algorithms use ‘digital footprint’ information, not typically available to credit bureaus, to help determine if applicants qualify for a loan or credit card [Ber+20]. Companies argue that this practice can reduce disparities in lending by making credit more available to minorities who do not have traditional credit scores, while not increasing the risk of charge-offs to lenders [Kah20]. Insurance companies use machine learning to make underwriting decisions, determining whether to approve or deny applicants [Adv], and to identify claims patterns that are likely to be fraudulent [Mel18]. In both applications, these processes allow the companies to pass difficult cases to human agents while rapidly approving low risk applicants and claims with a high probability of legitimacy.

Machine learning algorithms permeate the more quotidian aspects of our lives as well, helping to determine the news we read [LDP10], the routes we take to work, the items we purchase [LSY03], and even our romantic partners [Tif19].

In all of these applications, the data of interest are generated by humans and often concern human activity. Thus, where humans have a significant advantage over machine learning is in understanding the processes that generate the data in the first place. Machine models are by comparison extremely narrow in scope: the provided data is evaluated without consideration for any biases and errors that may have gone into its creation, and the loss criterion is optimized absent constraints that comparable human decision-makers would consider.

In particular, in high-stakes domains, in which predictions may have a lasting impact on people’s lives, there is increasing awareness that accuracy is often not appropriate as a singular objective. Researchers have raised concerns in each of the aforementioned domains. Predictive policing is widely criticized for perpetuating biased police practices that generate the training data [RSC19; Ens+18]. Because police data only reflect crimes that were reported to or observed by the police, crime data will be biased toward the areas where areas where the police were to begin with. In this way, human biases are encoded in the data: if police were more likely to target a neighborhood before the system’s existence, they were more likely to identify crime there. Algorithms that are not carefully engineered to consider this bias will identify the pattern of higher crime without taking into account the higher surveillance, and will thus suggest sending even more police presence to these areas. This can be expected to further influence future training data in a reinforcing feedback loop.

Risk assessment tools, despite promising less incarceration and more objective outcomes, have been criticized as inaccurate and biased against minority groups [Ang+16b]. Even when race is not used as an input to models, as in the widely analyzed Northpointe algorithm, other variables which correlate with race can be unintentionally used as proxies [BS16]. For example, when neighborhoods tend to have different racial concentrations, it is often possible for zip code to act as a proxy for race.

In medicine, several examples have identified machine learning algorithms with apparently high diagnostic accuracy that rely on correlations related to care decisions made by the human medical team, which differ by severity of condition, rather than causal features in the data. Zech et al. [Zec+18] find that CNNs trained to identify pneumonia in chest x-rays often make use of features of the image to identify the hospital and department from which the x-ray was obtained. Because hospitals and department (for example, specialty vs general, emergency vs outpatient) vary in the prevalence and severity of cases they handle, knowing this information alone can be predictive. However, using these features results in classifiers which rely on non-causal features and therefore have poor performance out of distribution or under distribution shift. Caruana et al. [Car+15] describe the application of machine learning models to a pneumonia dataset, predicting probability of death, and show that in this application, the presence of asthma is negatively associated with probability of death. This is highly counterintuitive, and can be attributed to the fact that individuals with asthma receive much more aggressive treatment compared to individuals without asthma because, in fact, their risks are higher. However, again, without this human knowledge, machine learning is likely to pick up on correlations which could cause the classifier to be inaccurate or even dangerous in deployment.

Patterns in data and the use of predictive machine learning have also led to concerns in the realm of algorithmic lending. Bartlett et al. [Bar+19] find that while fintech mortgage brokers exhibit much less discriminatory pricing relative to traditional lenders, some price disparity persists between Hispanic/African-American and White borrowers, with minority borrowers expected to pay slightly more than comparable white borrowers. The authors hypothesize that this may be due to tendencies of minority groups to both solicit fewer competing mortgage offers and to live in specific neighborhoods. Thus, even without an intention to discriminate, profit-maximizing algorithms in this case learn to extract higher rents in minority neighborhoods.

The above examples raise concerns related to fairness, safety, and accountability. These concepts can be difficult to mathematically define or may lack supporting data, making it challenging to develop models that satisfy these objectives. Human experts, on the other hand, can readily consider the sociological structures that generated the data. This can yield a highly informative prior concerning which features are likely to be relevant, contain errors, or relate to outcomes only through confounders. Additionally, understanding the source of the data can allow human experts to understand its limits: identifying out of distribution data or subpopulations where performance may not be guaranteed. These competing advantages, of accuracy and objectivity for machines and external perspective for humans, should allow for a highly productive collaboration between humans and ML. Indeed, much research has been dedicated to developing systems that allow humans to harness the advantages of machine learning while maintaining oversight with respect to properties outside of the machine's consideration. Unfortunately, many of these solutions are themselves limited due to the restrictive assumptions they place on humans within the partnership. In the following section, I categorize some existing approaches to human-ML partnerships by the role humans play and highlight empirical weaknesses and theoretical gaps.

1.1 Human Roles in ML Collaboration

1.1.1 Humans as Auditors

Methods in this class train machine models in isolation to maximize an objective, generally accuracy, then pass optimized models to a human-decision maker who issues the final prediction. Introducing machine learning algorithms but allowing incumbent human decision-makers to audit the predictions, at either a model or individual prediction level, seems at first a natural approach to ensure strict improvements. This allows for humans to accept high-accuracy machine predictions when the models meet external criteria (e.g. safety, fairness, causal validity) to the

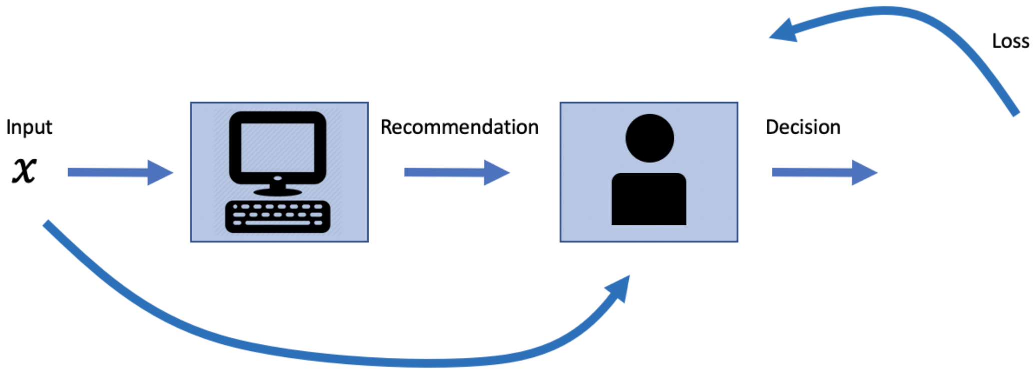


Figure 1.1: *Machine models pass a prediction to human decision-makers, who decide whether to accept or modify the prediction. The machine receives no feedback from the human or the pipeline loss.*

standard of human judgment and otherwise (1) default to the original human recommendation or (2) appropriately modify machine predictions, allowing recommendations to be informed by both machine and human advantages. Methods in this class focus on providing human experts with tools to effectively audit and incorporate machine predictions.

Interpretable and Explainable Machine Learning A first set of techniques are interpretable and explainable machine learning. *Interpretable machine learning*, in which a model is constrained to belong to a class of functions simple enough that humans can follow the decision function exactly from data input to prediction output, has been proposed as a widely-applicable solution for human inspection of machine recommendations [Lip18; DVK17]. *Explainable machine learning* attempts to do the same, but with proxy "explanations" of models: simple models fitted to the output of complicated models, often in the locality of a given prediction. These models often take the form of feature importance scores (often sparse, as in sparse linear classifiers) [Tib96; UR16; RSG16; LL17; Smi+17], decision trees or decision lists [Ang+17; Lin+20; LBL16], or prototype-based models [KRS14; Che+18].

Even assuming appropriate human usage, these approaches have a number of limitations. Interpretable approaches require simple models, which are necessarily less flexible than black-box models such as deep networks and random forests. Further, they often require that the input features are themselves interpretable; even if highly accurate, a decision tree that considers individual pixel values is unlikely to yield an intuitive human understanding of the model. Advocates for interpretable models have suggested that it is more common than not that interpretable models exist with comparable predictive power to black box models [SRP19], but finding these models often requires time-consuming feature engineering and fundamentally hard constrained optimization relative to deep learning methods [Rud19].

Explanation methods suffer from the obvious setback that the simpler explanations do not fully represent the more complicated model they seek to explain [Rud19]. Further, recent work has also shown that there may exist many differing explanations of approximately equal fidelity. This multiplicity allows users to draw a range of conclusions about the same model, including that a model is more fair than it is in actuality [Aïv+19; LB20].

When interpretable models are provided to humans, still more complications arise. While studies have shown that interpretable models and explanations can improve performance on specific tasks relative to a non-interpretable benchmark [LBL16; KRS14; RSG16], it is less clear that they can generally resolve the difficulty in human incorporation or evaluation of machine predictions arises from an inability to reason about the machine's decisions. Poursabzi-Sangdeh et al. [PS+18] find that providing interpretable models hindered study participants' ability to identify and correct model errors relative to a black box benchmark and provided no additional benefit in encouraging participants to use predictions when it would have improved performance. Lage et al. [Lag+19] find inconsistent effects of increasing numerical proxies for interpretability on response time and accuracy. Kaur et al. [Kau+20] conduct a user study of trained data scientists and find that interpretability methods are ineffective in helping them identify many common model problems.

Explanations have also been shown to increase the probability that humans yield to a machine recommendation, even when the explanation is nonsensical [LT19] or the prediction is incorrect [Ban+20]. Additional concerns arise when considering that explainability tools often require careful tuning of hyperparameters to yield sound insights [Zha+19]. When deployed under default settings, the explanations may be silently incongruous with the task the user has in mind, and can therefore backfire, increasing confidence in incorrect suppositions about the model internals.

Post-hoc fairness metrics A second set of techniques are post-hoc fairness metrics that allow human users to more effectively audit machine models and predictions for violations of specific fairness definitions. This is particularly relevant for black box models, in which it is impossible to know whether protected features such as age, gender, or race are being used in prediction, but can also be relevant when protected features correlate with non-protected features. In these cases, it is often possible for a model to achieve comparable outputs to a biased model while using only proxy features for protected features. Adebayo et al. [Ade+16] creates a toolkit for auditing models with respect to the relative contribution of input variables to the model output. Unlike many explainability methods for variable importance, it does not require access to the black box model. Bellamy et al. [Bel+18] implement a suite of output fairness metrics for comparing model outputs across different definitions of fairness, of which there are many [GP17]. Unfortunately, it has been shown that it is impossible to simultaneously satisfy many of these fairness definitions [KMR16; FSV16], and the appropriate definition is often application-specific and subjective [Sax+19]. A public debate between Propublica [Ang+16b] and Northpointe [DMB16] regarding whether the latter's risk assessment software exhibits racial bias reveals that it is possible for an algorithm to be fully vetted with respect to fairness by one party and violate the fairness perceptions of another. Without a more unified framework in which the algorithmic fairness goal is aligned with the goal of the human decision-maker, such audits may result in inconsistent outcomes.

More generally, all systems in which computers provide recommendations to human users suffer from a number of obstacles related to broad human biases. Research consistently shows that human decision-makers follow algorithmic recommendations less frequently than would be optimal [DSM15; YWVW19; PS+18; LT19]. This phenomenon is not well understood, but has been at least partially attributed to error intolerance [DSM15], a desire to retain agency [DSM18], and expertise bias (the propensity of experts to overestimate their performance) [Log17]. Recent work proposes that at least some of these factors relate to a decreasing sensitivity to forecasting error: that is, the more inherent uncertainty exists in a prediction problem, the less people believe that a machine would be capable of achieving high accuracy [DB20]. This results in humans preferring their own more flexible predictions more often than is ideal.

Further, when humans have the opportunity to reject or modify machine outputs, they may (intentionally or unintentionally) do so inconsistently across groups due to personal biases, potentially increasing rather than correcting for any unfairness present in the machine predictions. In particular, research has shown that risk scores may exacerbate disparate outcomes for minority groups in the hands of human users, as judges use the scores selectively to justify harsh punishments for some groups while exercising the ability to overturn unfavorable predictions for others [SD19; GC19a].

For all of these reasons, incorporating humans as auditors, without feedback to the machine learning system, may result in reduced accuracy or unintended consequences resulting from human interpretation and biases.

1.1.2 Humans as Fallback

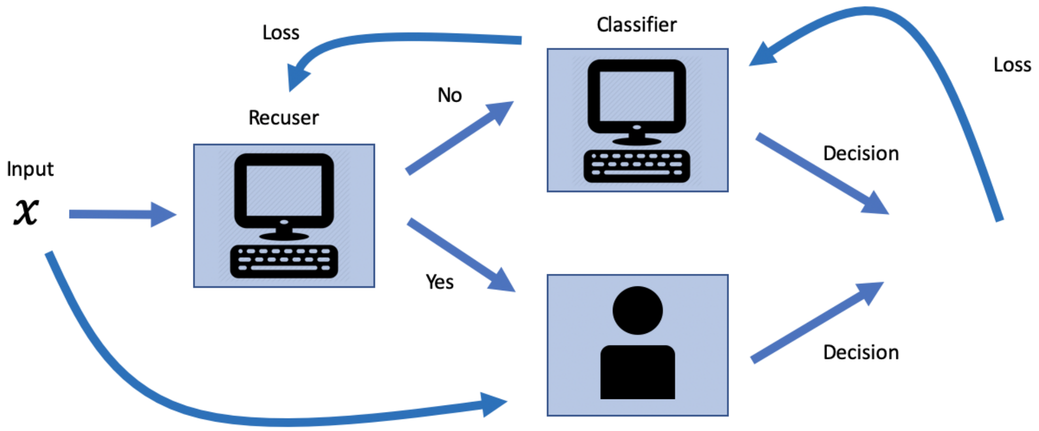


Figure 1.2: *Machine models decide whether or not to recuse themselves from decision-making. When machines do choose to predict, humans do not participate in the process*

Methods in this class accept that human decision-makers are flawed and time-constrained, and that in many cases allowing humans to make the final decision on all predictions is both inefficient and likely to result in worse outcomes with respect to accuracy and fairness. However, because humans may have higher accuracy on specific examples, for example due to access to side information or unavailability of training examples for the machine algorithm, they allow for the machine model to choose not to issue a prediction, instead passing the prediction problem to a human expert.

Learning to Defer A particular technique in this "humans as fallback" class is composed of two separate classifiers: one that learns to minimize loss for the classification problem and one that learns on which examples to recuse itself [CDM16]. The earliest works of this kind assumed a fixed machine classifier which minimizes loss over the entire data distribution [MPZ18], as well as fixed human performance (here, simulated by a fixed neural network). This setup does not allow for the classifier to adapt at all to the human expert; in situations where the classifier has limited flexibility, this may decrease performance as the pipeline may be better off learning to predict well on only those examples which will not be deferred to the human decision-maker. Later works incorporate this flexibility, training the recuser and classifier jointly with access to information on human decision-maker performance [WHK20; MS20]. An alternate extension keeps the fixed machine classifier but allows for human performance to vary with the number of examples deferred, as the human decision-maker can dedicate more resources to a fewer number of cases [Rag+19].

However, all of these models are still limited in that either the human decision-maker or the machine classifier is solely responsible for making the end decision. (While Wilder, Horvitz, and Kamar [WHK20] theoretically allow for the machine classifier to vary predictions based on input from a human, in practice deferred decisions are always equal to the human output.) The human decision-maker does not receive assistance from the machine model. Moreover, the machine model does not incorporate human feedback.

Further, in many domains humans are unlikely to allow for machines to make final predictions without human approval, especially in domains in which the available data is known to be insufficient to fully model the phenomenon of interest. For example, Neufeld [Neu17] gives suggestions for public officials to audit and adapt recidivism prediction algorithms to the goals of individual jurisdictions before deployment, and Chan and Siegel [CS19] discusses a robust future of human oversight in radiology, despite AI superiority in specific tasks [Kil20].

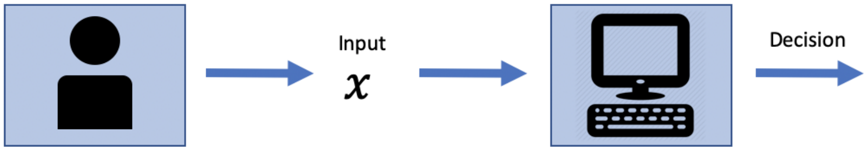


Figure 1.3: *Humans generate the data on which computer algorithms are trained, but the machine does not consider the possibility of feedback from decisions to humans.*

1.1.3 Humans as Static Data Distribution

Until fairly recently, applications of machine learning to data generated by humans have generally focused on modeling the learning environment as static. However, when machine models recommend or incentivize specific behaviors, the deployment of a model can create feedback loops. In recommendation systems, commonly these are positive feedback loops, where the outputs reinforce the conclusions of the model. For example, popular items are generally more likely to be recommended than unpopular items, and this greater exposure tends over time to exacerbate this relationship [Man+20]. While this appears to make the model better (more accurate) in the short run, in the long run it can lead to low system welfare, reducing diversity and fairness among recommended items and disincentivizing new content creation [Mla+20].

In other applications, incentives can shift the data distribution such that models are less accurate in deployment than in training. This relates to the observation commonly known as Goodhart's law, that "when a measure becomes a target, it ceases to become a good measure". One example of this is domains in which it is possible for users to "game" the system, changing superficial features to improve their predictive outcomes without affecting the latent quality the model attempts to measure; in these scenarios, negatively labeled points can with relatively low cost move into positive regions of the classifier, creating a data distribution shift that decreases the efficacy of the classifier. Particular concerns have been raised about this behavior in the context of credit scoring and college admissions. Internet finance sites commonly recommend strategies for credit card usage which manipulate the factors of the FICO credit score [Har21] without changing the inherent creditworthiness of the individual. The recent college admissions scandal highlighted the many ways wealthy parents attempt to increase their children's prospects outside of improving their academic abilities [Boa19].

Alternatively, models may disincentivize investment in positive causal behaviors for certain groups when the groups are heterogeneous in their features and only a single classifier is deployed. Liu et al. [Liu+20] cite the example of college admissions, in which appropriate SAT score cutoffs may vary by group. If the college sets a single cutoff and places a high cost on false positives, the group with lower scores may be discouraged from investing in academic qualifications. Had this not been the case, the college may have observed these points as false negatives and been encouraged to lower the SAT score cutoff to capture more qualified students. When the students do not invest, however, the model appears to have classified these students correctly.

Furthermore, interpretable models may affect human choices even in situations where the model itself does not directly determine users' outcomes. Interpretable models can be simulated by human users, thus users are able to imagine how different outcomes might be possible with altered features. This may encourage users to improperly understand predictive interpretable models prescriptively. For example, many nutritional recommendations are based on epidemiological studies (due largely to the difficulty of rigorously controlling study participants' diets for any extended period of time). This has often led to population-level correlations being interpreted causally, for example leading individuals to reduce dietary cholesterol [CI04]. In fact, the causal link between dietary cholesterol and heart disease is weak at best [Kra05], and there is evidence that dietary shifts away from cholesterol-containing whole foods such as eggs resulted in higher consumption of processed grains, which have themselves been associated with higher levels of heart disease [Yu+13].

These examples emphasize that when subjects are liable to alter their behaviors based on model predictions, it is important that models encourage feature changes which are safe and beneficial.

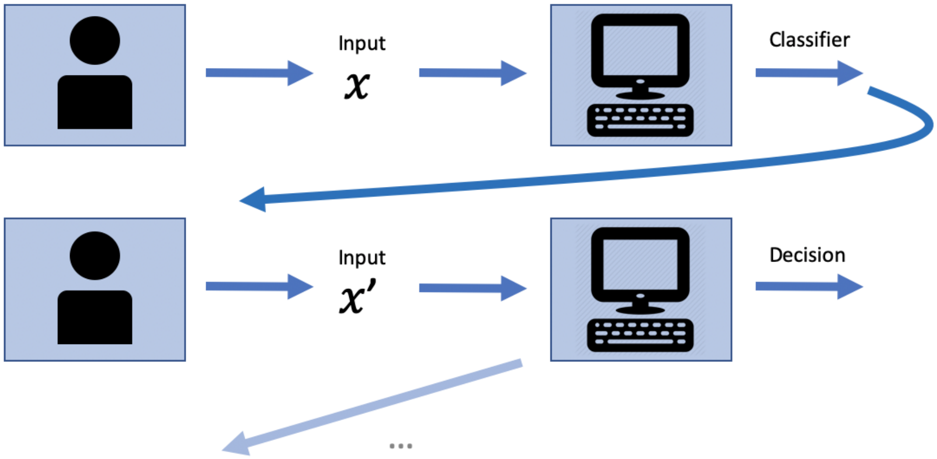


Figure 1.4: *The classifier is trained on an initial dataset to maximize predictive accuracy. As deployed classifiers alter human behavior, the data distribution shifts, and the classifier must be retrained to maintain high predictive accuracy.*

1.1.4 Humans as Adversaries

In response to the above examples in which the subjects of classification algorithms are able to change their input features, labels, or both, several methods have emerged which frame classification as a game between the classifier and the subjects of the classifier. In *strategic classification*, the classification problem is formed as a Stackelberg game, in which the classifier has the goal of achieving maximum accuracy but must commit to a classifier before observing the realized feature changes of the model subjects [Har+16; BS11]. Thus the goal of the classifier is to anticipate feature changes, ensuring that the classifier is still maximally accurate after subjects perform strategic modifications. *Performative prediction* explores a similar setup but allows for repeated retraining and studies the conditions under which alternating steps by classifier and classified can be expected to converge to a "performatively stable" equilibrium: that is, a model which achieves optimal accuracy for the behavior it induces [Per+20].

A number of works have brought to light the ways in which strategic classification may increase inequality among classified groups. For example, when the disadvantaged group has a higher cost of modifying features relative to the advantaged population, the equilibrium solution found using a strategic classification framework will tend to create errors that benefit the advantaged population and hurt the disadvantaged population [HIV19]. It has additionally been shown that strategic classification results in a burden to subjects with positive true labels (as positive individuals who wish to be classified positively must incur costs to optimally change their features to meet the classifier's expectations). Furthermore, this burden falls more heavily on disadvantaged groups when they either tend to have lower feature values for the same outcome or experience higher costs of modifying features [Mil+19].

Furthermore, strategic classification and performative prediction do not consider the quality of the equilibrium outcome for members of the classified population. In strategic classification, it is assumed that all feature changes are superficial and therefore the distribution of true outcomes is unaffected by any modifications. Performative prediction, however, allows for modifications to both input features and outcomes, and thus we should be concerned whether or not the equilibrium outcomes are safe and preferable to the original set of outcomes.

1.1.5 Humans as Collaborators

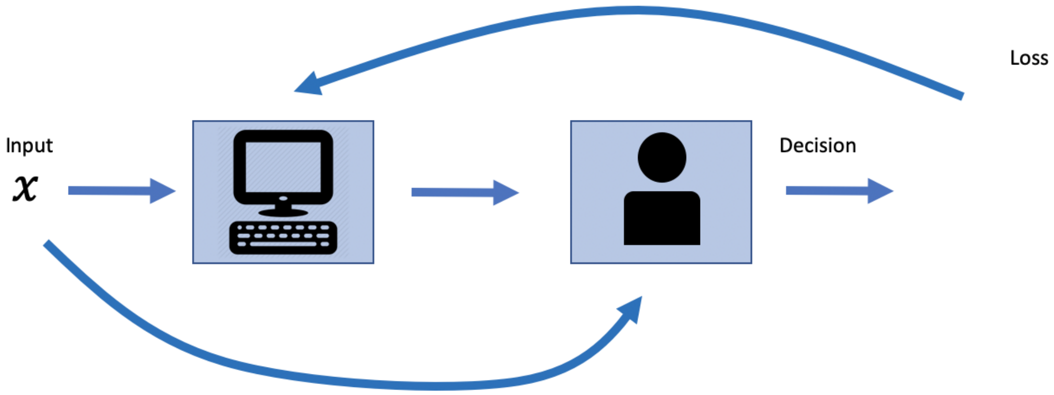


Figure 1.5: *Machine models provide information to a human decision-maker. The machine model receives feedback based on the quality of the human output.*

In this thesis, I argue that improving human interactions with machine learning models requires that models take into account human decision processes, preferences, and biases. In order to properly account for the performance of the entire machine-human pipeline, machine models must "close the loop", considering the human outcomes that result from their outputs. Optimization of machine parameters should be based on the pipeline loss, rather than optimizing the machine model in isolation.

Bansal et al. [Ban+21] demonstrate some simple ways in which optimizing machine models for total pipeline expected utility may differ from optimizing for maximum machine-only accuracy. The authors show that, for example, when humans reject low-confidence machine predictions, it can be preferable for the machine model to sacrifice accuracy in low-confidence regions to increase the number of high-confidence, high-accuracy examples. While these effects are illustrated in controlled, stylized experiments in which mathematical models take the place of human users, the disparity between machine-optimal model and human-optimal model may be even more pronounced with actual human users. Foundational research in behavioral economics and psychology reveals that humans make decisions less predictably than mathematical calculations would suggest: for example, valuing risk and probabilities inconsistently [Sim09; GLW06; TK74], arriving at different decisions when the same information is provided in different ways [TK81], and overestimating their own performance [KD99]. Algorithms that consider only the static data and not the dynamic human response in these situations will be unable to adapt if and when human behavior affects the optimal machine output.

In this thesis, I aim to develop methodologies for incorporating human actions into the machine training process. In particular, I provide frameworks and proof of concept experiments for incorporating human behavior in machine learning pipelines in two settings: (a) learning optimal computer-generated decision aids for human decision-makers (chapter 2) and (b) learning predictive models that reliably lead to good outcomes when interpreted prescriptively by users (chapter 3). I additionally demonstrate a vulnerability in a popular class of explainable machine learning methods. This vulnerability allows explanations to be manipulated by the party that trains the model, and more generally reveals that these explanations may not reliably represent the model's characteristics (chapter 4).

In what follows, I provide an overview of the problems addressed in the following chapters, as well as key results, in the order in which they appear in the thesis.

1.2 Human-Optimized Decision Aids

The goal of this chapter is to introduce a new paradigm for machine learning models, in which the objective is to produce outputs which, rather than being predictively optimal in isolation, are optimal when taking into account the quality of outcomes selected by human decision-makers on the basis of the machine output. As machine learning models are increasingly deployed to assist human decision-makers in high-stakes environments such as law [Kle+18], social welfare [Cho+18], and medicine [Cha+20], examples continue to emerge suggesting that predictions alone may result in a variety of human use patterns [SD19; Alb19; DAFC20].

Rather than provide human users with a single static prediction, the framework introduced in this chapter (called "mind composed with machine", or M∘M) uses neural network architectures to learn representations that are optimized for human decision quality. In the same way that representation learning is able to automatically extract and engineer features to reveal to a simple (e.g. linear) classifier for high performance, I aim to realize similar results with a *human* classifier [BCV13]. To achieve this, the pipeline models humans in-the-loop based on actual human decision outcomes made in response to provided machine outputs. Outputs from a parameterized machine network pass through a fixed visualization mechanism (e.g. plots, feature highlighting), chosen to enable human respondents to distinguish parameter variation in a way that is relevant to the task. Human respondents use this data representation (either alone or in conjunction with the original data) to make decisions.

Ideally, the loss would flow directly from the quality of the human decisions back to the machine model parameters, allowing for optimization of the full pipeline. However, since human decisions are not differentiable (we cannot "backprop through them"), I introduce a proxy network that is trained to reproduce human decisions on machine outputs. This network is then frozen and used to train the parameters of the machine model to discover new representations that are expected to improve human performance. As the representation network is updated, the representation domain drifts from that on which the original human queries were acquired, and so new human responses must be gathered periodically. In this way, training alternates between the human proxy model and the machine representation-generating network until convergence.

I provide several experiments that vary in the form of representation, the level of human interaction (synthetic, small survey group, large-scale Amazon Mechanical Turk), and the target task. These experiments show that the M○M framework is capable of generating representations that enable humans to make good decisions, including in situations where optimizing for a machine-only prediction would not produce useful aids for skeptical human users.

1.3 Predictive Models that Encourage Improvement in Human Subjects

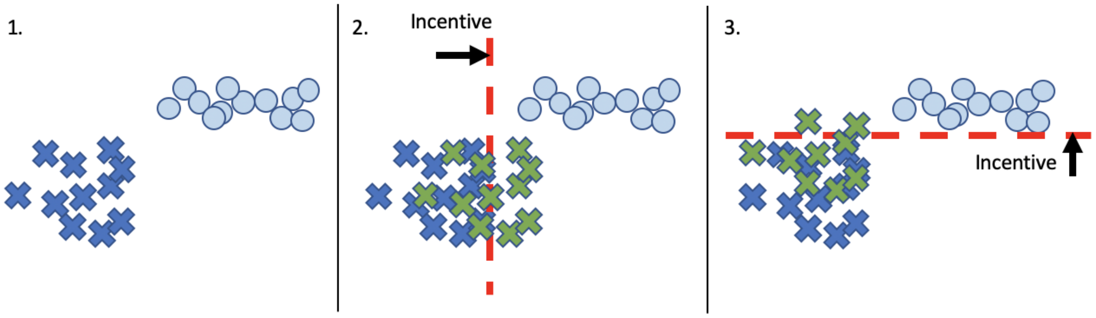


Figure 1.6: *There are several different predictors which achieve maximum accuracy on the dataset shown in (1), in which light circles are positive examples and dark crosses are negative examples. The predictor in (2) encourages subjects to make changes that shift points into areas with low data density, in which it is unclear that positive outcomes are likely. The predictor in (3) encourages subjects to shift features in ways that have more data evidence of positive outcomes.*

The goal of this chapter is to develop a new objective for training predictive models that also lead to safe and productive decisions when they are interpreted prescriptively. In particular, I formulate a new form of regularization, called "lookahead", which seeks to anticipate the actions that model subjects will take in response to model parameters and penalize models when those actions do not result in improved outcomes with high probability. The highest accuracy model on a given training dataset may be highly inaccurate outside of the training distribution, for example because it relies on feature correlations which break outside of the training set. When users are able to simulate models, they may be inclined to change their features to achieve better outcomes. However, if the model is invalid outside of the training distribution, these feature shifts could result in arbitrarily bad outcomes for the users. A more decision-minded approach would ensure that users' anticipated feature modifications continue to lie in regions for which there is sufficient training data to be confident in the model prediction. *Lookahead regularization* seeks to balance between the two objectives of predictive accuracy on the original training distribution and high decision quality. When a dataset admits multiple high-accuracy predictive models, lookahead regularization can be seen as a type of model selection, promoting the use of those models whose features are likely to encourage positive outcomes for model subjects. Because I assume that the outcome distribution is constant as input features vary (that is, feature modifications should be thought of as "investments" by the subjects, rather than attempts at "gaming" the classifier), lookahead regularization also relates to the literature on causality; models which rely on causal features will generally be more likely to yield consistent improvements, and will thus be preferred over models which rely on correlated, non-causal variables.

Lookahead regularization works by calculating the uncertainty and accompanying confidence intervals associated with outcomes as model incentives encourage users to modify features away from the original domain of the training data. Points for which high uncertainty results in the confidence lower bound lying beneath the original outcome (that is, the outcome given no strategic manipulation of features) are penalized, as we cannot be confident that these manipulations will result in safe or positive outcomes for users. The algorithm alternately optimizes three different components: the *predictive model* (with lookahead regularization), the *uncertainty model*, and a *propensity model* used to weight data points to account for covariate shift. The framework requires that the uncertainty model is differentiable and that the predictive model is twice-differentiable (as the gradient is used to predict user actions) for gradient-based optimization to be applied to the full pipeline.

I provide three experiments quantifying the accuracy-improvement tradeoffs that exist in real datasets and showing that lookahead regularization can reliably improve decision quality and in some cases also improve generalization accuracy, similarly to ℓ_2 regularization.

1.4 Vulnerabilities in Explainability Methods

The goal of this chapter is to reveal vulnerabilites in a popular class of post-hoc explanation methods, showing that these are not a reliable solution for enabling human-machine collaboration. I develop a framework for training a black-box model that can successfully hide aspects of the learned model from LIME [RSG16] and KernelSHAP [LL17]. LIME and KernelSHAP are *perturbation-based local explanation models*, meaning that they work by querying a black box model on a series of new, "perturbed" points around the explanation point of interest. Model outputs at these locations are then weighted with a method-specific distance function and used to train a local linear model, the coefficients of which are to be interpreted as feature importances. These methods are frequently promoted as a means to audit model behavior, both at individual points and over the entire domain, to determine whether model predictions align with human intuition and/or human objectives.

My experiments demonstrate how an adversarial model developer can adopt a learned model that may be biased or illegal while generating benign explanations of their choosing. While this particular scenario is hopefully unlikely, I believe that this suggests broader concerns with the family of perturbation-based explanation methods: without prior knowledge of the model's behavior, it may be difficult to choose an appropriate family of perturbations for the auditing task. Different perturbation sets can lead to different explanations, only some provide an accurate characterization of model behavior that is relevant to the auditing task. While LIME allows for perturbation distributions to be defined flexibly, there is little guidance on how best to set these parameters. I also outline some recent work in response to this discovery that has attempted to create more robust perturbation-based explanations or to more theoretically explore the limitations of LIME.

Chapter 2

Learning Representations for Human Decision-Makers

2.1 Introduction

Advancements in machine learning algorithms, as well as increased data availability and computational power, have led to the rise of predictive machines that outperform human experts in controlled experiments [Est+17; NR14; Tab+19b]. However, human involvement remains important in many domains, [Liu+19b], especially those in which safety and equity are important considerations [PON19; Bar+17] and where users have external information or want to exercise agency and use their own judgment. In these settings, humans are the final arbiters, and the goal of algorithms is to produce useful decision aids.

Given that learning algorithms excel at prediction, previous efforts in this space have largely focused on providing predictions as decision aids. This has led to a large body of work on how to make predictions accessible to decision makers, whether through models that are *interpretable* [LBL16], or through *explainable machine learning*, in which machine outputs (and so human inputs) are assumed to be predictions and are augmented with explanations [RSG16; LL17]. We see two main drawbacks to these approaches. First, setting the role of machines to ’predict, then explain’ reduces humans to auditors of the ’expert’ machines [LT19]. With loss of agency, people are reluctant to adopt predictions and even inclined to go against them [Ban89;

Ban10; Yeo+17; DSM16; YWVW19; GC19b]. This leads to a degradation in performance of the human-machine pipeline over time [Elm+15; DSM15; Log17; SD19]. More importantly, these methods cannot adapt to the ways in which predictions are used, and so are unable to adjust for systematic human errors or to make use of human capabilities.

Moving beyond predictions, in this paper we advocate for broader forms of learnable advice and capitalize on a different strength of machine learning: the ability to learn useful *representations*. Inspired by the success of representation learning, in which deep neural networks learn data representations that enable 'simple' (i.e., linear) predictors to perform well [BCV13], we leverage neural architectures to learn representations that best support human decision-makers [Kah11; Mil56]. Consider a multi-layered neural network $\mathcal{N} = f \circ \phi$ composed of a high-dimensional representation mapping ϕ and a predictor f. Our key proposal is to remove the predictor and instead plug the *human decision function* h into the learning framework to obtain $h \circ \phi$, allowing us to optimize the representation mapping to directly improve human performance.

Our framework for optimizing $h \circ \phi$, which we refer to as 'Mind Composed with Machine' (M∘M) contributes to work that seeks to bridge machine learning with human-centric design [Sut+20; Ven+03], and we make two key contributions in this regard. First, rather than machines that predict or decide, we train models that learn how to *reframe problems* for a human decision-maker. We learn to map problem instances to representational objects such as plots, summaries, or avatars, aiming to capture problem structure and preserve user autonomy. This approach of "advising through reframing" draws on work in the social sciences that shows that the quality of human decisions depends on how problems are presented [Tho80; CT92; GH95; KT13; Bro+13]. Second, rather than optimizing for machine performance, we *directly optimize for human performance*. We learn representations of inputs for which human decision-makers perform well rather than those under which machines achieve high accuracy. In this, we view our approach as taking a step towards promoting machine learning as a tool for human-intelligence augmentation [Lic60; Eng62].

The immediate difficulty in learning human-facing representations in M∘M is that h encodes how actual human decision-makers respond to representational advice and so is not amenable to differentiation (we cannot "backprop through h.") To overcome this, we propose an iterative human-in-the-loop procedure that alternates between (i) learning a differentiable *surrogate model* of human decision-making at the current representation, and (ii) training the machine model end-to-end using the current surrogate. For estimating the surrogate model we query actual humans for their decisions given a current representation.

We demonstrate the M$\circ$M framework on three distinct tasks, designed with two goals in mind: to explore different forms of human-facing representations and to highlight different benefits that come from the framework. The first experiment focuses on classifying *point clouds* in a controlled environment. Here we show how the M$\circ$M framework can learn scatter-plot representations that allow for high human accuracy without explicitly presenting machine-generated predictions (or decisions). The second experiment considers loan approvals and adopts *facial avatars* as the form of representational advice. Here we demonstrate that the framework can be applied at scale (we train using $\sim$ 5,000 queries to Amazon mTurk) and also explore what representations learn to encode and how these representations are used to support human decision-making. The third experiment is designed to demonstrate the capacity of our framework to support decision-making in ways that outperform either human or machine alone. Here we use a simulated environment to show how M$\circ$M can learn a representation that enables a human decision-maker to incorporate *side-information* (consider e.g. a hospital setting, in which doctors have the option to run additional tests or query the patient for information not included in the machine model), even when this information is known only to the user.

On the use of facial avatars: In our study on loan approval we convey advice through a facial avatar that represents an algorithmic assistant. We take care to ensure that users understand this, and understand that the avatar does *not* represent a loan applicant. We also restrict the avatar to carefully chosen variations on the image of a single actor. We are interested to experiment with facial avatars as representations because facial avatars are high dimensional, abstract (i.e., not an object that is in the domain studied), and naturally accessible to people. We are aware of the legitimate concerns regarding the use of faces in AI systems and the potential for discrimination [WC19] and any use of facial representations in consequential decision settings must be done with similar care.

2.2 Related Work

2.2.1 Modeling human factors

Recent studies have shown that the connections between trust, accuracy, and explainability can be complex and nuanced. Human users tend to use algorithmic recommendations less frequently than would be beneficial [GC19a; LT19], and user trust (as measured by agreement with algorithmic recommendation) does not increase proportionately to model accuracy [YWVW19]. Increasing model interpretability may not increase trust (as measured by agreement with the model), and may decrease users' ability to identify model errors [PS+18]. Further, even when explanations increase acceptance of model recommendations, they do not increase self-reported user trust or willingness to use the model in the future [Cra+08]. In fact, explanations increase acceptance of model recommendations even when they are nonsensical [LT19] or support incorrect predictions [Ban+20]. At the same time, understanding human interactions with machine learning systems is crucial; for example, whether or not users retain agency has been shown to affect users' acceptance of model predictions [DSM16], providing support for our approach.

Recent work acknowledges that human decision processes must be considered when developing decision support technology [LCT20; Ban+19], and work in cognitive science has shown settings in which accurate models of human decision-making can be developed [Bou+19]. Trained models of human decision-making have been successfully used to enhance the performance of reinforcement learning (RL) agents relative to that of agents trained with self-play when the evaluation requires engaging with humans. This is shown both in settings where the human interaction involves teamwork between human and computer agent [Car+19] and in settings where the computer agent attempts to adversarially manipulate human actions [DND20]. Abramson et al. [Abr+20] additionally train models that successfully imitate human evaluation of machine performance in subjective human-AI cooperative tasks, suggesting that it may be possible to tune models for human cooperation without additional human queries.

2.2.2 Humans in the loop

Despite much recent interest in training with "humans in the loop," experimentation in this setting remains an exceptionally challenging task. The field of interactive machine learning has successfully used human queries to improve machine performance in tasks where human preferences determine the gold standard [Ame+14], but human-in-the-loop training has been less productive in adapting predictive machines to better accommodate human decision-makers. In the field of interpretable machine learning, optimization for human usage generally relies on proxy metrics of human interpretability in combination with machine accuracy [Lag+19], with people only used to evaluate performance at test time. A few exceptions have allowed human feedback to guide model selection among similarly-accurate machine-optimized models [RHDV17; Lag+18], incorporating human preferences. In regard to using human responses as part of a feedback loop to a learning system, we are only aware of Lage et al. [Lag+18], and the authors actually abandoned attempts to train with mTurkers.

2.2.3 Collaboration with machine arbiters

A related field considers learning when a machine learning system should defer to a human user instead of making a prediction. This setting, unlike ours, allows the machine to bypass a human decision-maker [MPZ18; MS20; WHK20]. In this setting, human accuracy is considered to be fixed and independent of the machine learning system, and in evaluation human decisions are either fully simulated or based on previously gathered datasets.

2.3 Method

In a typical setting, a decision-making user is given an *instance* $x \in \mathcal{X}$. For clarity, consider $\mathcal{X} = \mathbb{R}^d$. Given x, the user must decide on an *action* $a \in \mathcal{A}$. For example, if x are details of a loan application, then users can choose $a \in \{\texttt{approve}, \texttt{deny}\}$. Each instance is also associated with a ground-truth *outcome* $y \in \mathcal{Y}$, so that (x, y) is sampled from an unknown distribution D. We assume that users seek to choose actions that minimize an incurred *loss* $\ell(y, a)$, with ℓ also known to the system designer; e.g., for loans, y denotes whether a loan will be repaid.

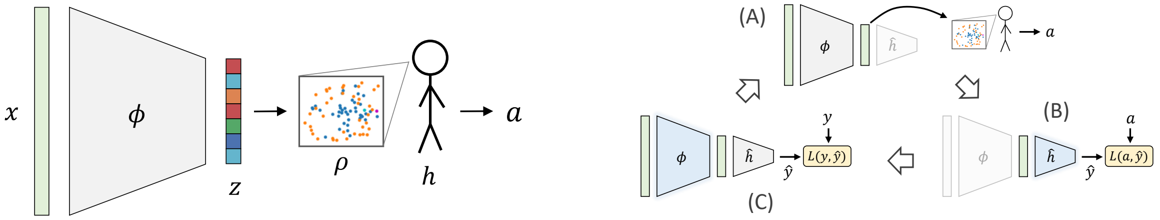


Figure 2.1: ***Left***: *The M∘M framework. The neural network learns a mapping ϕ from inputs x to representations z, such that when z is visualized through ρ, representations elicit good human decisions.* ***Right***: *Training alternates between (A) querying users for decisions on the current representations, (B) using these to train a human surrogate network $\hat{h}$, and (C) re-training representations.*

We consider the general class of *prediction policy problems* [Kle+15], where the loss function is known and the difficulty in decision-making is governed by how well y can be predicted.

We denote by h the *human mapping* from inputs to decisions or actions. For example, $a = h(x)$ denotes a decision based on raw instances x. Other sources of input such as *explanations* e or representations can be considered; e.g., $a = h(x, \hat{y}, e)$ denotes a decision based on x together with prediction $\hat{y}$ and explanation e. We allow h to be either deterministic or randomized, and conceptualize h as either representing a particular target user or a stable distribution over different kinds of users. We assume the mapping h is fixed (if there is adaptation to a representation, then h can be thought of as the end-point of this adaptation).

Crucially, we also allow machines to present users with machine-generated *advice* $\gamma(x)$, with human actions denoted as $a = h(\gamma(x))$. Users may additionally have access to *side information* s that is unavailable to the machine, in which case user actions are $a = h(\gamma(x), s)$.[1] Advice $\gamma(x)$ allows for a *human-centric representation* of the input, and we seek to *learn* a mapping γ from inputs to representations under which humans will make good decisions. The benchmark for evaluation is the expected loss of human actions given this advice:

$$\mathbb{E}_D[\ell(y, a)], \qquad \text{for} \quad a = h(\gamma(x)). \tag{2.1}$$

[1] This notion of machine-generated advice generalizes both explanations (as $\gamma = (x, \hat{y}, e)$, where e is the explanation) and deferrals (as $\gamma = (x, \bar{y})$, where $\bar{y} \in \{0, 1, \text{defer}\}$, with a human model that always accepts $\{0, 1\}$) [MPZ18].

2.3.1 Predictive advice

A standard approach provides human users with machine-generated predictions, $\hat{y} = f(x)$, where f is optimized for predictive accuracy and there is a straightforward mapping from predictions to prescribed actions $\hat{y} \to \hat{y}_a$ (e.g., for some known threshold, ′probability of returning loan′ corresponds to ′approve loan′). This is a special case of our framework where advice $\gamma = (x, \hat{y})$, and the user is modeled as $a = \hat{y}_a = h(x, \hat{y})$. The predictive model is trained to minimize:

$$\min_f \mathbb{E}_D[\ell(y, \hat{y}_a)], \qquad \text{for} \quad \hat{y} = f(x). \tag{2.2}$$

In this approach, predictions $f(x)$ are useful only to the extent that they are followed. Moreover, predictions provide only a scalar summary of the information in x, and limit the degree to which users can exercise their cognitive and decision-making capabilities; e.g., in the context of side information.

2.3.2 Representational advice

In M∘M, we allow advice γ to map inputs into representations that are designed to usefully convey information to a human decision-maker (e.g., a scatterplot, a compact linear model, or an avatar).[2] Given a *representation class* Γ we seek a mapping $\gamma \in \Gamma$ that minimizes expected loss $\min_{\gamma \in \Gamma} \mathbb{E}_D[\ell(y, h(\gamma(x)))]$. With a *training set* $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^m$ sampled from distribution D, and with knowledge of the human mapping h, we would seek γ to minimize the *empirical loss*:

$$\min_{\gamma \in \Gamma} \sum_{i=1}^{m} \ell(y_i, a_i), \qquad \text{for} \quad a_i = h(\gamma(x_i)), \tag{2.3}$$

possibly under some form of regularization (more details below). Here, Γ needs to be rich enough to contain flexible mappings from inputs to representations while also generating objects that are accessible to humans. To achieve this, we decompose algorithmic advice $\gamma(x) = \rho(\phi_\theta(x))$ into two components:

[2] We intend representations to generalize rather than oppose both predictive advice and explanations. Our primary concern is outcome quality, and in some settings predictions and explanations may be effective representations for supporting good human decisions. However, we imagine that a broader class of representations can convey more information, with representations exhibiting some correlation to inputs even after conditioning on prediction.

- $\phi_\theta : \mathbb{R}^d \to \mathbb{R}^k$ is a parameterized *embedding model* with learnable parameters $\theta \in \Theta$, that maps inputs into vector representations $z = \phi_\theta(x) \in \mathbb{R}^k$ for some $k > 1$, and
- $\rho : \mathbb{R}^k \to \mathcal{V}$ is a *visualization component* that maps each z into a visual object $v = \rho(z) \in \mathcal{V}$ (e.g., a scatterplot, a facial avatar).

This decomposition is useful because for a given application of M∘M we can now fix the visualization component ρ, and seek to learn the embedding component ϕ_θ. Henceforth, it is convenient to fold the visualization component ρ into the human mapping h, and write $h(z)$ to mean $h(\rho(z))$, for embedding $z = \phi_\theta(x)$. The training problem (2.3) becomes:

$$\min_{\theta \in \Theta} \sum_{i=1}^{m} \ell(y_i, a_i), \quad \text{for } a_i = h(\phi_\theta(x_i)), \tag{2.4}$$

again, perhaps with some regularization. By solving (2.4), we learn representations that promote good decisions by the human user. See Figure 2.1 (left).

Regularization

Regularization may play a number of different roles: as with typical L2 regularization, it may be used to reduce overfitting of the representation network, encouraging representations that generalize better to new data points. It may also be used to encourage some desired property such as sparsity, which may be beneficial for many visualizations, given the limited ability of human subjects to process many variables simultaneously. Regularization can also be used in our framework to encode domain knowledge regarding desired properties of representations, for example when the ideal representation has a known mathematical property. We utilize this form of regularization in Experiments 1 and 2.

Choosing appropriate representations

Determining the form of representational advice that best serves expert decision-makers in any concrete task will likely require in-depth domain knowledge and should be done with care. A variety of tools are available to designers with the intent of achieving a target behavior such as good decision outcomes [LHS10], and some may be more appropriate for a given task than others. At a minimum, we imagine that: i) User goals and system goals should be aligned.

Users should opt in to the system with knowledge of its methods and goals rather than being blindly manipulated [BN99]. 2) Representations must be in some sense "faithful" to the input; they should not seek to deceive users into believing something that is not true.

The characterization of varying visualizations' effects on decision-making is sufficiently elaborate as to warrant its own field of study [LM07], and thus we focus here on learning to adapt a particular choice of representation from within a set of "approved" representational forms.

2.3.3 Training procedure, and human proxy

We adopt a neural network to model the parameterized embedding $\phi_\theta(x)$, and thus advice γ. The main difficulty in optimizing (2.4) is that human actions $\{a_i\}_{i=1}^m$ depend on $\phi_\theta(x)$ via an unknown h and yet gradients of θ must pass through h. To handle this, we make use of a *differentiable surrogate for* h, denoted $\hat{h}_\eta : \mathbb{R}^k \rightarrow \Gamma$ with parameters $\eta \in H$. We learn this surrogate, referring to it as "h-hat."

The M○M *human-in-the-loop training procedure* alternates between two steps:

1. Use the current θ to gather samples of human decisions $a = h(z)$ on inputs $z = \phi_\theta(x)$ and fit $\hat{h}_\eta$.

2. Find θ to optimize the performance of $\hat{h}_\eta \circ \phi_\theta$ for the current η, as in (2.4).

Algorithm 2.1: Alternating optimization algorithm

```
Initialize $\theta = \theta_0$
repeat
    $x_1, \ldots, x_n \sim \mathcal{S}$                                    {Sample $n$ train examples}
    $z_i \leftarrow \phi_\theta(x_i) \ \ \forall\, i \in [n]$                 {Generate representations}
    $a_i \leftarrow h(\rho(z_i)) \ \ \forall\, i \in [n]$                    {Query human decisions}
    $\mathcal{T} = \{(z_i, a_i)\}_{i=1}^n$
    $\eta \leftarrow \text{argmin}_{\eta'} \mathbb{E}_\mathcal{T}[\ell(a, \hat{h}_{\eta'}(z))]$         {Train $\hat{h}$}
    $\theta \leftarrow \text{argmin}_{\theta'} \mathbb{E}_\mathcal{S}[\ell(y, \hat{h}_\eta(\phi_{\theta'}(x)))]$     {Train $\phi$}
until convergence
```

Figure 2.1 (right) illustrates this process and pseudocode is given in Algorithm 2.1. Since $\hat{h}$ is trained to be accurate for the current embedding distribution rather than globally, $\hat{h}$ is unlikely to exactly match h. However, for learning to improve, it suffices for $\hat{h}$ to induce parameter gradients that improve loss (see Figure A.3 in the Appendix). Still, h-hat must be periodically retrained because as parameters θ change, so does the induced distribution of representations z (and $\hat{h}_\eta$ may become less accurate).

Initialization of θ

In some applications, it may be useful to initialize ϕ using a machine-only model with architecture equal to $\hat{h}(\phi)$. In applications in which the human must attend to the same features as the machine model, this can help to focus ϕ on those features and minimize exploration of representations which do not contain decision-relevant information. This can be particularly useful when the representation lies within the domain of the data (e.g. plots, subsets). However, in domains in which it is possible for the machine-only setup to produce a high-accuracy model which relies on features inaccessible to human users (consider, for example, adversarial features in image recognition) this may focus ϕ too narrowly too early, making it more difficult to discover representations useful to humans. We note that the machine-only model with architecture equal to $\hat{h}(\phi)$ may be otherwise useful in model selection: the architecture of ϕ may be verified in a machine-only setting to be sufficiently flexible to achieve a desired representation distribution, and the architecture of $\hat{h}$ may be similarly verified to be capable of mapping a set of representation distributions to a set of binary answers with high accuracy.

When a desired initial distribution of representations is known, ϕ can be positioned as the generator of a Wasserstein GAN [ACB17]. In this case, the labels are not used at all, and thus the initial mapping is used only to achieve a certain coverage over the representation space and not expected to encode feature information from a machine-only model.

2.3.4 Handling Side Information

One way humans could surpass machines is through access to *side information* s that is informative of outcome y yet unknown to the machine. The M$\circ$M framework can be extended to learn a representation $\gamma(x)$ that is optimal conditioned on the existence of s, despite the machine having no access to s. At test time, the human has access to s, and so action $a = h(\phi(x), s)$. The observation is that the ground-truth outcome y, which is available during training, conveys information about s: if s is informative of y, then there exist x for which the outcome y varies with s. Thus (x, y) is jointly informative of s: for such x, knowing y and modeling the mechanism $y = g_x(s)$ by which s affects y for a given x would allow reverse-engineering the value of s as $g_x^{-1}(y)$. Although s cannot generally be exactly reconstructed without supervision on s (e.g. due to inexact modeling or non-invertibility of g_x), in some cases (x, y) can be used to make useful inference about s. Intuitively, note that for a given x, multiple $y \in \{y_1 \dots y_k\}$ values correspond to multiple s values. If h varies with s, without access to s or y, the best $\hat{h}(x)$ we can learn is $\mathbb{E}_{s \sim S}[h(x, s)]$. With varied y_i which correspond to different values of s, we can learn $\hat{h}(x, y_i) = \mathbb{E}_{s \sim S|y=y_i}[h(x, s)]$ for each y_i, which allow $\hat{h}$ to incorporate information about s.

2.4 Experimental Results

We report the results of three distinct experiments. Our intent is to demonstrate the breadth of the framework's potential, and the experiments we present vary in the task, the form of advice, their complexity and scale, and the degree of human involvement (one experiment is simulated, another uses thousands of mTurk queries). We defer some of the experimental details to the Appendix.

Model Selection

Experimenting with people in-the-loop is expensive and time-consuming, making standard practices for model selection such as cross-validation difficult to carry out. This necessitates committing to a certain model architecture at an early stage and after only minimal trail-and-error. In our experiments, we rely on testing architectures in a machine-only setting with various input and output distributions to ensure sufficient flexibility to reproduce a variety of potential mappings, as well as limited human testing with responses from the authors. Our model choices produced favorable results with minimal tuning. We believe this suggests some useful robustness of the approach to model selection choices, but future work would be beneficial to better understand sensitivity to model selection.

2.4.1 Decision-compatible scatterplots

In the first experiment, we focus on learning useful, low-dimensional representations of high-dimensional data, in the form of scatterplots. To make high-dimensional data more accessible to users, it is common practice to project into a low-dimensional embedded space and reason based on a visualization, for example a scatter plot or histogram. The choice of how to project high-dimensional data into a lower-dimensional space is consequential to decision-making [KAH19], and yet standard dimensionality-reduction methods optimize statistical criteria (e.g., maximizing directional variation in PCA) rather than optimizing for success in user interpretation. The M∘M framework learns projections that, once visualized, directly support good decisions.

We consider a setting where the goal is to correctly classify objects in p-dimensional space, $p > 2$. Each x is a p-dimensional point cloud consisting of $m = 40$ points in $\mathbb{R}^p$ (so $x \in \mathbb{R}^{40p}$). Point clouds are constructed such that, when orthogonally projected onto a particular linear 2D subspace of $\mathbb{R}^p$, denoted V, they form the shape of either an 'X' or an 'O', this determining their true label y. All directions orthogonal to V contain similarly scaled random noise. In the experiment, we generate 1,000 examples of these point clouds in 3D.

Subjects are presented with a series of scatterplots, which visualize the point clouds for a given 2D projection, and are asked to determine for each point cloud its label ('X' or 'O'). Whereas a projection onto V produces a useful representation, most others do not, including those learned coming from PCA. Our goal is to show that M∘M can use human feedback to learn a projection (ϕ) that produces visually meaningful scatterplots (ρ), leading to good decisions.

Model

Here, representation ϕ plays the role of a dimensionality reduction mapping. We use $d = 3$ and set ϕ to be a 3x2 linear mapping with parameters θ as a 3x2 matrix. This is augmented

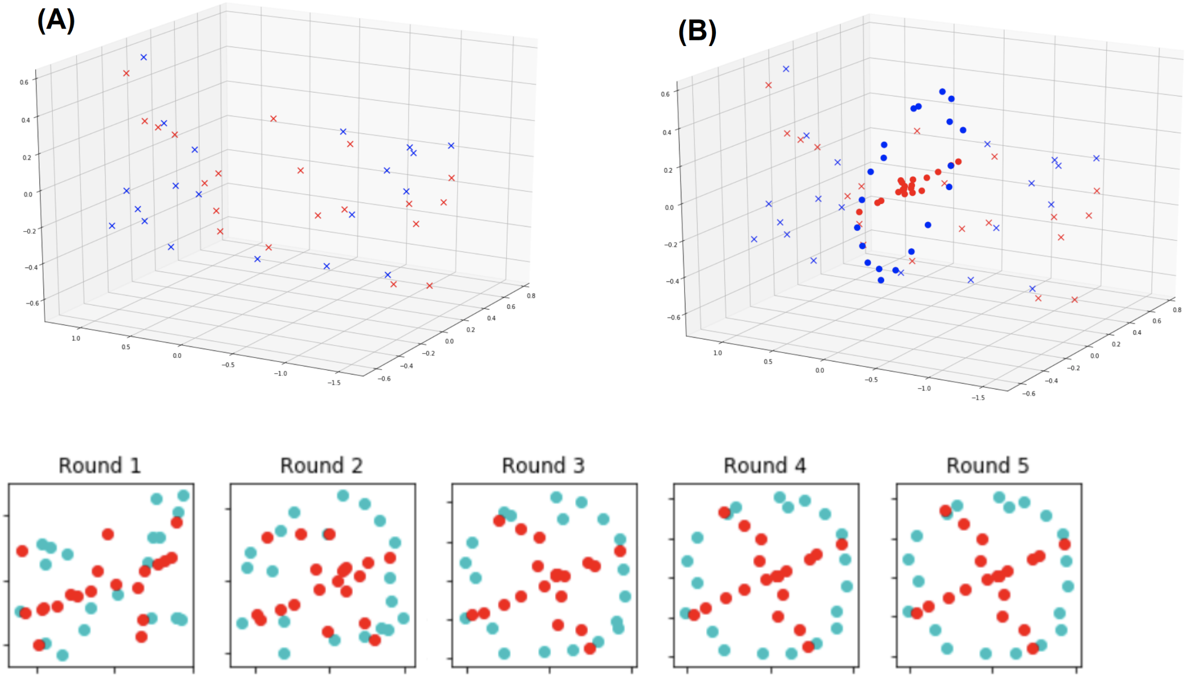


Figure 2.2: *2D representations of point clouds.* ***(A)*** *Points in their original 3D representation give little visual indication of class (X or O).* ***(B)*** *Shapes become easily distinguishable when projected onto an appropriate subspace (shown in bold).* ***(Bottom)*** *Learned 2D representations after each training round ('X', 'O' are overlaid). The initial 2D projection (round 1), on which a machine-classifier is fully accurate, is unintelligible to people. However, as training progresses, feedback improves the projection until the class becomes visually apparent (round 4), with very high human accuracy.*

with an orthogonality penalty $\phi^T\phi - \mathbb{I}$ to encourage matrices which represent rotations. For the human proxy model, we want to be able to roughly model the visual perception of subjects. For this, we use for $\hat{h}$ a small, single-layer 3x3 convolutional network, that takes as inputs a soft (differentiable) 6x6 histogram over the 2D projections.

For training, we use a fixed number of epochs (500 for $\hat{h}$ and 300 for ϕ) with base learning rates of .07 and .03, respectively, that increase with lower accuracy scores and decrease with each iteration. We have found these parameters to work well in practice, but observed that results were not sensitive to their selection.

Results

We recruited 12 computer science students to test the M$\circ$M framework.[3] Participants watched an instructional video and then completed a training and testing phase, each having five rounds (with intermittent model optimization) of 15 queries to label plots as either 'X' or 'O'. The results we provide refer to the testing phase. Round 1 includes representations based on a random initialization of model parameters and therefore serves as a baseline condition. The results show that participants achieve an average accuracy of 68% in round 1, but improve to an average accuracy of 91% in round 5, a significant improvement of 23% ($p < .01$, paired t-test) with 75% of participants achieving 100% accuracy by round 5. Subjects are never given machine-generated predictions or feedback, and improvement from training round 1 to testing round 1 is negligible (3%), suggesting that progress is driven solely by the successful reframing of problem instances (not humans getting better at the task).

Figure 2.2 demonstrates a typical example of a five-round sequential training progression. Initially, representations produced by M$\circ$M are difficult to classify when θ is initialized arbitrarily. (This is also true when θ is initialized with a fully accurate machine-only model.) As training progresses, feedback regarding subject perception gradually rotates the projection, revealing distinct class shapes. Training progress is made as long as subject responses carry some machine-discernible signal regarding the subject's propensity to label a plot as 'X' or 'O'. M$\circ$M utilizes these signals to update the representations and improve human performance.

2.4.2 Decision-compatible algorithmic avatars

For this experiment we consider a real decision task and use real data (approving loans), train with many humans participants (mTurkers), and explore a novel form of representational advice (facial avatars). Altogether we elicit around 6,000 human decisions for training and evaluation. Specifically we use the *Lending Club* dataset, focusing on the resolved loans, i.e., loans that were paid in full ($y = 1$) or defaulted ($y = 0$), and only using features that would

[3] All experiments are conducted subject to ethical review by the university's IRB.

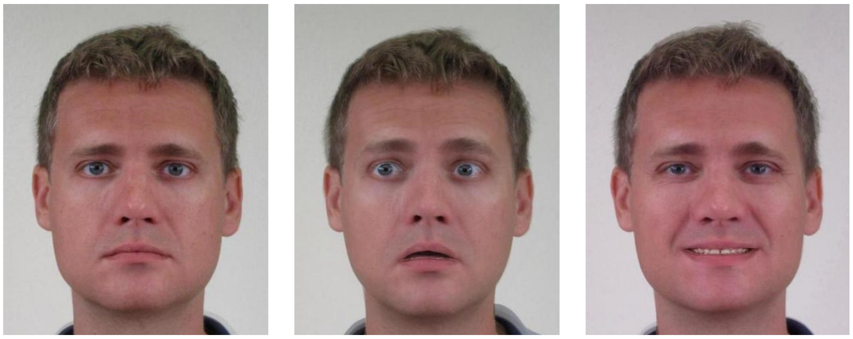


Figure 2.3: *Different facial avatars, each avatar representing an algorithmic assistant and not a loan applicant, and trained to provide useful advice through facial expressions. The leftmost avatar is set to a neutral expression ($z = 0$).*

have been available to lenders at loan inception.[4] The decision task is to determine whether to approve a loan ($a = 1$) or not ($a = 0$), and the loss function we use is $\ell(y, a) = \mathbb{1}_{\{y \neq a\}}$.

Goals, expectations, and limitations

Whereas professional decision-makers are inclined to exercise their own judgment and deviate from machine advice [SD19; DAFC20], mTurkers are non-experts and are likely to follow machine predictions [LT19; YWVW19].[5] For this reason, the goal of the experiment is *not to demonstrate performance superiority over purely predictive advice*, nor to show that mTurkers can become expert loan officers. Rather, the goal is to show that abstract representations can convey predictive advice in a way that requires users to deliberate, and to explore whether humans use learned representations differently than they use machine predictions in making decisions. In Appendix A.1 we further discuss unique challenges encountered when training with mTurkers in the loop.

Representations

With the aim of exploring broader forms of representational advice, we make use of a *facial avatar*, framed to users as an *algorithmic assistant*— not the recipient of the loan —and communi-

[4] https://www.kaggle.com/wendykan/lending-club-loan-data

[5] We only know of Turk experiments where good human performance from algorithmic advice can be attributed to humans accepting the advice of accurate predictions [LCT20].

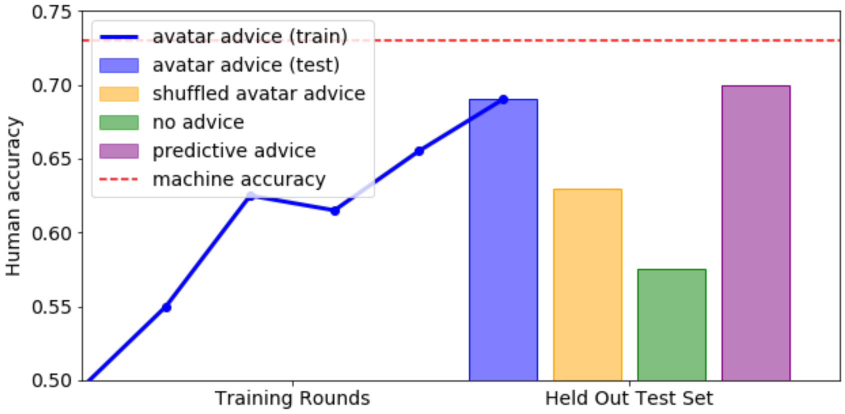


Figure 2.4: *Human accuracy in the algorithmic advice condition ('avatar advice') consistently increases over rounds. Performance quickly surpasses the 'no advice' (data only) condition, and steadily approaches performance of users observing algorithmic predictions ('predictive advice'), which in itself is lower than machine-only performance ('machine accuracy'). Human accuracy falls when faces are shuffled within predicted labels of $\hat{h}$, confirming that faces convey useful, multi-variate information.*

cating through its facial expressions information that is relevant to a loan decision. The avatar is based on a single, realistic-looking face capable of conveying versatile expressions (Figure 2.4 includes some examples). Expressions vary along ten dimensions including *basic emotions* [DTM14], *social dimensions* (e.g., dominance and trustworthiness [DTM14; Tod+08]), and subtle changes in *appearance* (e.g., eye gaze). Expressions are encoded by the representation vector z, with each entry corresponding to a different facial dimension. Thus, vectors z can be thought of as points in k-dimensional 'face-space' in which expressions vary smoothly with z.

We are interested in facial avatars because they are abstract (i.e., not in the domain of the input objects) and because they have previously been validated as useful representations of information [Che73; LD90]. They are also high-dimensional representations, and non-linear in the input features; that is, faces are known to be processed holistically with dependencies beyond the sum of their parts [Ric+09]. Faces also leverage innate human cognition—immediate, effortless, and fairly consistent processing of facial signals [Iza94; Tod+08; FJ16].

Through M∘M, we *learn* a mapping from inputs to avatars that is useful for decision-making. Training is driven completely by human responses, and learned expressions reflect usage patterns that users found to be useful, as opposed to hand-coded mappings as in *Chernoff faces* [Che73].

Model and training

We set ϕ to be a small, fully connected network with a single 25-hidden unit layer, mapping inputs to representation vectors $z \in \mathbb{R}^9$. The visualization component $\rho(z)$ creates avatars by morphing a set of base images, each corresponding to a facial dimension, with z used to weight the importance of each base image.[6,7] For regularization, we additionally consider the loss of a decoder network implemented by an additional neural network, which attempts to reconstruct the input x from the representation. This term encourages points in face-space to preserve distances in instance-space at the cost of some reduction in accuracy. This promotes representations that carry more information about inputs than that implied by simple predictions. For $\hat{h}$ we use a small, fully connected network with two layers of size 20 each, operating directly on representation vectors z.

In collecting human decisions for training $\hat{h}$, mTurkers were queried for their decisions regarding the approval or denial of loan applications.[8] New users were recruited at each round to obtain reports that are as independent as possible and to control for any human learning. Each user was queried for a random subset of 40 training examples, with the number of users chosen to ensure that each example would receive multiple responses (w.h.p.). For predictive purposes, binary outputs were set to be the majority human response. Each loan application was presented using the most informative features as well as the avatar. We did not relate to users any specific way in which they should use avatar advice, and *care was taken to ensure users understood that the avatar does not itself represent an applicant*.[9] Appendix A.2.2 provides additional experimental details.

[6] Morphed images were created using the *Webmorph* software package [DT16].

[7] All base images correspond to the same human actor, whose corresponding avatar was used throughout the experiment.

[8] As all users share the same representation mapping, we restrict to US participants to promote greater cross-user consistency.

[9] Respondents who did not understand this point in a comprehension quiz were not permitted to complete the task.

Results

Our results show that M$\circ$M can learn representations that support good decisions through a complex, abstract representation, and that this representation carries multivariate information, making it qualitatively different than prediction. As benchmarks, we consider the accuracy of a trained neural network model $\mathcal{N}(x)$ having architecture equal to $\hat{h} \circ \phi$ (but otherwise unrelated to our human experiments), as well as human performance under predictive advice $\gamma(x) = \tilde{y} \in [0, 1]$ where $\tilde{y}$ is the predicted probability of $\mathcal{N}(x)$. We also consider a condition with 'shuffled' avatar advice, which we describe below.

Figure 2.4 shows the training process and resulting test accuracy (data is balanced so chance ≈ 0.5).[10] At first, the (randomly-initialized) representation ϕ produces arbitrary avatars, and performance in the avatar condition is lower than in the no-advice condition. This indicates that users take into account the (initially uninformative) algorithmic advice. As learning progresses, user feedback accumulates and the accuracy from using the M$\circ$M framework steadily rises. After six rounds, avatar advice contributes to a boost of 11.5% in accuracy (0.69) over the no-advice condition (0.575), reaching 99% of the accuracy in the predictive advice condition (0.70). Performance in the predictive advice condition does not reach machine accuracy (0.73), showing that not all subjects follow predictive advice.

Analysis. We additionally explore what the representations learn, and how humans incorporate them into predictions. One possible concern is that despite regularization, learned avatars may simply convey stylized binary predictions (e.g., happy or sad faces). To explore this, we added a 'shuffled' condition in which faces are shuffled within predicted labels of $\hat{h}$. As shown in Figure 2.4, shuffling degrades performance, confirming that faces convey more information than the system's binary prediction. Moreover, the avatars do not encode a univariate (but not binary) prediction, and humans do not use the information in the same way that they use numeric predictions: (i) no single feature of z has a correlation with human responses $\hat{h}(z)$ of more than $R^2 = 0.7$, (ii) correlations of average human response with features z are low ($R^2 \leqslant 0.36$ across features) while responses in the predictive condition have $R^2 = 0.73$ with the

[10] Results are Statistically significant under one-way ANOVA, $F(3, 196) = 2.98, p < 0.03$.

predictions, and (iii) users in the avatar condition self-report using the data as much or more than the advice 83% of the time, compared to 47% for the predictive advice condition.

At the same time, z preserves important information regarding x. To show this, we train linear models to predict from z each of the data features: interest rate (RATE), loan term (TERM), debt to income ratio (DTI), negative public records (REC), annual income (INC), employment length (EMP). Results show that z is highly informative of RATE ($R^2 = 0.79$) and TERM (0.57), mildly informative of REC (-0.21), INC (0.23), and EMP (0.13), and has virtually no predictive power of DTI (-0.03). Further inspecting model coefficients reveals a complex pattern of how z carries information regarding x (see Appendix A.2.2 for all coefficients). E.g.: trustworthiness plays an important part in predicting all features, whereas anger is virtually unused; happiness and sadness do not play opposite roles—happiness is significant in TERM, while sadness is significant in RATE; and whereas EMP is linked almost exclusively to age variation, INC is expressed by over half of the facial dimensions.

Because representations are driven solely by performance, any analysis of *how* they are used is necessarily post hoc. In our setting, with no individualized user-representation training loop, a reasonably hypothesis is that avatars help convey how a given data point relates to the full training set of data points. For example, extreme avatars suggest outliers, either positive or negative, and help users both to evaluate those data points and to gain a better understanding of the data distribution (e.g. means, deviations) earlier in the decision process. Again, we do not suggest that avatars are the ideal data representation for *this* task. Avatars may, however, be exceptionally useful in scenarios where engaging some level of emotion or social reasoning (e.g. retirement planning [Her+11]) has been shown to lead to better decisions than data alone.

2.4.3 Incorporating side information

To demonstrate additional capabilities of M$\circ$M we show that the framework can also learn representations that allow a decision maker to leverage side information that is unavailable to the machine. Access to side information is one advantage humans may have over machines, and our goal here is to show the potential of representations in eliciting decisions whose quality surpasses that attainable by machines alone. We adopt simulation for this experiment because it is challenging for non-experts (like mTurkers) to outperform purely predictive advice, even with access to additional side information. Simulation also allows us to systematically vary the synthetic human model, and we consider four distinct models of decision-making.

We consider a medical decision-making task in which doctors must evaluate the health risk of incoming ER patients and have access to a predictive model. [11] Here, we focus on compact, linear models, and view the model coefficients along with the input features as the representation, affecting the decision process of doctors. Doctors additionally have access to side information that is *unavailable to the model* and may affect their decision. Our goal is to learn a model that can account for how doctors use this side information.

Setup

There are four primary binary features $x \in \{0,1\}^4$: diabetes (x_d), cardiovascular disease (x_c), race (x_r), and income level (x_i). An integer 'side-information' variable $s \in \{0,1,2,3\}$ encodes how long the patient's condition was allowed to progress before coming to the ER and is available only to the doctor. We assume ground-truth risk y is determined only by diabetes, cardiovascular disease, and time to ER, through $y = x_d + x_c + s$, where x_d, x_c, s are sampled independently. We also assume that x_r, x_i jointly correlate with y (e.g. due to disparities in access), albeit not perfectly, so that they carry some but not all signal in s, whereas x_d, x_c do not. In this way, x_r and x_i offer predictive power beyond that implied by their correlations with known health conditions (x_d, x_c), but interfere with use of side information.

Specifically, a latent variable $l_0 \sim \mathcal{N}(.3, .1)$ introduces a low correlation between x_i and x_r by setting a common mean for their Bernoulli probabilities l_1, l_2:

- $l_1, l_2 \sim \text{Unif}(\max(l_0 - .3, 0), \min(l_0 + .3, 1))$
- $x_i \sim \text{Bernoulli}(1 - l_1)$
- $x_r \sim \text{Bernoulli}(1 - l_2)$

An additional latent variable l_3 provides a similar correlation between x_c and x_d, which also correlate, respectively, with x_i and x_r:

[11] MDCalc.com is one example of a risk assessment calculator for use by medical professionals.

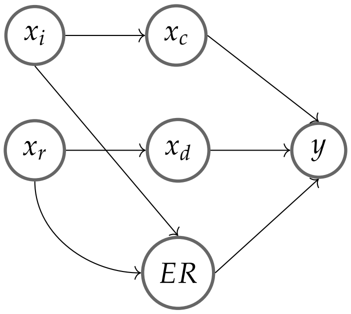


Figure 2.5: *Relationship of variable correlations in the side information experiment*

- $l_3 \sim \text{Unif}(.5, .7)$
- $x_c \sim \text{Bernoulli}(l_3 + x_i)$
- $x_d \sim \text{Bernoulli}(l_3 + x_r)$

A directed graph showing the variable correlations is shown in Figure 2.5.

Side information s is highly correlated with x_r and x_i but noisy: s is drawn from a normal distribution centered at $x_r + x_i$ before rounding to an integer value between 0 and 3.

- $s_{cont} \sim \mathcal{N}(x_r + x_i, .5)$
- $s = \max(0, \min(3, \text{round}(s_{cont})))$

The integer outcome variable y is the sum of x_c, x_d, and s. The binary outcome variable y_{bin} is thresholded at $y > 3$.

We model a decision maker who generally follows predictive advice $\hat{y} = f_w(x) = \langle w, x \rangle$, but with the capacity to adjust the machine-generated risk scores at her discretion and in a way that depends on the model through its coefficients w. We assume that doctors are broadly aware of the correlation structure of the problem, and are prone to incorporate the available side information s into $\hat{y}$ if they believe this will give a better risk estimate. We model the decisions of a population of doctors as incorporating s additively and with probability that decreases with the magnitude of either of the coefficients w_r or w_i. We refer to this as the *or* model and set $h_{\text{or}}(x, s, w) = \hat{y} + I(w) \cdot s$ with $I(w) \propto 1/(\max\{w_r, w_i\})$. We also consider simpler decision models: *always* using side information (h_{always}), *never* using side information (h_{never}), and a *coarse* variant of h_{or} using binarized side information, $h_{\text{coarse}} = \hat{y} + I(w) \cdot 2 \cdot \mathbb{1}\{s \geqslant 2\}$.

	M∘M	h(**Machine**)
Or	1.0	.894
Coarse Or	.951	.891
Never	.891	.891
Always	1.0	.674

Table 2.1: *Performance of M∘M with side information on four synthetic human models. Machine-only performance is 0.890.*

Model

The representation $\rho(z)$ consists of x, coefficients w (these are learned within ϕ), and $\hat{y} = \langle w, x \rangle$. [12] The difficulty in optimizing ϕ is that s is never observed, and our proposed solution is to use y (which is known at train time) as a proxy for s when fitting $\hat{h}$, which is then used to train ϕ (see Section 2.3). Since x and y jointly carry information regarding s, we define $\hat{h}(x, y; w) = \langle w, x \rangle + \hat{s}(x, y)$, where $\hat{s}(x, y) = v_0 y + \sum_{j=1}^{4} v_j x_j$, and v are parameters. Note that it is enough that $\hat{s}$ models how the user *utilizes* side information, rather than the value of s directly; s is never observed, and there is no guarantee about the relation between $\hat{s}$ and s.

Results

We compare M∘M to two other baselines: a machine-only linear regression, and the human model h applied to this machine-only model, and evaluate performance on the four synthetic human models ($h_{\text{or}}, h_{\text{coarse}}, h_{\text{never}}$, and h_{always}). Both M∘M and the baselines use a linear model but the model in M∘M is trained to take into account how users incorporate side information. For evaluation, we consider binarized labels $y_{bin} = \mathbb{1}\{y > 3\}$.

We report results averaged over ten random data samples of size 1,000 with an 80-20 train-test split. As Table 2.1 shows, due to its flexibility in finding a representation that allows for incorporation of side information by the user, M∘M reaches 100% accuracy for the *or* and *always* decision models. M∘M maintains its advantage under the *coarse-or* decision model (i.e., when doctors use imperfect information), and remains effective in settings where side information is never used. The problem with the baseline model is that it includes non-zero

[12] In an application, the system should convey to users that it is aware they may have side information.

coefficients for all four features. This promotes accuracy in a machine-only setting, and in the absence of side information. Given this, the *or* and *coarse-or* decision models only very rarely introduce the side information— and this is indeed the best they can do given that the machine model uses all four variables. In contrast, for the *always* decision model the user always introduces side information, causing over-counting of the time to ER effect on patient outcomes (because of correlations between s and x_r and x_i). In contrast, M○M learns a linear model that is responsive to the human decision-maker: for example, including non-zero coefficients for only x_d and x_c with the *or* decision model.

2.5 Discussion

We have introduced a novel learning framework for supporting human decision-making. Rather than view algorithms as experts, asked to explain their conclusions to people, we position algorithms as advisors whose goal is to help humans make better decisions while retaining human agency. The M○M framework learns to provide representations of inputs that provide advice and promote good decisions. We see this as a promising direction for promoting synergies between learning systems and people and hope that by tapping into innate cognitive human strengths, learned representations can improve human-machine collaboration by prioritizing information, highlighting alternatives, and correcting biases.

2.5.1 Ethics

By incorporating the use of human judgment rather than encouraging human automation bias or simply automation, the kinds of methods suggested here have the potential to fail more gracefully than traditional decision support systems. Still, the idea of seeking to optimize for human decisions should not be considered lightly.

It is our belief that a responsible and transparent deployment of models with "h-hat-like" components should encourage environments in which humans are aware of what information they provide about their thought processes. Unfortunately, this may not always be the case, and ethical, legal, and societal aspects of systems that are optimized to promote particular human decisions must be subject to scrutiny by both researchers and practitioners. If designed to correct for inadvertent user biases, for example, the system will first have to learn these biases, and this can be sensitive information and damaging to users if not properly managed. These kinds of issues are not specific to our framework and have been a concern of the HCI community as early as 1998 [Fog98]. The opportunities and dangers of our framework generally reflect those of the broader field of persuasive technology [BN99], where system goals may be poor proxies for user goals [Rib+20], or even at odds with user goals. Moreover, the method does not in itself prevent biases from being passed through the data without appropriate care in the design of loss functions.

Still, we see reasons to be optimistic regarding the future of algorithmic decision support. Systems designed specifically to provide users with the information and framing they need to make good decisions can harness the strengths of both computer pattern recognition and human judgment and information synthesis. We can hope that the combination of man and machine can do better than either one alone. The ideas presented in this paper serve as a step toward this goal.

Chapter 3

Learning Models that Induce Good Decisions

3.1 Introduction

Machine learning is increasingly being used in domains that have considerable impact on people, ranging from healthcare [CS17], to banking [Sid12], to manufacturing [Wue+16]. In many of these domains, fairness and safety concerns promote the deployment of fully transparent predictive models. An unavoidable consequence of this transparency is that end-users are prone to use models *prescriptively*: if a user (wrongly) views a predictive model as a description of the real world phenomena it models (e.g., heart attack risk), then she may look to the model for how to adapt her features in order to improve future outcomes (e.g., reduce her risk). But predictive models optimized for accuracy cannot in general be assumed to faithfully reflect post-modification outcomes, and model-guided actions can prove to be detrimental. The goal in this chapter is to present a learning framework for organizations seeking to deploy learned models in a way that is transparent *and* responsible, a setting I believe applies widely. Consider a medical center who would like to publish an online tool to allow users to estimate their heart attack risk, while keeping in mind that users may also infer how lifestyle changes will affect future risk.[1] Consider a lender who would like to be transparent about their first-time mortgage approval process, while knowing that this will suggest to applicants how they may alter their credit profiles. Consider a wine reseller, who would like to provide demand guidance to producers through an interpretable model while considering that producers may use the same guidance to modify future vintages. Each of these organizations must be cognizant of the actions their classifiers promote, and seek to emphasize features that promote safe adaptations as well as predictive accuracy.

It is well understood that correlation and causation need not go hand-in-hand [Pea+09; Rub05]. What is novel about this work is that I seek models that serve the dual purpose of achieving predictive accuracy and providing high confidence that decisions made with respect to the model are safe. That is, I care foremost about the utility that comes from having a predictive tool, but recognize that these tools may also drive decisions.

To illustrate the potential pitfalls of a naïve predictive approach, consider a patient who seeks to understand his or her heart attack risk. If the patient consults a linear predictive model (as is often the case for medical models, see [UR16]), then a negative coefficient for alcohol consumption may lead the patient to infer that a daily glass of red wine would improve his or her prognosis. Is this decision justified? Perhaps not, although this recommendation has often been made based on correlative evidence and despite a clear lack of experimental support [HAB17; Sah+15].

The main insight is that *controlling the tradeoff between accuracy and decision quality, where it exists, can be cast as a problem of model selection*. For instance, there may be multiple models with similar predictive performance but different coefficients, that therefore induce different decisions [Bre+01]. To achieve this tradeoff, this chapter introduces *lookahead regularization*, which balances accuracy and the improvement associated with induced decisions. This is achieved by modeling how users will act, and penalizing a model unless there is high confidence that decisions will improve outcomes.

Formally, these decisions in response to a model f induce a *target distribution* p^f on

[1] For example, the Mayo Clinic, a leading medical center in the U.S., provides such a calculator [Cli20]. `MDCalc.com` is an example of a site that provides medical many risk assessment calculators to the public.

covariates that may differ from the distribution of data at training, p. In particular, a decision will map an individual with covariates x to new covariates x^f. For a prespecified confidence level τ, we want to guarantee improvement for at least a τ-fraction of the population, comparing outcomes under p^f in relation to outcomes in p (under an invariance assumption on $p(y|x)$, where y is the outcome). The technical challenge is that p^f may differ considerably from p, resulting in uncertainty in estimating the effect of decisions. To solve this, lookahead regularization makes use of an uncertainty model that provides confidence intervals around decision outcomes for different examples x^f. A discriminative uncertainty model is trained through importance weighting [Gre+09; Shi00; Sug+08] to handle covariate shift with the distribution of x^f different from x, and is designed to estimate accurate intervals for p^f.

Lookahead regularization has stages that alternate between optimizing the different components of the framework: the *predictive model* (under the lookahead regularization term), the *uncertainty model* (used within the regularization term), and the *propensity model* (used for covariate shift adjustment). If the uncertainty model is differentiable and the predictive model is twice-differentiable, then gradients can pass through the entire pipeline and gradient-based optimization can be applied. We run three experiments. One experiment uses synthetic data to illustrate the approach, helping to understand what is needed for lookahead regularization to succeed. The second experiment considers an application to wine quality prediction, and shows that even simple tasks lead to interesting tradeoffs between accuracy and improved decisions. The third experiment focuses on predicting diabetes progression and includes a demonstration of the framework in a setting with individualized actions.

3.1.1 Related work

Strategic Classification. In the field of *strategic classification*, the learner and agents, who are the subjects of a model, engage in a Stackelberg game, where the learner attempts to publish a maximally accurate classifier taking into account that agents will shift their features to obtain better outcomes under the classifier [Har+16]. While early efforts viewed all modifications as "gaming"— an adversarial effect to be mitigated [Don+18; BS11] —a recent trend has focused on creating incentives for modifications that lead to better outcomes *under the ground* *truth function* rather than simply better classifications [KR19; Alo+; Hag+20; Tab+19a]. In the absence of a known mapping from effort to ground truth, Miller, Milli, and Hardt [MMH20] show that incentive design relates to causal modeling, and several responsive works explore how the actions induced by classifiers can facilitate discovery of these causal relationships [Bec+20; SEA20]. The effect of strategic classification on algorithmic fairness has also motivated several works [Liu+20; HIV19; Mil+19]. Generally, these works consider the equilibrium effects of classifiers, or other sequential aspects of their deployment, where the choice of model affects covariate distributions and in turn predictive accuracy. In contrast, we consider what can be done at a particular point in time, with awareness of the decisions a model will induce and consideration for subject outcomes under those decisions. Further, our invariance assumption on $p(y|x)$ rules out "gaming" of features. Rather, we view feature changes as effortful investments that affect ground truth outcomes.

Recent work on *performative prediction* [Per+20] studies the equilibrium of retraining dynamics in settings where the model at each round affects the next input distributions (this generalizes strategic classification). Training is focused entirely on accuracy, and does not consider the quality of induced decision outcomes (these can be arbitrarily bad). We study a different setting of one-time interactions between users and a model (consider first-time mortgage buyers or consumers who access a medical risk calculator online), focusing on the tradeoff between predictive accuracy (and as it relates to p and not p^f) and decision outcomes (under p^f). Our model is also relevant in settings with feedback coming in slowly, with models being intermittently re-trained, and where decision outcomes are consequential at each step of the retraining process.

Causality, Covariate Shift, and Distributionally Robust Learning. There are many efforts in ML to quantify the uncertainty associated with predictions and identify domain regions where models err [LPB17; HLA15; GG16; Guo+17; TLP19; Liu+19a]. However, most methods fail to achieve desirable properties when deployed out of distribution (OOD) [Sno+19]. When the shifted distribution is unknown at train time, distributionally robust learning can provide worst-case guarantees for specific types of shifts but require unrealistic computational expense or restrictive assumptions on model classes [SND18]. Our framework is concerned only with the single, specific OOD distribution that is induced by the learned predictive model. Hence, we need only guarantee robustness to this particular distribution, for which we make use of tools from learning under covariate shift [BBS09]. Relevant to our task, Mueller et al. [Mue+17] seek to identify treatments that are beneficial with high probability under the invariance assumption on $p(y|x)$. These treatments are chosen directly from a flexible set of available modifications on covariates, whereas we assume treatments on covariates are induced by published predictive models and additionally consider the accuracy of such models. Because model variance generally increases when covariate shift acts on non-causal variables [PBM16], our framework of trading off uncertainty minimization with predictive power relates to efforts in the causal literature to find models that have optimal predictive accuracy while being robust to classes of interventional perturbations [Mei18; Rot+18].

Conformal Prediction. Techniques from *conformal prediction* attempt to identify an interval around a given model prediction such that any given prediction interval contains the true label with some specified probability $1 - \epsilon$ [SV08]. Tibshirani et al. [Tib+19] adapt traditional conformal prediction techniques, which rely on *exchangeability* of training and test example sequences, to settings with covariate shift. The conformal prediction setting differs from ours in that conformal prediction considers a guarantee on *each label* (and often in an online setting), rather than a guarantee over a population.

3.2 Method

Let $x \in \mathcal{X} = \mathbb{R}^d$ denote a feature vector and $y \in \mathbb{R}$ denote a label, where x describes the object of interest (e.g., a patient, a customer, a wine vintage), and y describes the quality of an outcome associated with x, where we assume that higher y is better (e.g. life expectancy, creditworthiness, wine score). We assume an observational dataset $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^m$, which consists of IID samples from a population with joint distribution $(x, y) \sim p(x, y)$ over covariates (features) x and outcomes y. We denote by $p(x)$ the marginal distribution on covariates.

Let $f : \mathcal{X} \to \mathbb{R}$ denote a model trained on $\mathcal{S}$. We assume that f is used in two different ways:

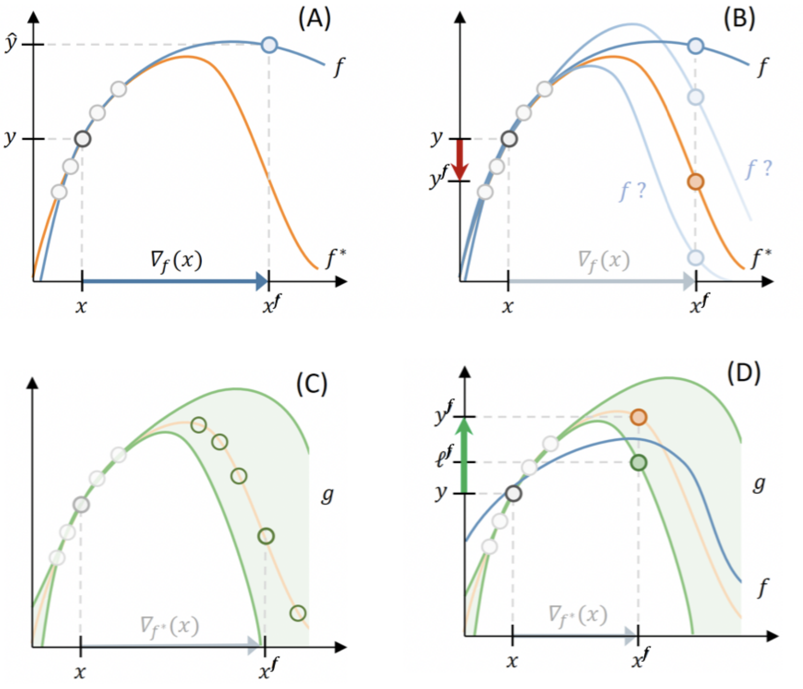


Figure 3.1: *An illustration of our approach. Here $p(y|x)$ is deterministic, $y = f^*(x)$, and the data density $p(x)$ is concentrated to the left of the peak. (A) Users (x) seeking to improve their outcomes (y) often look to predictive models for guidance on how to act, e.g. by following gradient information ($x \mapsto x^f$). (B) But actions may move x^f into regions of high uncertainty, where f is unconstrained by the training data. Models of equally good fit on p can behave very differently on p^f, and hence induce very different decisions. (C) Promoting good decisions requires reasoning about the uncertainty in decision outcomes. For this, our approach learns an interval model $g(x^f) = [\ell^f, u^f]$ guaranteeing that $y^f \in [\ell^f, u^f]$ with confidence τ, decoupled from f and targeted specifically at p^f. (D)* Lookahead regularization *utilizes these intervals to balance between accuracy and improvement, achieved by penalizing f whenever $y > \ell^f$ (Eq. (3.4)). By incorporating into the objective a model of user behavior, our approach learns predictive models encouraging safe decisions, i.e., having $y^f \geqslant y$ w.p. at least τ.*

1. **Prediction:** To predict outcomes y for objects x, sampled from $p(x)$.

2. **Decision:** To take action, through changes to x, with the goal of improving outcomes.

We will assume that user actions map each x to a new $x^f \in \mathcal{X}$. We refer to x^f as a user's *decision* or *action* and denote decision outcomes by $y^f \in \mathbb{R}$. We set $x^f = d(x)$ and refer to $d : \mathcal{X} \to \mathcal{X}$ as the *decision function*. We will assume that users consult f to drive decisions—either because they care only about predicted outcomes (e.g., the case of bank loans), or because they consider the model to be a valid proxy of the effect of a decision on the outcome (e.g., the case of heart attack risk or wine production). As in other works incorporating strategic users into learning [Per+20; Har+16], our framework requires an explicit model of how users use the model to make decisions. For concreteness, we model users as making a step in the direction of the gradient of f, but note that the framework can also support any other differential decision model.[2] Since not all attributes may be susceptible to change (e.g., diet can be changed, height is fixed), we distinguish between *mutable* and *immutable* features using a task-specific *masking operator* $\Gamma : \mathcal{X} \to \{0,1\}^d$. Further, immutable features are never affected by changes to mutable features.[3]

Assumption 1 (User decision model)**.** *Given masking operator* Γ *and predictive model* f*, we define the decision of user with features* x *as:*

$$x^f = x + \eta\Gamma(\nabla_f(x)), \tag{3.1}$$

where the step size $\eta > 0$ *is a design parameter. Any features not present in* f*, including unobserved features, do not change.*

Through Assumption 1, user decisions induce a particular decision function $d^f(x)$, and in turn, a *target distribution* over $\mathcal{X}$, which we denote $p^f(x)$. This leads to a new joint distribution $(x^f, y^f) \sim p^f(x,y)$, with decisions inducing new outcomes. Note that this assumption implies that unobserved features that do not appear in the model f will remain unchanged after user decisions. To achieve causal validity in the way we reason about the effect of decisions on outcomes, we follow Peters, Bühlmann, and Meinshausen [PBM16] and assume that y depends only on x (and optionally on additional unobserved, unconfounding variables), and that this dependence is invariant to the distribution on x.

Assumption 2 (Covariate shift [Shi00])**.** *The conditional distribution on outcomes,* $p(y|x^f)$*, is invariant for any marginal distribution* $p^f(x)$ *on covariates, including the data distribution* $p(x)$*.*

Assumption 2 says that whatever the transform d^f, the conditional distribution $p(y|x)$ is

[2] Many works consider a 'rational' decision model $x^f = \text{argmax}_{z \in \mathcal{X}} f(z) - c(x,z)$ where c is a cost function. Eq. (1) can be thought of as modeling a boundedly-rational agent, taking action to optimize a local first-order approximation of f, with η serving as a hard constraint on the amount of change that is possible. Our choice flows mostly for reasons of feasibility; in principal, rational decision models can be incorporated into learning using differentiable optimization layers (e.g., [Agr+19]). Like cost functions in all other works, η is a design choice that can be set e.g. by an expert.

[3] Γ does not reflect any causal assumptions; it merely states which features are amenable to change.

fixed, and the new joint distribution is $p^f(x^f, y) = p(y|x^f)p^f(x^f)$, for any p^f. This covariate-shift assumption ensures the causal validity of the approach (and entails the property of *no-unobserved confounders*, see Figures 3.2 and 3.3). Note that we do not require that there are no unobserved features, and nor do we require that we observe all causal features. Rather, we require covariate shift on observed features (and thus no-unobserved *confounders*), and as per Assumption 1, we require unobserved causal features to be unaffected by the published model.

Although covariate shift is a strong assumption, this kind of invariance is reasonable for many applications (e.g. writing improvement, gene perturbation), and has been leveraged in other works that relate directly to questions of causality [RC+18; Mue+17] as well as more generally for settings in which the target and training distributions differ [Sch+09; QC+09; Shi00; Sug+08].[4] There also exist important domains in which violations are sufficiently minor that this is a reasonable assumption (see [Mue+17; PBM16] for discussion).

3.2.1 Learning objective

Our goals in designing a learning framework are twofold. First, we would like learning to result in a model whose predictions $\hat{y} = f(x)$ closely match the corresponding labels y for $x \sim p(x)$. Second, we would like the model to induce decisions x^f for counterfactual distribution p^f whose outcome y^f improves upon the initial y. To balance between these two goals, we construct a learning objective in which a predictive loss function is augmented with a regularization term that promotes good decisions. The difficulty is that decision outcomes y^f depend on decisions x^f through the learned model f. Hence, realizations of y^f are unavailable at train time, as they cannot be observed until after the model is deployed. For this reason, simple constraints of the form $y^f \geqslant y$ are ill-defined, and to regularize we must reason about outcome distributions $y^f \sim p(y|x^f)$, for $x^f \sim p^f$.

One approach might consider the average improvement, with $\mu^f = \mathbb{E}_{y^f \sim p(y|x^f)}[y^f]$, for a given $x^f \sim p^f$, and penalize the model whenever $\mu^f < y$, for example linearly using $\mu^f - y$.

[4] One possibility to relax covariate shift is to instead assume Lipschitzness, i.e., that $p(y|x)$ changes smoothly with changes to $p^f(x)$. This would affect the correctness of propensity weights, but can be accounted for by smoothly increasing uncertainty intervals (or reducing τ). We leave this to future work.

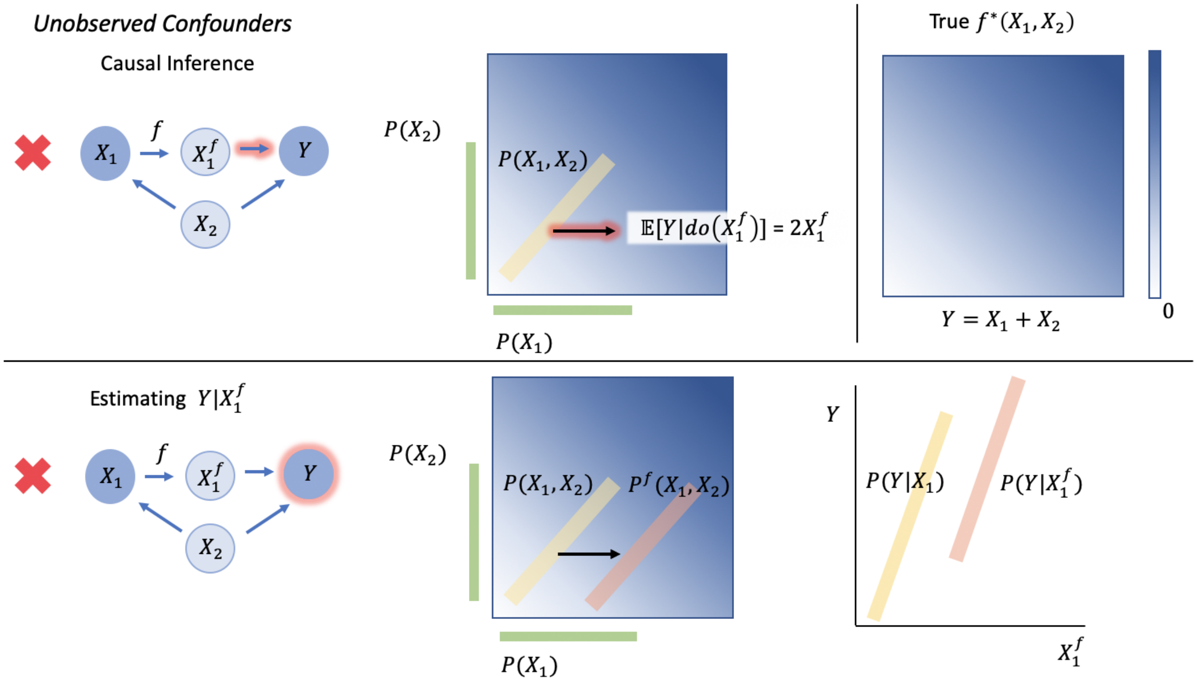


Figure 3.2: ***Unobserved Confounders*** *are a challenge for both causal inference and estimating outcomes under covariate shift. (Note: dark nodes are observed; light nodes are unobserved) For X_1 correlated with unobserved X_2 (here, let $X_1 = X_2$) that also affects $Y = X_1 + X_2$, it is impossible to isolate the effect of X_1, leading to an overestimation of the causal effect of X_1 on Y (top). Furthermore, when only X_1 is observed, the best estimate of $P(Y|X_1)$ is also $2X_1$. However, because this does not reflect the true causal mechanism behind Y, shifts in X_1 that hold X_2 constant (as when model subjects react only to published coefficients) will result in $P(Y|X_1^f)$ that does not align with the empirical $P(Y|X_1)$ (bottom). Thus samples from $P(Y|X_1)$ cannot be used to accurately estimate $P(Y|X_1^f)$.*

Concretely, μ^f must be estimated, and since f minimizes MSE, then $\hat{y}^f = f(x^f)$ is a plausible estimate of μ^f, giving:

$$\min_{f \in F} \mathbb{E}_{p(x,y)}[(\hat{y} - y)^2] + \lambda \mathbb{E}_{p(x,y)}[\hat{y}^f - y], \qquad \hat{y}^f = f(x^f), \qquad x^f = d^f(x), \tag{3.2}$$

where $\lambda \geqslant 0$ determines the relative importance of improvement over accuracy.

There are two issues with this approach. First, learning can result in an f that severely overfits in estimating μ, meaning that at train time the penalty term in the (empirical) objective will appear to be low whereas at test time its (expected) value will be high. This can happen, for example, when x^f is moved to a low-density region of $p(x)$ where f is unconstrained by the data and, if flexible enough, can artificially (and wrongly) signal improvement. To address this we use two decoupled models—one for predicting y on distribution p, and another for

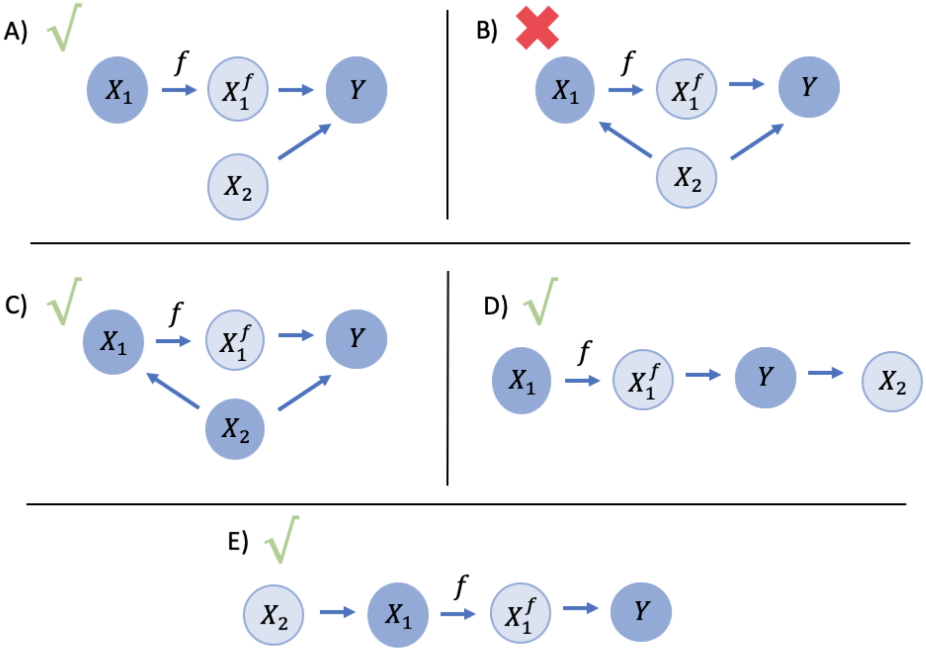


Figure 3.3: *Examples of graphical models that can and cannot accommodate the covariate shift assumption (Note: dark nodes are observed; light nodes are unobserved). In A), X_2 is causal and unobserved but uncorrelated with X_1^f. This allows for covariate shift to hold for $Y|X_1^f$, assuming as in Assumption 1 that unobserved feature X_2 will not change under the user decision function d^f induced by f. In B), as seen in Figure 3.2, X_2 is causal, unobserved, and correlated with X_1^f, violating "no unobserved confounders" and therefore violating covariate shift. In C), X_2 is causal and correlated with X_1^f but observed. This allows for covariate shift to hold for $Y|X_1^f, X_2$. In D), X_2 is downstream (a causal* child *of Y). This allows for covariate shift to hold for $Y|X_1^f$. In E), X_2 is non-confounding after conditioning on X_1. A, D, and E represent acceptable instances of unobserved non-confounders.*

handling y^f on distribution p^f.

Second, in many applications it may be unsafe to guarantee that improvement hold only on average per individual (e.g., heart attack risk, credit scores). To address this, we encourage f to improve outcomes with a certain degree of confidence τ, for $\tau > 0$, i.e., such that $\mathbb{P}[y^f \geqslant y] \geqslant \tau$ for a given (x, y) and induced x^f and thus $p(y^f|x^f)$. Importantly, while one source of uncertainty in y^f is $p(y|x^f)$, other sources of uncertainty exist, including those coming from insufficient data as well as model uncertainty. Our formulation is useful when additional sources of uncertainty are significant, such as when the model f leads to actions that place x^f in low-density regions of p.

In our method, we replace the average-case penalty in Eq. (3.2) with a *confidence-based penalty*:

$$\min_{f \in F} \mathbb{E}_{p(x,y)}[(\hat{y}-y)^2] + \lambda \mathbb{E}_{p(x,y)}[\mathbb{1}\{\mathbb{P}\left[y^f \geqslant y\right] < \tau\}], \qquad y^f \sim p(y|x^f), \qquad x^f = d^f(x) \quad (3.3)$$

where $\mathbb{1}\{A\} = 1$ if A is true, and 0 otherwise. In practice, $\mathbb{P}\left[y^f \geqslant y\right]$ is unknown, and must be estimated. For this, we make use of an *uncertainty model*, $g_\tau : \mathcal{X} \to \mathbb{R}^2$, $g_\tau \in G$, which we learn, and maps points $x^f \in \mathcal{X}$ to intervals $[\ell^f, u^f]$ that cover y^f with probability τ. With this, we replace the penalty term in Eq. (3.3) with the slightly more conservative $\mathbb{1}\{\ell^f < y\}$, and to make learning feasible we use the hinge loss $\max\{0, y - \ell^f\}$ as a convex surrogate.[5]

Definition 1 (Lookahead Learning Objective)**.** *For a given uncertainty model, g_τ, the empirical learning objective for model f on sample set $\mathcal{S} = \{x_i, y_i\}_{i=1}^m$ is:*

$$\min_{f \in F} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 + \lambda R(g_\tau; \mathcal{S}), \quad \textit{for} \;\; R(g_\tau; \mathcal{S}) = \sum_{i=1}^{m} \max\{0, y_i - \ell_i^f\}, \quad \ell_i^f = g_\tau(x_i^f), \qquad (3.4)$$

where $R(g_\tau; \mathcal{S})$ is the lookahead regularization *term.*

By anticipating how users decide, this penalizes models whose induced decisions do not improve outcomes at a sufficient rate (see Figure 3.1). The novelty in the regularization term is that it accounts for uncertainty in assessing improvement, and does so for points x^f that are out of distribution. If f pushes x^f towards regions of high uncertainty, then the interval $[\ell^f, u^f]$ is likely to be large, and f must make more "effort" to guarantee improvement, something that may come at some cost to in-distribution prediction accuracy. While the objective encodes the *rate* of decision improvement, increasing the $(1-\tau)$th percentile of outcomes can broadly be achieved either by reducing uncertainty for a given $\mathbb{E}[y^f]$ or by increasing $\mathbb{E}[y^f]$ for a given level of uncertainty, and so we will also see the *magnitude* of improvement increase in our experiments.

Note that the regularization term R depends both on f and g_τ—to determine x^f, and to

[5] The penalty is conservative in that it considers only one-sided uncertainty, i.e., $y^f < \ell^f$ and u^f is not used explicitly. Although open intervals suffice here, most methods for interval prediction consider closed intervals, and in this way our objective can support them. For symmetric intervals, τ simply becomes $\tau/2$.

determine ℓ^f given x^f, respectively. This justifies the need for the decoupling of f and g. Without this, uncertainty estimates are prone to overfit by artificially manipulating intervals to be higher than y, resulting in low penalization at train time without actual improvement (see Figure 3.4 (right)).

3.2.2 Estimating uncertainty

The usefulness of lookahead regularization relies on the ability of the uncertainty model g to correctly capture the various kinds of uncertainties about the outcome value for the perturbed points. This can be difficult because uncertainty estimates are needed for out-of-distribution points x^f.

Fortunately, for a given f the counterfactual distribution p^f is known (by Assumption 1), and we can use the covariate transform associated with the decision to construct sample set $\mathcal{S}^f = \{x_i^f\}_{i=1}^m$. Even without labels for $\mathcal{S}^f$, estimating the uncertainty model g is now a problem of *learning under covariate shift*, where the test distribution p^f can differ from the training distribution p. In particular, we are interested in learning uncertainty intervals that provide good coverage. There are many approaches to learning under covariate shift. Here we describe the simple and popular method of importance weighting, or inverse propensity weighting [Shi00]. For a loss function $\ell(g) = \ell(y, g(x))$, we would like to minimize $\mathbb{E}_{p^f(x,y)}[\ell(g)]$. Let $w(x) = p^f(x)/p(x)$, then by the covariate shift assumption:

$$\mathbb{E}_{p^f(x,y)}[\ell(g)] = \int \ell(g)\, \mathrm{d}p^f(x)\, \mathrm{d}p(y|x) = \int \frac{p^f(x)}{p(x)} \ell(g)\, \mathrm{d}p(x)\, \mathrm{d}p(y|x) = \mathbb{E}_{p(x,y)}[w(x)\ell(g)].$$

Hence, training g with points sampled from distribution p but weighted by w will result in an uncertainty model that is optimized for the counterfactual distribution p^f. In practice, w is itself unknown, but many methods exist for learning an approximate model $\hat{w}(x) \approx w(x)$ using sample sets $\mathcal{S}$ and $\mathcal{S}^f$ (e.g. [KHS09]). To remain within our discriminative approach, we follow [BBS09] and train a logistic regression model $h : \mathcal{X} \to [0, 1]$, $h \in H$, to differentiate between points $\tilde{x} \in \mathcal{S}$ (labeled $\tilde{y} = 0$) and $\tilde{x} \in \mathcal{S}^f$ (labeled $\tilde{y} = 1$) and set weights to $\hat{w}(x) = e^{h(x)}$. As we are interested in training g to gain a coverage guarantee, we define $\ell(y, g(x)) = \mathbb{1}\{y \notin [\ell, u]\}$ as in [RMYT18].

3.2.3 Algorithm

All the elements in the framework— the predictive model f, the uncertainty model g, and the propensity model h —are interdependent. Specifically, optimizing f in Eq. (3.4) requires intervals from g; learning g requires weights from h; and h is trained on $\mathcal{S}^f$ which is in turn determined by f. The algorithm therefore alternates between optimizing each of these components while keeping the others fixed. At round t, $f^{(t)}$ is optimized with intervals $[\ell_i^f, u_i^f] = g^{(t-1)}(x_i^f)$, $g^{(t)}$ is trained using weights $w_i = h^{(t)}(x_i)$, and $h^{(t)}$ is trained using points x_i^f as determined by $f^{(t)}$. The procedure is initialized by training $f^{(0)}$ without the lookahead term R. For training g and h, weights $w_i = \hat{w}(x_i)$ and points $\{x_i^f\}_{i=1}^m$, respectively, can be computed and plugged into the objective. Training f with Eq. (3.4), however, requires access to the *function* g, since during optimization, the lower bounds ℓ^f must be evaluated for points x^f that vary as updates to f are made. Hence, to optimize f with gradient methods, we use an uncertainty model g that is differentiable, so that gradients can pass through the model (while keeping the parameters of the uncertainty model fixed). Furthermore, since gradients must also pass through x^f (which includes ∇_f), we require that f be twice-differentiable.

In the experiments we consider two methods for learning g:

1. Bootstrapping [ET94], where a collection of models $\{g^{(j)}\}_{j=1}^k$ is trained, each on a subsampled dataset, and combined to produce a single interval model g, and

2. Quantile regression [KH01], where models $g^{(\ell)}, g^{(u)}$ are discriminatively trained to estimate the τ and $1 - \tau$ quantiles, respectively, of the counterfactual target distribution $p^f(y|x^f)$.

3.3 Experiments

In this section, I evaluate the approach in three experiments of increasing complexity and scale, where the first is synthetic and the latter two use real data. Because the goal of regularization is to balance accuracy with decision quality, we will be interested in understanding the attainable frontier of accuracy vs. improvement in outcomes. For our method, this will mostly be controlled by varying lookahead regularization parameter, $\lambda \geqslant 0$. In all experiments we measure predictive accuracy with root mean squared error (RMSE), and decision quality in two ways: *mean improvement rate* $\mathbb{E}[\mathbb{1}\{y_i^f > y_i\}]$ (corresponding to the the regularization term in Eq. (3.3)), and *mean improvement magnitude* $\mathbb{E}[y_i^f - y_i]$ (corresponding to its convex proxy in Eq. (3.4)).[6]

To evaluate the approach, we need a means for evaluating counterfactual outcomes y^f for decisions x^f. Therefore, and similarly to Shavit and Moses [SM19], we make use of an inferred 'ground-truth' function f^* to test decision improvement, assuming $y^f = f^*(x^f)$. Model f^* is trained on the entirety of the data. By optimizing f^* for RMSE, we think of this as estimating the conditional mean of $p(y|x)$, with the data labels as noisy observations. To make for an interesting experiment, we learn f^* from a function class F^* that is more expressive than F or G. The sample set $\mathcal{S}$ will contain a small and possibly biased subsample of the data, which we call the 'active set', and that plays the role of a representative sample from p. This setup allows us not only to evaluate improvement, but also to experiment with the effects of different sample sets.

3.3.1 Experiment 1: Quadratic curves

For a simple setting, we explore the effects of regularized and unregularized learning on decision quality in a stylized setting using unidimensional quadratic curves. Let $f^*(x) = -x^2$, and assume $y = f(x) + \varepsilon$ where ε is independently, normally distributed. By varying the decision model step-size η, we explore three conditions: one where a naïve approach works well, one where it fails but regularization helps, and one where regularization also fails.

In Figure 3.4 (top left), η is small, and the x^f points stay within the high certainty region of p. Here, the baseline works well, giving both a good fit and effective decisions, and the regularization term in the lookahead objective remains inactive. In Figure 3.4 (top right), η is larger. Here, the baseline model pushes x^f points to a region where y^f values are low. Meanwhile, the lookahead model, by incorporating into the objective the decision model and

[6] Our code can be found at https://github.com/papushado/lookahead.

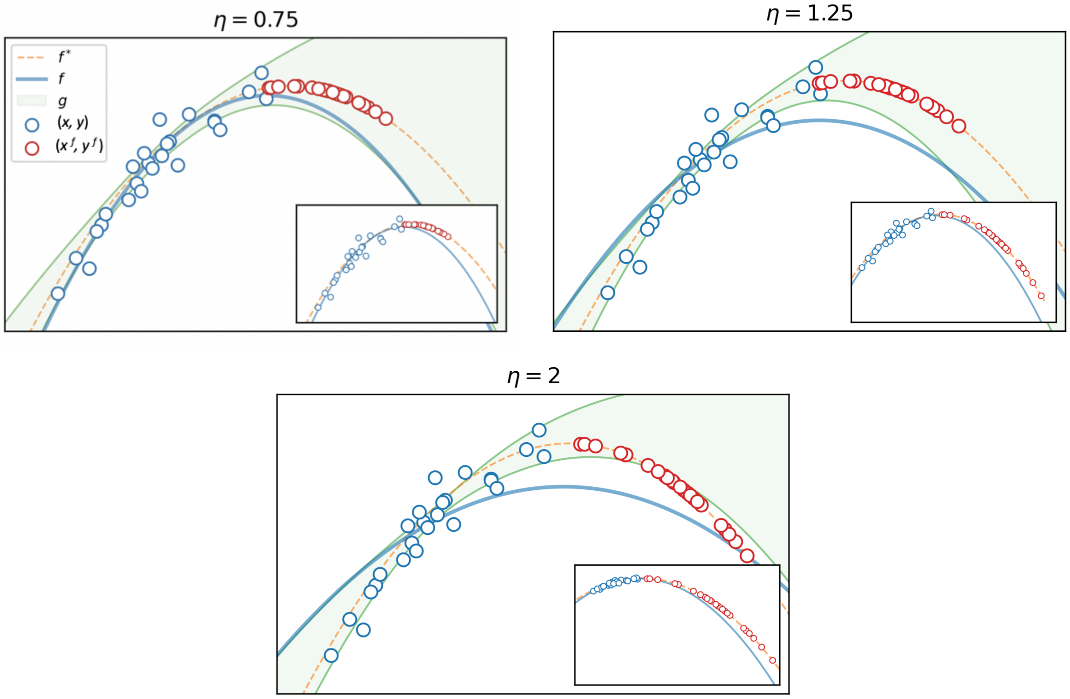


Figure 3.4: *Results for the synthetic experiment comparing lookahead (main plots) to a baseline model (inlays; note the change in scale across plots). In both models, decisions move points x over the peak. Under the baseline model, as η increases, decision outcomes y^f worsen. Lookahead corrects for this, and at a small cost to accuracy (on p) ensures good decision outcomes (on p^f, with sufficient overlap).*

estimating uncertainty surrounding y^f, is able to adjust the model to induce good decisions with some reduction in accuracy. In Figure 3.4 (bottom), η is large. Here, the x^f points are pushed far into areas of high uncertainty. The success of lookahead relies on the successful construction of intervals at p^f through the successful estimation of w, and may fail if p and p^f differ considerably, as is the case here.

3.3.2 Experiment 2: Wine quality

The second experiment focuses on wine quality using the wine dataset from the UCI data repository [DG17]. The wine in the data set has 13 features, most of which correlate linearly with quality y, but two of which (alcohol and malic acid) have a non-linear U-shaped or inverse-U shaped relationship with y. For the ground truth model, we set $f^*(x) = \sum_i \theta_i x_i + \sum_i \theta'_i x_i^2$ (RMSE $= 0.2$, $y \in [0, 3]$) so that it captures these nonlinearities. To better demonstrate the capabilities of our framework, we sample points into the active set non-uniformly by thresholding on the non-linear features. The active set includes $\sim$30% of the data, and is further split 75-25 into a train set used for learning and tuning and a held-out test set used for final evaluation.

For the predictive model, our focus here is on linear models. The baseline includes a linear f_{base} trained with ℓ_2 regularization (i.e., Ridge Regression) with regularization coefficient $\alpha \geqslant 0$. Our lookahead model includes a linear f_{look} trained with lookahead regularization (Eq. (3.4)) with regularization coefficient $\lambda \geqslant 0$. In some cases we will add to the objective an additional ℓ_2 term, so that for a fixed α, setting $\lambda = 0$ recovers the baseline model. Lookahead was trained for 10 rounds and the baseline with a matching number of overall epochs. The uncertainty model g uses residuals-based bootstrapping with 20 linear sub-models. The propensity model h is also linear. We consider two settings: one where all features (i.e., wine attributes) are mutable and using decision step-size $\eta = 0.5$, and another where only a subset of the features are mutable and using step-size $\eta = 2$.

Full mutability. Figure 3.5 (top left) presents the frontier of accuracy vs. improvement on the test set when all features are mutable. The baseline and lookahead models coincide when $\alpha = \lambda = 0$. For the baseline, as α increases, predictive performance (RMSE) displays a typical learning curve with accuracy improving until reaching an optimum at some intermediate value of α. Improvement, however, monotonically decreases with α, and is highest with no regularization ($\alpha = 0$). This is because in this setting, gradients of f_{base} induce reasonably good decisions: f_{base} is able to approximately recover the dominant linear coefficients of f^*, and shrinkage due to higher ℓ_2 penalization reduces the magnitude of the (typically positive, on average) change. With lookahead, increasing λ leads to better decisions, but at the cost of higher (albeit sublinear) RMSE. The initial improvement rate at $\lambda = 0$ is high, but lookahead and ℓ_2 penalties have opposing effects on the model. Here, improvement is achieved by (and likely requires) increasing the size of the coefficients of linear model, f_{look}. We see that f_{look} learns to do this in an efficient way, as compared to a naïve scaling of the predictively-optimal f_{base}.

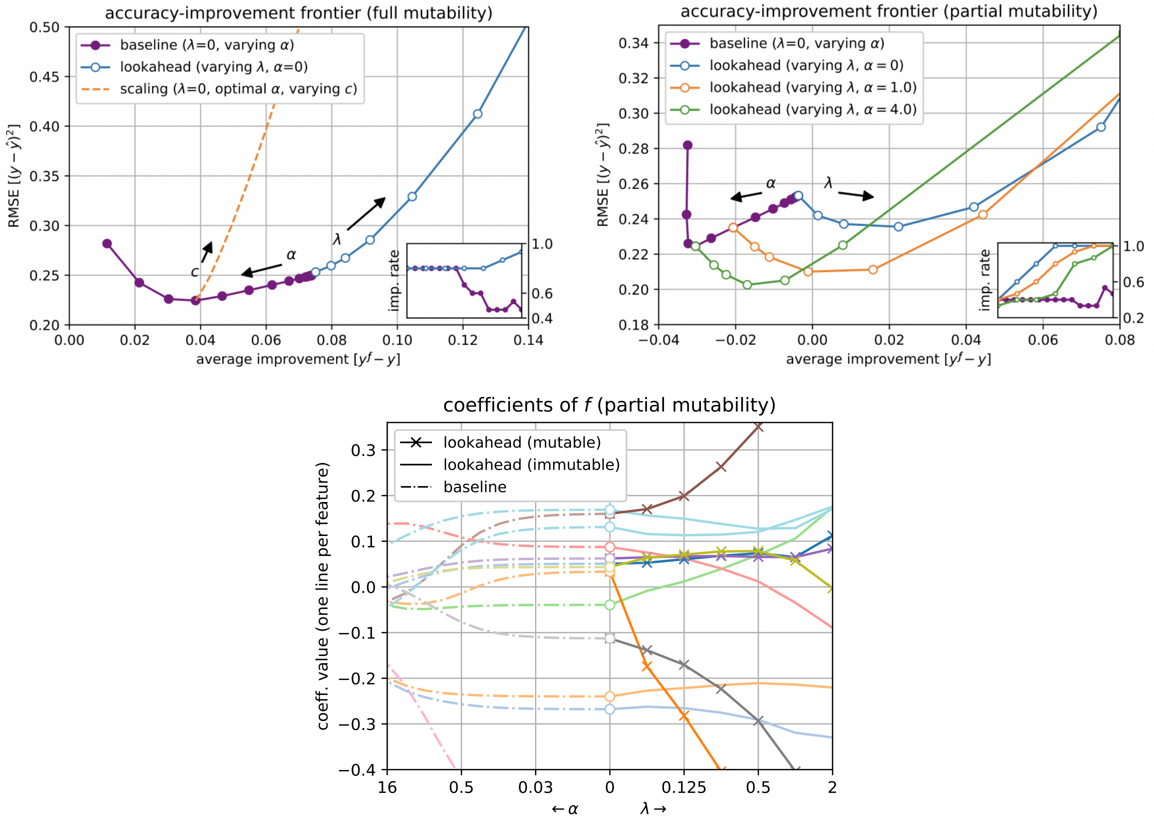


Figure 3.5: *Results for the wine experiment. Tradeoff in accuracy and improvement under full mutability (left) and partial mutability (center), for which model coefficients are also shown (right).*

Partial mutability. Figure 3.5 (top right) presents the frontier of accuracy vs. improvement when only a subset of the features are mutable (note that this effects the scale of possible improvement). The baseline presents a similar behavior to the fully-mutable setting, but with the optimal predictive model inducing a negative improvement. Here we consider lookahead with various degrees of additional ℓ_2 regularization. When $\alpha = \lambda = 0$, the models again coincide. However, for larger λ, significant improvement can be gained with very little or no loss in RMSE, while moderate λ values improve both decisions and accuracy. This holds for various values of α, and setting α to the optimal value of f_{base} results in lookahead dominating the trade-off curve for all observed λ. Improvement is reflected in magnitude and rate. Improvement rate (see Figure 3.5 (top right) inlay) rises quickly from the baseline's $\sim 40\%$ to an optimal 100%, showing how lookahead learns models that lead to safe decisions.

Figure 3.5 (bottom) shows how the coefficients of f_{base} and f_{look} change as α and λ increase, respectively (for lookahead $\alpha = 0$). As can be seen, lookahead works by making substantial changes to mutable coefficients, sometimes reversing their sign, with milder changes to immutable coefficients. Lookahead achieves improvement by capitalizing on its freedom to learn a useful direction of improvement within the mutable subspace, while compensating for the possible loss in accuracy through mild changes in the immutable subspace.

3.3.3 Experiment 3: Diabetes

The final experiment focuses on the prediction of diabetes progression using the diabetes dataset[7] [Efr+04]. The dataset has 10 features describing various patient attributes. We consider two features as mutable: BMI and T-cell count (marked as 's1'). While both display a similar (although reversed) linear relationship with y, feature s1 is much noisier. The setup is as in wine but with two differences: to capture nonlinearities we set f^* to be a flexible generalized additive model (GAM) with splines of degree 10 (RMSE $= 0.15$), and train and test sets are sampled uniformly from the data. We normalize y to $[0, 1]$ and set $\eta = 5$. Appendix B.3.4 includes a sensitivity analysis to learning with misspecified η.

Figure 3.6 (top) presents the accuracy-improvement frontier for linear f and bootstrapped linear g. Results show a similar trend to the wine experiment, with lookahead providing improved outcomes (both rate and magnitude) while preserving predictive accuracy. Here, lookahead improves results by learning to increase the coefficient of s1, while adjusting other coefficients to maintain reasonable uncertainty. The baseline fails to utilize s1 for improvement since from a predictive perspective there is little value in placing weight on s1.

When f is linear, decisions are uniform across the population in that $\nabla_{f_\theta}(x) = \theta$ is independent of x. To explore individualized actions, we also consider a setting where f is a more flexible quadratic model (i.e., linear in x and x^2) in which gradients depend on x and uncertainty is estimated using quantile regression. Figure 3.6 (bottom left) shows the data as projected onto the subspace $(x_{\text{BMI}}, x_{\text{s1}})$, with color indicating outcome values $f^*(x)$, interpolated within this subspace. As can be seen, the mapping $x \mapsto x^f$ due to f_{look} generally improves

[7] https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html

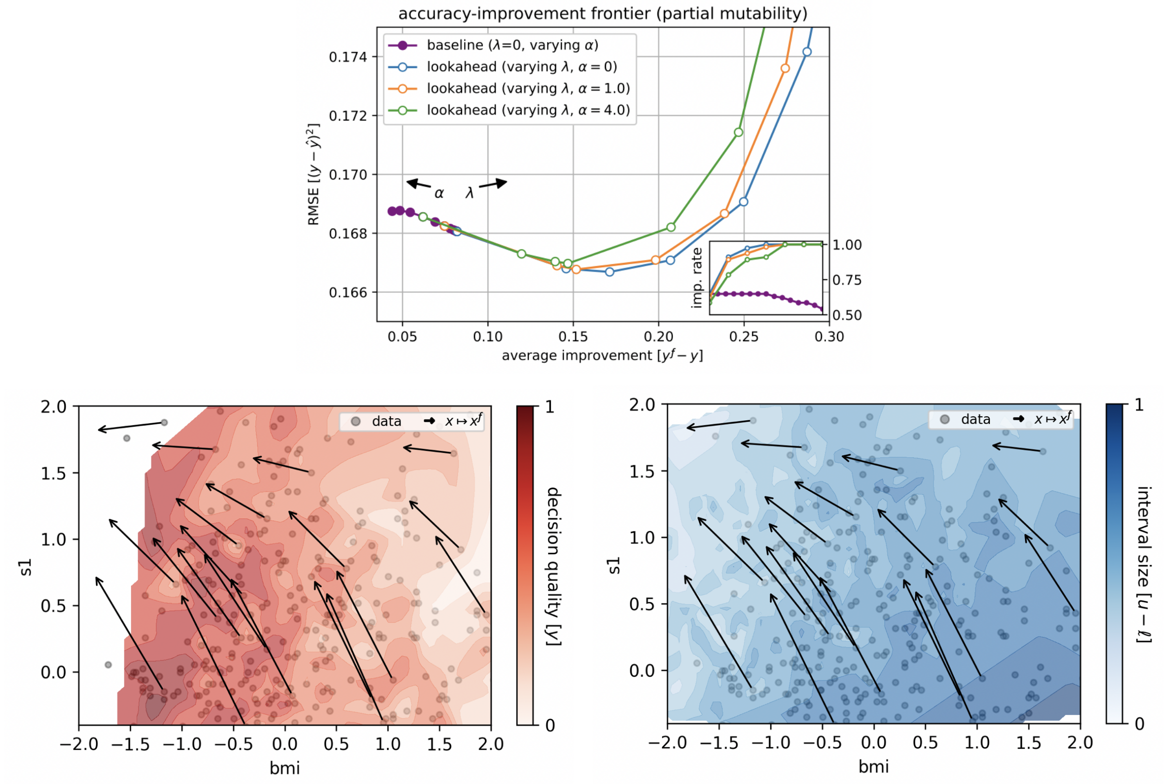


Figure 3.6: *Results for the diabetes experiment. Tradeoff in accuracy and improvement under linear f with partial mutability (left), visualization of shift $p \to p^f$ with non-linear f to regions of higher decision quality (center), and regions of lower uncertainty (right).*

outcomes. The plot reveals that, had we had knowledge of $f^*(x)$, uniformly decreasing BMI would also have improved outcomes, and this is in fact the strategy invoked by the linear f_{base}. But decisions must be made based on the sample set, and so uncertainty must be taken into account. Figure 3.6 (bottom right) shows a similar plot but with color indicating uncertainty estimates as measured by the interval sizes given by g. The plot shows that decisions are directed towards regions of lower uncertainty (i.e., approximately following the negative gradients of the uncertainty slope), showing how lookahead successfully utilizes these uncertainties to adjust the predictive model f_{look}.

3.4 Discussion

Given the extensive use of machine learning across an ever-growing range of applications, it is appropriate to assume that predictive models will remain in widespread use, and that at the same time, and despite well-understood concerns, users will continue to act upon them. In line with this, the goal with this work has been to develop a machine learning framework that accounts for decision making by users but remains fully within the discriminative framing of statistical machine learning. The lookahead regularization framework that we have proposed augments existing machine learning methodologies with a component that promotes good decisions. I have demonstrated the utility of this approach across three different experiments, one on synthetic data, one on predicting and deciding about wine, and one on predicting and deciding in regard to diabetes progression. I hope that this work will inspire continued research in the machine learning community that embraces predictive modeling while also being cognizant of the ways in which our models are used.

Future work is needed to understand how robust the method is to different violations of Assumptions 1 and 2. In particular, I am interested in finding settings where it may be possible to directly observe the changes that users make in response to deployed models, and to use these observations to evaluate how much decisions may vary from those assumed both in this chapter and in related literature, such as strategic classification and recourse. Additionally, the assumption of covariate shift may often be violated to some degree. Understanding of such violations may take the form either of assessing theoretical relaxations, such as assuming Lipschitzness rather than invariance of $p(y|x)$, or evaluating empirical datasets with confounding variables withheld from the model.

Broader Impact

In our work, the learning objective was designed to align with and support the possible use of a predictive model to drive decisions by users. Responsible and transparent deployment of models with "lookahead-like" regularization components should avoid the kinds of mistakes that can be made when predictive methods are conflated with causally valid methods.

At the same time, this work makes a strong simplifying assumption, that of covariate shift, which requires that the relationship between covariates and outcome variables is invariant as decisions are made and the feature distribution changes. This strong assumption is made to ensure validity for the lookahead regularization, since we need to be able to perform inference about counterfactual observations. As discussed by Mueller et al. [Mue+17] and Peters, Bühlmann, and Meinshausen [PBM16], there exist real-world tasks (e.g. writing improvement, gene perturbation) that reasonably satisfy this assumption, and yet at the same time, other tasks— notably those with unobserved confounders —where this assumption would be violated. Moreover, this assumption is not testable on the observational data. This, along with the need to make an assumption about the user decision model, means that an application of the method proposed here should be done with care and will require appropriate domain knowledge to understand whether or not the assumptions are plausible.

Furthermore, the validity of the interval estimates requires that any assumptions for the particular interval model used are satisfied and that weights w provide a reasonable estimation of p^f/p. In particular, fitting propensity scores for a distribution p^f that has little to no overlap with p (see Figure 3.4) may result in underestimating the possibility of bad outcomes.

If used carefully and successfully, then lookahead regularization provides safety and protects against the misuse of a model. If used in a domain for which the assumptions fail to hold then the framework could make things worse, by trading accuracy for an incorrect view of user decisions and the effect of these decisions on outcomes.

We also caution against any specific interpretation of the application of the model to the wine and diabetes data sets. Model misspecification of f^* can result in arbitrarily bad outcomes, and estimating f^* in any high-stakes setting requires substantial domain knowledge and should err on the side of caution. The data sets are used to illustrate the kinds of results that should be available when the method is correctly applied to a domain of interest.

Chapter 4

Vulnerabilities in Popular Explainability Methods

4.1 Introduction

Owing to the success of machine learning (ML) models, there has been an increasing interest in leveraging these models to aid decision makers (e.g., doctors, judges) in critical domains such as healthcare and criminal justice. The successful adoption of these models in domain-specific applications relies heavily on how well decision makers are able to understand and trust their functionality [DVK17; Lip16]. Only if decision makers have a clear understanding of the model behavior can they diagnose errors and potential biases in these models and decide when and how much to rely on them. However, the proprietary nature and increasing complexity of machine learning models makes it challenging for domain experts to understand these complex *black boxes*, motivating the need for tools that can explain them in a faithful and interpretable manner.

As a result, there has been a recent surge in post hoc techniques for explaining black box models in a human interpretable manner. One of the primary uses of such explanations is to help domain experts detect discriminatory biases in black box models [Tan+18; Kim+18]. Among the most prominent of these techniques are *local, model-agnostic* methods that focus on explaining individual predictions of a given black box classifier, including LIME [RSG16] and SHAP [LL17]. These methods estimate the contribution of individual features towards a specific prediction by generating perturbations of a given instance in the data and observing the effect of these perturbations on the output of the black-box classifier. Due to their generality, these methods have been used to explain a number of classifiers, such as neural networks and complex ensemble models, and in various domains ranging from law, medicine, finance, and science [EAMS19; Ibr+19; WGH16]. However, there has been little analysis of the reliability and robustness of these explanation techniques, especially in the adversarial setting, making their utility for critical applications unclear.

In this work, we demonstrate significant vulnerabilities in post hoc explanation techniques that can be exploited by an adversary to generate classifiers whose post hoc explanations can be arbitrarily controlled. More specifically, we develop a novel framework that can effectively mask the discriminatory biases of any black box classifier. Our approach exploits the fact that post hoc explanation techniques such as LIME and SHAP are perturbation-based, to create a *scaffolding* around any given biased black box classifier in such a way that its predictions on the input data distribution remain biased, but its behavior on the perturbed data points is controlled to make the post hoc explanations look completely innocuous. In particular, using our framework, we generate highly discriminatory scaffolded classifiers (that, for example, *only* use race to make their decisions) whose post hoc explanations (generated by LIME and SHAP) effectively hide their discriminatory biases, making them look unobjectionable.

We evaluate the effectiveness of the proposed framework on multiple real world datasets — COMPAS [Lar+16], Communities and Crime [Red11], and German loan lending [DG17]. For each dataset, we craft classifiers that heavily discriminate based on protected attributes such as race (demographic parity ratio = 0), and show that our framework can effectively hide their biases. In particular, our results show that the explanations of these classifiers generated using off-the-shelf implementations of LIME and SHAP often do not flag *any* of the relevant sensitive attributes (e.g., race) as important features of the classifier for the test instances, thus demonstrating that the adversarial classifiers successfully fooled these explanation methods. These results suggest that it is possible for malicious actors to craft adversarial classifiers that are highly discriminatory, but can effectively fool existing post hoc explanation techniques.

This further establishes that existing post hoc explanation techniques are not sufficiently robust for ascertaining discriminatory behavior of classifiers in sensitive applications.

4.2 Building Adversarial Classifiers to Fool Explanation Techniques

In this section, we discuss our framework for constructing adversarial classifiers (*scaffoldings*) that can fool post hoc explanation techniques that rely on input perturbations. We first provide a detailed overview of popular post hoc explanation techniques, namely, LIME [RSG16] and SHAP [LL17], and then present our framework for constructing adversarial classifiers.

4.2.1 Background: LIME and SHAP

While simpler classes of models (such as linear models and decision trees) are often readily understood by humans, the same is not true for complex models (e.g., ensemble methods, deep neural networks). Even when model examiners have full access to architectures and parameters, these complex models are essentially black boxes for all practical purposes. This obscurity is often further exacerbated when model examiners have only query access to models, for example due to models being protected due to proprietary knowledge. One way to better *understand* the behavior of such classifiers is to build simpler *explanation models* that are interpretable approximations of these black boxes. This approach to model understanding is known as *post hoc* explanation, as the explanation methods are applied after model training. This is in contrast to *intrinsically interpretable models* which are trained with interpretability constraints (e.g. on model class and complexity) in place.

To this end, several techniques have been proposed in the existing literature. LIME [RSG16] and SHAP [LL17][1] are two popular *model-agnostic*, *local explanation* approaches designed to explain any given black box classifier. These methods explain individual predictions of any classifier in an interpretable and faithful manner, by learning an interpretable model (e.g., linear model) locally around each prediction. The intuition behind LIME and SHAP is the following:

[1] Here we focus specifically on KernelSHAP. While Lundberg and Lee [LL17] provide several alternative implementations, only KernelSHAP is applicable to any black box model.

while complex black box models typically exhibit highly non-linear decision boundaries globally (and are therefore harder to explain *overall*), the behavior of these models tends to be much less complex within a smaller region of the feature space. Therefore, simple, human interpretable models such as linear models are often a useful description of model behavior in a particular input locality. Specifically, LIME and SHAP use the coefficients on local linear models to estimate feature attributions for individual instances, which capture the *contribution* of each feature to the black box prediction. Below, we provide some details of these approaches, while also highlighting how they relate to each other.

Let $\mathcal{D}$ denote the input dataset of N data points, $\mathcal{D} = (\mathcal{X}, \boldsymbol{y}) = \{(x_1, y_1), (x_2, y_2) \cdots (x_N, y_N)\}$ where x_i is a vector that captures the feature values of data point i, and y_i is the corresponding class label. Let there be M features in the dataset $\mathcal{D}$ and let $\mathcal{C}$ denote the set of class labels in $\mathcal{D}$ i.e., $y_i \in \mathcal{C}$. Let f denote the black box classifier that takes a data point as input and returns a class label i.e., $f(x_i) \in \mathcal{C}$. The goal here is to explain f in an interpretable and faithful manner. Note that neither LIME nor SHAP assume any knowledge about the internal workings of f. Let g denote an explanation model that we intend to learn to explain f. For LIME and SHAP, $g \in G$ where G is the class of linear models.

$$g(x) = \phi_0 + \sum_{i=1}^{M} \phi_i x_i,\ \phi_i \in \mathbb{R} \tag{4.1}$$

Let the complexity of the explanation g be denoted as $\Omega(g)$ (the complexity of a linear model can be measured as the number of non-zero weights), and let $\pi_x(x')$ denote a proximity measure between inputs x and x', used to define the vicinity (neighborhood) around x. With all this notation in place, the objective function for both LIME and SHAP is crafted to generate an explanation that: (1) approximates the behavior of the black box accurately within the vicinity of x, and (2) achieves lower complexity and is thereby interpretable. In particular, the objective function is

$$\underset{g \in \mathcal{G}}{\operatorname{argmin}}\, L(f, g, \pi_x) + \Omega(g), \tag{4.2}$$

where the loss function L is defined as

$$L(f, g, \pi_x) = \sum_{x' \in X'} [f(x') - g(x')]^2 \pi_x(x')$$

, where X' is the set of inputs constituting the neighborhood of x. Often, these instances are sampled by drawing some subset of features of x uniformly at random and setting the other features to be *missing*.[2] Missing features can be set exactly equal to zero or to some other context-specific definition of zero; e.g. in image classification, setting all pixel values to zero may be less appropriate than setting them equal to the average pixel value.

The primary difference between LIME and SHAP lies in how Ω and π_x are chosen. In LIME, these functions are defined heuristically: $\Omega(g)$ is the number of non-zero weights in the linear model and $\pi_x(x')$ is defined using cosine or $l2$ distance.

On the other hand, KernelSHAP grounds these definitions in a game theoretic method used to value the contributions of individual players in a coalition, called *Shapley values*. [Sha51]. SHAP guarantees that explanations satisfy three desired properties which relate to the Shapley value axioms:

1. **Local Accuracy** states that $g(x) = f(x)$; that is, the additive contributions of all features at a point x, as represented by linear model g, must add up to the function value $f(x)$. This is exactly the *efficiency* axiom of the Shapley values, which states that the contributions of individual players must sum to the total proceeds of a game.

2. **Missingness** states that a feature that is missing must be constrained to have zero contribution. This property is added to the Shapley axioms to adapt to the model valuation setting.

3. **Consistency** ensures a monotonicity property over feature contributions ϕ_i: if the function f to be explained changes to some f' such that the marginal value of including a given feature i (as opposed to setting that feature to be missing) increases or stays the same regardless of how many other features are included, the contribution ϕ'_i of that feature

[2] The standard implementation of LIME varies from this behavior for continuous features, perturbing inputs according to a normal distribution rather than sampling uniformly at random, see Section 4.3

> must not decrease relative to ϕ_i of the original function. That is, for two functions f, f' and S any subset of features taken to be nonzero:
>
> $$f(S \cup i) - f(S) \geqslant f'(S \cup i) - f'(S) \;\Rightarrow\; \phi_i(f) \geqslant \phi_i(f')$$
>
> This property entails the other three axioms of the Shapley values: *symmetry, null player*, and *linearity*. *Symmetry* states that two players who contribute identically receive identical values. *Null player* states a player who always contributes nothing receives zero value. *Linearity* is a combination of additivity, which suggests that the value of a player in any two games is the sum of their individual values in those games, and scalar multiplication, which suggests that the units in which utility is measured does not affect relative contributions. In particular, Young [You85] demonstrated that monotonicity enforces the linearity and null player axioms, and Lundberg and Lee [LL17] show that in the model evaluation setting, it also implies symmetry.

Lundberg and Lee [LL17] show that the unique choices of Ω and π to satisfy these properties are:

$$\Omega(g) = 0 \;\text{ and }\; \pi_x(x') = \frac{(M-1)}{\binom{M}{|x'|}|x'|(M-|x'|)},$$

where $|x'|$ is the number of non-missing (non-zero) elements in x'. Further, for x' that represent the presence of a subset S of features these weights relate to the classical expression of Shapley values

$$\phi_i = \sum_{x' \in X'} \frac{(M-|x'|)!(|x'|-1)!}{M!}\left[f(x') - f(x' \backslash i)\right],$$

as Lundberg and Lee [LL17] prove that this is exactly the result in the analytical solution to the weighted linear regression

$$\phi = (X^{\intercal}\Pi X)^{-1}X^{\intercal}\Pi y$$

where X is a $2^M \times M$ matrix with rows representing all possible binary feature inclusions and y is a 2^M vector of corresponding function values.

More details about the intuition behind the definitions of these functions and their computation can be found in Ribeiro, Singh, and Guestrin [RSG16] and Lundberg and Lee [LL17].

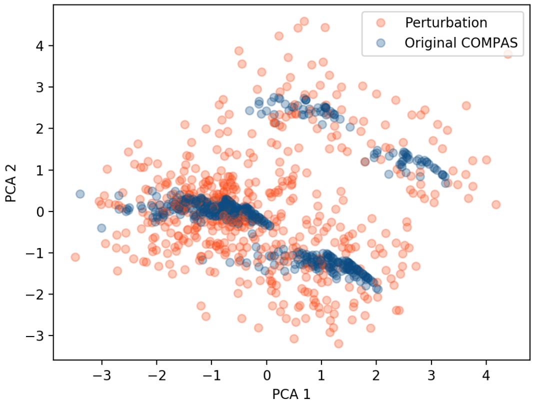


Figure 4.1: *PCA applied to the COMPAS dataset (blue) as well as its LIME style perturbations (red). Even in this low-dimensional space, we can see that data points generated via perturbations are distributed very differently from instances in the COMPAS data. In this paper, we exploit this difference to craft adversarial classifiers.*

4.2.2 Proposed Framework

In this section, we discuss our framework in detail. First, we discuss some preliminary details about our set up. Then, we discuss the intuition behind our approach. Lastly, we present the technical details of our approach along with a discussion of some of our design choices and implementation details.

Preliminaries

Setting: Assume that there is an adversary with an incentive to deploy a biased classifier f for making a critical decision (e.g., parole, bail, credit) in the real world. The adversary must provide black box access to customers and regulators [Reg16], who may use post hoc explanation techniques to better understand f and determine if f is ready to be used in the real world. If customers and regulators detect that f is biased, they are not likely to approve it for deployment. The goal of the adversary is to fool post hoc explanation techniques and hide the underlying biases of f.

Input: The adversary provides the following to our framework: 1) the biased classifier f which they intend to deploy in the real world and, 2) an input dataset $\mathcal{X}$ that is sampled from the real world input data distribution $\mathcal{X}_{dist}$ on which f will be applied. Note that neither our framework nor the adversary has access to $\mathcal{X}_{dist}$.

Output: The output of our framework will be a scaffolded classifier e (referred to as the *adversarial classifier* henceforth) that behaves exactly like f when making predictions on instances sampled from $\mathcal{X}_{dist}$, but will not reveal the underlying biases of f when probed with leading post hoc explanation techniques such as LIME and SHAP.

Intuition As discussed in the previous section, LIME and SHAP (and several other post hoc explanation techniques) explain individual predictions of a given black box model by constructing local interpretable approximations (e.g., linear models). Each such local approximation is designed to capture the behavior of the black box within the neighborhood of a given data point. These neighborhoods constitute synthetic data points generated by perturbing features of individual instances in the input data. However, instances generated using such perturbations could potentially be off-manifold or out-of-distribution (OOD) [MRW19].

To better understand the nature of the synthetic data points generated via perturbations, we carried out the following experiment. First, we perturb input instances using the approach employed by LIME (See previous section). We then run principal component analysis (PCA) on the combined dataset containing original instances as well as the perturbed instances, and reduce the dimensionality to 2. As we can see from Figure 4.1, the synthetic data points generated from input perturbations are distributed significantly differently from the instances in the input data. This result indicates that detecting whether or not a data point is a result of a perturbation is not a challenging task, and thus approaches that rely heavily on these perturbations, such as LIME, can be *gamed*.

This intuition underlies our proposed approach. By differentiating between data points coming from the input distribution and instances generated via perturbation, an adversary can create an adversarial classifier (*scaffolding*) that behaves like the original classifier (and in particular may be extremely discriminatory) on the input data points, but behaves arbitrarily differently (looks unbiased and *fair*) on the perturbed instances, thus effectively fooling LIME or SHAP into generating innocuous explanations.

Next, we formalize this intuition and explain our framework for building adversarial classifiers that can fool explanation techniques.

Building Adversarial Classifiers Let f be the biased classifier described earlier. The adversary would like to design a framework such that if and when end users generate explanations of this black box, post hoc techniques can be fooled into thinking that the model is innocuous. Recall that the real world data on which this classifier is likely to be applied follows a distribution $\mathcal{X}_{dist}$, and $\mathcal{X}$ is a set of N data points sampled from this distribution that the adversary has access to. To fool the post hoc techniques, the adversary could design an adversarial classifier that exhibits biased behavior on instances sampled from $\mathcal{X}_{dist}$, and remains unbiased on instances that do not come from $\mathcal{X}_{dist}$. Since the feature importances output by LIME and SHAP rely heavily on perturbed instances (which may typically be OOD samples, e.g. Figure 4.1), the resulting explanations will make the classifier designed by the adversary look innocuous.

Assuming ψ is a *unbiased* classifier (e.g., makes predictions based on features that are uncorrelated with sensitive attributes), the adversarial classifier e takes the following form:

$$e(x) = \begin{cases} f(x), & \text{if } x \in \mathcal{X}_{dist} \\ \psi(x), & \text{otherwise} \end{cases}$$

In order to create such a classifier, we need to be able to decide whether a given data point x comes from $\mathcal{X}_{dist}$ or not.

Detecting OOD Samples To build a classifier `is_OOD` that detects if a given data point is an *out-of-distribution* (OOD) sample ($\texttt{is\_OOD}(x) = \texttt{True}$) or not, we construct a new dataset from the instances in $\mathcal{X}$. Specifically, we *perturb* (more details in Experimental Evaluation) each instance x in $\mathcal{X}$ to generate a new instance x_p. Let us denote the set of all the instances generated via perturbation as $\mathcal{X}_p$. The instances in $\mathcal{X}$ are then assigned the class label `False` indicating that they are not OOD samples, while the instances in $\mathcal{X}_p$ are assigned the class label `True` (indicating that they are OOD samples) unless they are already in $\mathcal{X}$. We then train an off-the-shelf classifier on the combined dataset $\mathcal{X} \cup \mathcal{X}_p$ and their corresponding class labels (assigned as discussed above).

Dataset	Size	Features	Positive Class	Sensitive Feature
COMPAS	6172	*criminal history, demographics, COMPAS risk score, jail and prison time*	High Risk (81.4%)	African-American (51.4%)
Communities & Crime	1994	*race, age, education, police demographics, marriage status, citizenship*	Violent Crime Rate (50%)	White Population (continuous)
German Credit	1000	*account information, credit history, loan purpose, employment, demographics*	Good Customer (70%)	Male (69%)

Table 4.1: *Summary of Datasets*

4.3 Experimental Results

In this section, we discuss the detailed experimental evaluation of our framework. First, we analyze the effectiveness of the adversarial classifiers generated by our framework. More specifically, we test how well these classifiers can mask their biases by fooling multiple post hoc explanation techniques. Next, we evaluate the robustness of our adversarial classifiers by measuring how their effectiveness varies with changes to different parameters (e.g., weighting kernel, background distribution). Lastly, we present examples of post hoc explanations (both LIME and SHAP) of individual instances in the data to demonstrate how the biases of the classifier f are successfully hidden.

Datasets We experimented with multiple datasets pertaining to diverse yet critical real world applications such as recidivism risk prediction, violent crime prediction, and credit scoring. Below, we describe these datasets in detail (See Table 4.1 for detailed statistics). Our first dataset is the **COMPAS** dataset which was collected by ProPublica [Ang+16a]. This dataset captures detailed information about the criminal history, jail and prison time, demographic attributes, and COMPAS risk scores for 6172 defendants from Broward County, Florida. The sensitive attribute in this dataset is race – 51.4% of the defendants are African-American. Each defendant in the data is labeled either as high-risk or low-risk for recidivism. Our second dataset is **Communities and Crime** (CC) that captures various socio-economic and law enforcement aspects, as well as crime across various communities in the US [RB02]. This dataset contains information about 1994 communities (each community is a data point) in total. The sensitive attribute in this dataset is the percentage of Caucasian population. This percentage varies widely across communities and ranges from 2.68% to 99.63%. Each community is assigned a label depending on if the violent crime proportion in that community is above the median (violent crime rampant) or not. The third dataset we use is the **German credit** dataset from the UCI repository [BKM99]. This dataset captures financial and demographic information (including account information, credit history, employment, gender) of about 1000 loan applicants. The sensitive attribute in this dataset is gender – 69% of the individuals are male. Each individual in the dataset is labeled either as a good customer or a bad one depending on their credit risk.

Experimental Setup Here, we discuss the implementation details of our framework. We randomly split all the datasets into training (90%) and test (10%) sets. We build our framework using the training set and evaluate it on the test set.

Biased Classifier f: We construct f to be a perfectly discriminatory classifier that uses a sensitive feature i.e., f makes predictions purely based on race in the case of COMPAS and Communities and Crime datasets (if *race* = *African American*, then predict 1, otherwise set predict 0), and based on gender for the German Credit dataset.

Perturbations & OOD classifier: For LIME, we generate perturbations for every data point in the data set $\mathcal{X}$ by adding random noise sampled from $\mathcal{N}(0,1)$ to each feature value. For SHAP, we randomly choose a subset of features for each data point and mark their values as missing by replacing them with their corresponding values from background distribution[3].We choose the background distribution for SHAP to be cluster centers from k-means with 10 clusters. We use these newly generated perturbations as well as the input data points to train the OOD classifier to detect OOD samples (See Proposed Framework section). We use the standard sklearn implementation of a random forest with 100 tree estimators as our OOD classifier.

Unbiased Classifier ψ: We build the unbiased classifier ψ by constructing synthetic *uncorrelated features* that have zero correlation with sensitive attributes (e.g., race or gender). We experiment with one or two uncorrelated features. When we only have one uncorrelated feature in a

[3] See Lundberg and Lee [LL17] for more details.

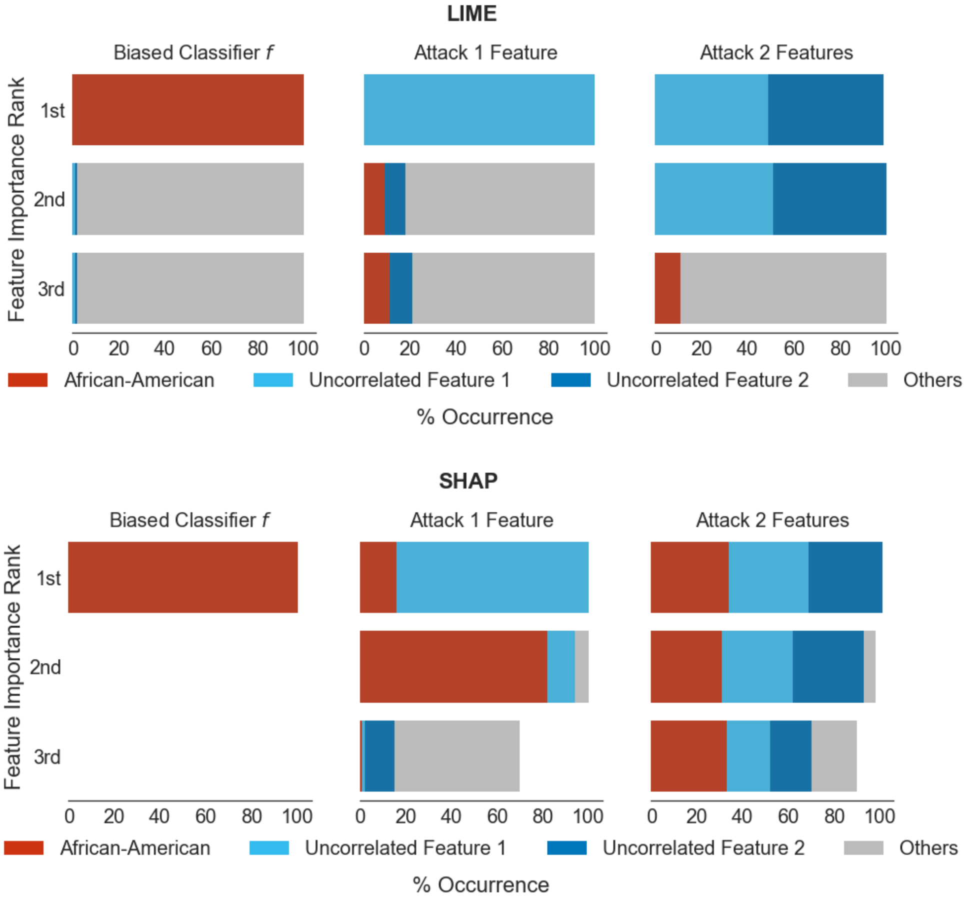


Figure 4.2: ***COMPAS:*** *% of data points for which each feature (color coded) shows up in top 3 (according to LIME and SHAP's ranking of feature importances) for the biased classifier f (left), our adversarial classifier where ψ uses only one uncorrelated feature to make predictions (middle), and our adversarial classifier where ψ uses two uncorrelated features to make predictions (right).*

particular experiment, ψ solely uses that to make predictions (if uncorrelated feature = 1, then predict 1, else predict 0). On the other hand, when we have two uncorrelated features in an experiment, we base the predictions on the xor of those two features. Note that ψ does not have to be restricted to always use synthetic uncorrelated features. It can also use any other existing feature in the data to make predictions. We experiment with synthetic uncorrelated features on COMPAS and CC dataset, and with *Loan Rate % Income* feature on the German credit dataset.

Generating Explanations: We use default LIME tabular implementation without discretization, and the default Kernel SHAP implementation with k-means with 10 clusters as the background distribution.

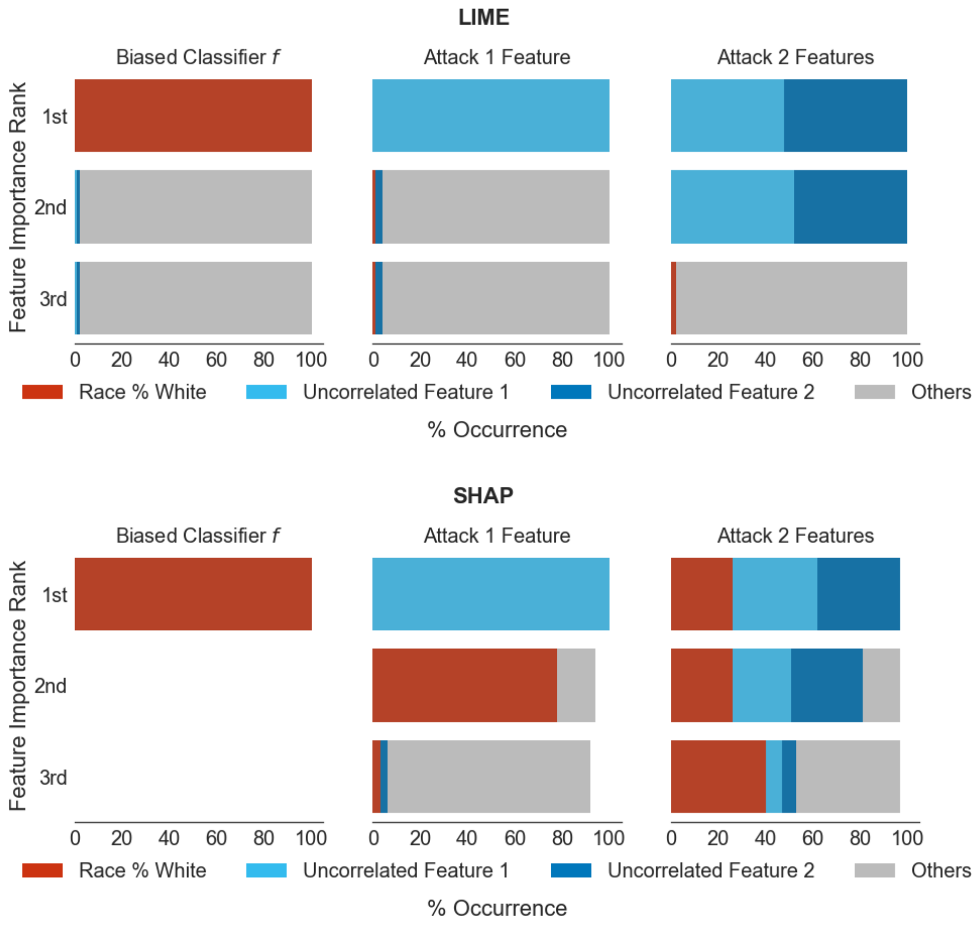


Figure 4.3: ***Communities and Crime:*** *Similar to Fig 4.2;* Race % White *is the sensitive feature here.*

4.3.1 Effectiveness of Adversarial Classifiers

To evaluate how successful our *attacks* are on LIME and SHAP, we compute the percentage of data points for which race, uncorrelated features (in case of COMPAS and CC) or *Loan Rate % Income* (in case of German credit data) show up in top 3 when features are ranked based on feature attributions output by LIME and SHAP. In figures 4.2, 4.3, and 4.4, we show the results of these experiments.[4]

In case of **LIME**, when a single feature is used for the attack, i.e., ψ uses a single feature for making predictions, the adversarial classifier e successfully shifts the feature importance in

[4] Code available at: `https://github.com/dylan-slack/Fooling-LIME-SHAP`

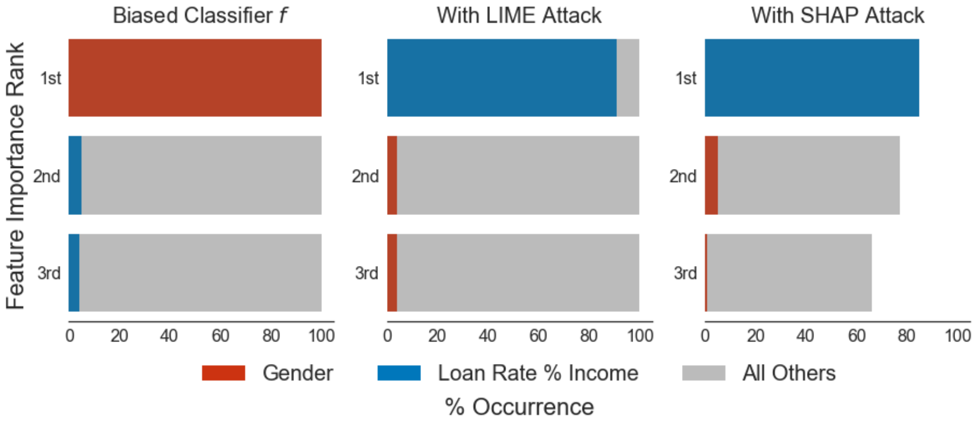
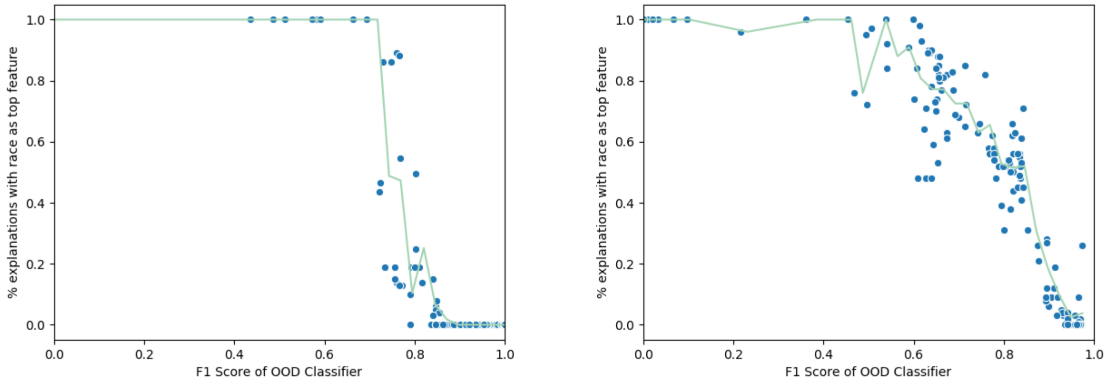


Figure 4.4: ***German credit:*** *Similar to Fig* 4.2 *and* 4.3, *but unbiased classifier ψ uses an existing feature (*Loan Rate % Income*) to make predictions, and* Gender *is the sensitive feature. Feature importances for the biased classifier f shown in the figure (left) are generated using LIME; SHAP also produces similar feature importance values.*

LIME from the true informative feature (the sensitive feature used by f) to the uncorrelated feature in 100% of held-out points for COMPAS (Figure 4.2 - top and middle) and CC (Figure 4.3 - top and middle), and to the *Loan % Income* feature in 91% for German (Figure 4.4 - middle). Furthermore, the sensitive feature appears at 2nd or 3rd rank position roughly only as often as other features. When both uncorrelated features are used to define ψ (for COMPAS and CC), the 1st and 2nd rank positions for nearly all the data points are occupied by these features, with the sensitive feature entering into 3rd place at about the same rate as other features (Figures 4.2 & 4.3 - top right). These results demonstrate that the LIME explanation technique has been effectively fooled by the adversarial classifier e for all three datasets.

For **SHAP**, when a single uncorrelated feature is used for the attack, the adversarial classifier e successfully shifts the feature importance from the sensitive feature in 100% of data points in CC (Figure 4.3 - bottom and middle), 84% of data points in COMPAS (Figure 4.2 - bottom and middle), and 85% of data points in German. When two uncorrelated features are used in the attack (for COMPAS and CC), the adversarial classifier is less successful in removing the bias feature from the first position in the ranking, succeeding in only 67% and 71% instances for COMPAS and CC respectively (Figures 4.2 & 4.3 - bottom right). This is due to SHAP's local accuracy property that ensures that feature attributions must add up to the difference between



(a) *LIME COMPAS Sensitivity Analysis*

(b) *SHAP COMPAS Sensitivity Analysis*

Figure 4.5: *Effectiveness of our attack: % of data points for which race is the most important feature vs. the F1 score of the OOD classifier (perturbation detection) for both LIME and SHAP on the COMPAS dataset over a variety of models (fit with line plot over mean of binned points).*

a given prediction and the average prediction for the background distribution. This property will tend to distribute feature attributions across several features when it is not possible to identify a single most informative feature.

To ensure that the adversarial classifiers e are faithfully approximating the original classifiers f on instances from the original dataset, we evaluate the *fidelity* of e w.r.t. f by computing the percentage of held-out instances on which their predictions agree. In case of LIME, our adversarial classifiers perfectly reproduce the classifier f on all datasets 100% of the time. On the other hand, in case of SHAP, our adversarial classifiers are slightly less effective: 75% for CC, 80% for German, and 91% for COMPAS. These results indicate that our approach has constructed adversarial classifiers e that successfully fool LIME and SHAP, making them look like the unbiased ψ classifiers, while behaving like the original biased classifiers f on input data.

4.3.2 Effect of Perturbation Detection Accuracy

Our approach relies on the OOD classifier to detect whether a given instance is a result of perturbation or not. Here, we investigate how accurate does this classifier have to be for our attacks to be effective. We train a number of different OOD classifiers with varying accuracies

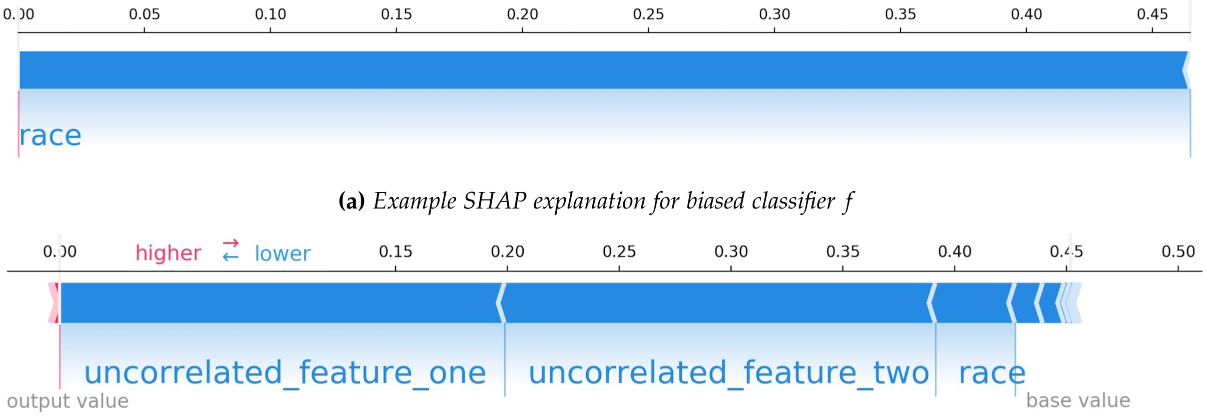
(F1 scores) on the COMPAS dataset. For each OOD classifier, we evaluate the effectiveness of the resulting attacks (measured as percentage of data points for which race is the most important feature in the post hoc explanations). The results are presented in Figure 4.5. When the OOD classifiers are unable to adequately differentiate between perturbed instances and points from the input dataset, both LIME and SHAP explanations detect race as the most important feature i.e., the adversarial attacks are not successful. However, once the OOD classifier is more accurate, the adversarial attacks are successful, and both LIME and SHAP do not identify race as the most important feature. Between the two, LIME requires the OOD classifier to be more accurate for the attack to be effective, but is very vulnerable to these attacks once the F1 score is above 0.8. SHAP, on the other hand, starts responding to the attacks with a less accurate OOD classifier (F1 score $\sim$ 0.45). However, the effect is more gradual; it needs a very accurate OOD classifier to completely hide the bias. These results indicate that our attacks are effective as long as it is possible to differentiate between perturbed instances and input data points with a reasonable accuracy.

4.3.3 Robustness to Hyperparameters

We analyze the behavior of our framework when different parameters are varied *at test time* when generating LIME and SHAP explanations for COMPAS; we summarize the key results here. First, we observe that classifiers that are adversarial to LIME are ineffective against SHAP explanations, while adversarial classifiers designed for SHAP have varying effectiveness against LIME. On varying the LIME kernel width between $0.15\sqrt{M}$ to $0.95\sqrt{M}$ (default kernel width is $0.75\sqrt{M}$, where M is the number of features), the attacks remain equally effective. Additionally, changing the LIME kernel to use $l1$ norm for distance instead of the default $l2$ norm also does not impact the effectiveness. For SHAP, we train our adversarial classifier assuming the default background distribution of k-means with 10 clusters; 5 and 15 clusters at test-time are also equally effective. Using a background distribution of all zeroes for SHAP makes our attacks even more pronounced.

Feature	Before	After attack
Race	-0.42	0.0
Uncorrelated feature 1	0.0	0.42
Average of other features	0.0	0.0

Table 4.2: *Feature coefficients of LIME explanations for an instance from COMPAS, before and after an attack (ψ uses a single feature).*



(a) *Example SHAP explanation for biased classifier f*

(b) *SHAP explanation of the same instance after the attack (e)*

Figure 4.6: *Example SHAP explanations for a data point with biased classifier f (top) and adversarial classifier e (bot.).*

4.3.4 Example Explanations

Here, we present few example explanations from the COMPAS dataset that demonstrate the effectiveness of the technique. In case of LIME explanations in Table 4.2, we see that *Race* as a feature is completely hidden from the explanation after the attack, even though it was the sole important feature for the original classifier f. Similarly, for SHAP explanations in Figure 4.6, the sole important feature (*race*) is considerably hidden in the explanation after the attack, although not completely nullified as in the LIME explanation.

4.4 Additional Contemporaneous Research

The field of explainable machine learning continues to be extremely active, generating an ongoing dialogue of critiques of existing methodologies and new techniques designed to respond to these criticisms. In this section, I discuss both the work that preceded and inspired ours, as well as a large volume of work generated in response to this and other contemporary research highlighting the possible vulnerabilities of explanation methods.

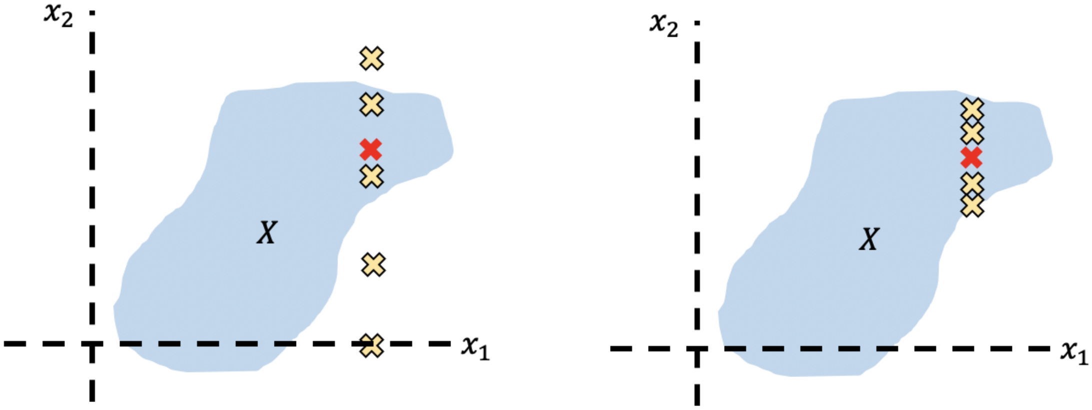
This work focuses on post hoc explainability, in which simpler human-interpretable models are fit to the output of a complex black box after training, as most vulnerabilities result from the necessary mismatch between black boxes and their interpretable proxies. Inherently interpretable models, in which the interpretability constraints are included in the training objective, exhibit no such discrepancy between explanation and ground truth model behavior, but constraints may lead to lower predictive power. I first describe a number of additional preliminaries to support the discussion of a broader scope of work in post hoc explainable machine learning.

4.4.1 Terminology

Perturbation-based Explanation Methods refers to a class of post hoc explanation methods in which explanations are generated by querying labels from the black box model at a selected set of inputs which are related to the input being explained through some *perturbation function* specific to the explanation tool. It is often required (as in e.g. LIME [RSG16] and Anchors [RSG18]) that these perturbations take place in a human-interpretable feature space, which may be some transformation of the original feature space.

Gradient-based Explanation Methods refers to a class of post hoc explanation methods in which feature importances are generated by taking the partial derivatives of the output with respect to the input [SVZ13]. Variations consider multiplying gradients by the input [SGK17] or taking the average gradient over a path from some default input to the input of interest [STY17] to name a few. Note that while perturbation-based approaches require only black box query access and may be used on any model, gradient-based approaches require gradient access.

Propagation-based Explanation Methods refers to a class of post hoc explanation methods specifically for neural network architectures, in which a relevance score initiated at an output node is backpropagated through the network according to some set of axioms to assign



(a) *Interventional Distribution Sampling* **(b)** *Conditional Distribution Sampling*

Figure 4.7: *Selecting perturbations (in yellow) to explain the effect of x_2 on the red point.*

importance to inputs. Layer-wise Relevance Propagation (LRP) is one example, in which the axiom states that relevance must be conserved in every layer [Bac+15].

On-manifold vs **Off-manifold** are used to refer to data points which are either within the considered data distribution (*on-manifold*) or outside the considered data distribution (*off-manifold*).

Interventional Distribution Sampling vs **Conditional Distribution Sampling** are two competing strategies for generating perturbations in perturbation-based post hoc explanation methods such as LIME and SHAP. In interventional distribution sampling, samples are drawn to estimate the effect of *intervening* on a feature or set of features. That is, these feature values are perturbed independently of the remaining features, which are held constant. This sampling strategy may result in sampled points being off-manifold when features are not truly independent. In conditional distribution sampling, samples are drawn to estimate the effect of altering a feature or set of features *conditional* on the remaining features. That is, perturbed features are constrained to values that naturally occur in combination with the fixed feature values. This sampling strategy results in sampled points being on-manifold. See Figure 4.7.

4.4.2 Criticism of Post hoc Explanations

General Criticism Rudin [Rud19] argues that post hoc explanations are not reliable indications of black box model behavior, as these explanations are necessarily unfaithful to the underlying models; if the models were simple enough to be fully represented by an explanation, the model would *be* the explanation. Furthermore, because model agnostic techniques including LIME and KernelSHAP necessarily do not encode any information about the computation of the black box model, the explanations they yield aren't truly descriptive of the model's decision process but rather present correlations between inputs and model outputs.

As a Tool for Bias Detection Doshi-Velez and Kim [DVK17] argue that interpretability, including post hoc explanations, can help us evaluate if a model is biased or discriminatory. On the other hand, Lipton [Lip16] posits that post hoc explanations can never definitively prove or disprove unfairness of any given classifier. Selbst and Barocas [SB18] and Kroll et al. [Kro+16] show that even if a model is completely transparent, it is hard to detect and prevent bias due to the existence of correlated variables.

More recently, Aïvodji et al. [Aïv+19] demonstrated that global post hoc explanations in the form of rule lists can achieve high fidelity to the original model outputs while displaying highly divergent fairness metrics. This property can potentially be exploited to *fairwash* decisions made by unfair black-box models, by presenting high-fidelity explanations of models that appear more fair than the model is in actuality. Dimanov et al. [Dim+20] show in adversarially trained models that the apparent importance of a feature, as measured by a variety of explanation methods, has no consistent relationship to a set of fairness metrics measured with respect to the feature.

Criticism of LIME and SHAP Early criticism of LIME and SHAP provided empirical evidence of a lack of robustness in perturbation-based explanations. Alvarez-Melis and Jaakkola [AMJ18] demonstrate that LIME and SHAP explanations are sensitive to small changes in input, even when the model prediction remains constant. Several works note that results may vary between runs of the algorithms [Lee+19; Zha+19; ZK19; Che+19], and hyperparameters used to select the perturbations can greatly influence the resulting explanation [Zha+19].

Garreau and Luxburg [GL20] provide a theoretical analysis of LIME for tabular, discretized features.[5] The authors develop a closed-form expression for the explanations returned by LIME under mild assumptions and use this expression to develop properties and intuitions for the explanations returned by LIME for simple models. In some cases, this analysis reveals favorable properties of LIME explanations, such as that features that do not affect the output provably receive an attribution of zero. However, the analysis also uncovers many qualities that raise concerns about the reliability of LIME explanations. In particular, the authors find that the setting of certain hyperparameters can cause the sign of coefficients to reverse (that is, for one hyperparameter choice, a feature appears to contribute positively to an outcome, but for another hyperparameter choice the contribution appears to be negative) or to reduce feature contributions toward zero. LIME provides little guidance on how to set these hyperparameters, and the default hyperparameter values are unlikely to be appropriate in every use case. The authors additionally find that for indicator or partition-based functions, such as regression trees and the scaffolded classifier used in the present chapter, it is possible for LIME to report large feature contributions despite the function of interest being completely flat in the vicinity of the explained example.

Conditional vs Interventional Distributions Many critiques of Shapley values for explainability focus on the variation in computational methods and approximations attributed to the same conceptual goal of adapting Shapley values to model feature attribution [SN20]. Most central to the execution of the attack in Section 4.2.2 is the distinction between the use of *conditional* and *interventional* distrbutions for calculation of Shapley values. Both sampling strategies have benefits and drawbacks, and these apply to LIME as well; however, criticism of LIME sampling (which is interventional by design) tends to refer to the issue as "unrealistic" sampling, rather than in these more principled terms. This difference in interpretation is reasonable given the motivation for each of the methods. LIME is explicitly intended to explore model behavior, which often requires breaking data correlations through interventional sampling. However, the specification of a sampling neighborhood that is overly off-manifold can lead to results

[5] Note that while this is the setting explored in the original LIME paper, our work uses the default LIME implementation for continuous, non-discretized features.

that don't explain model behavior within the scope expected by the user. Shapley methods, on the other hand, intend to adapt a game theoretic concept that values player contributions conditional on the presence of other players to the model explanation setting, necessarily creating a theoretical tension.

Because the work in this chapter highlights a vulnerability related to interventional distribution sampling, it has been seen as advocating for conditional distribution sampling [CLL20]. In fact, I believe that both sampling methods have merits and that comparing the results of one to the other can result in additional insights for a user who understands the implications of each.

Kumar et al. [Kum+20] provide a critical look at the use of both conditional and interventional distributions. I focus first on the issues they describe with interventional implementations, as this class includes KernelSHAP, the method used in this chapter. In interventional methods for estimating Shapley values, the value of a feature set in an example is calculated by substituting in values from a specified background distribution, meant to represent the baseline state when features are absent. This distribution is often taken to be either a set of random values from the marginal distribution of each feature, the average value of the feature over the training set, or zero. Importantly, these methods assume feature independence: when features are not truly independent, this results in evaluating the model at points that are out-of-distribution relative to the training set. On these points, model behavior can vary widely as it is unconstrained by the training objective. This behavior will then affect explanations, potentially allowing for multiple explanations for a model that behaves similarly on the example of interest. Even without intervention by an adversary, the explanation will partially depend on the extrapolation behavior of the model class [HM19] (see Figure 4.8 for intuition). In addition, the work in this chapter shows that an adversary can also use this property to manipulate the explanation.

In conditional methods for estimating Shapley values, the value of a feature set in an example is calculated as the expected value of the function conditional on the features in the set being equal to the example values. In particular, this means that out-of-distribution samples are *never* used in calculating the feature values. This avoids the issues above, although Kumar et al. [Kum+20] note that it also introduces new concerns: when features are correlated, it

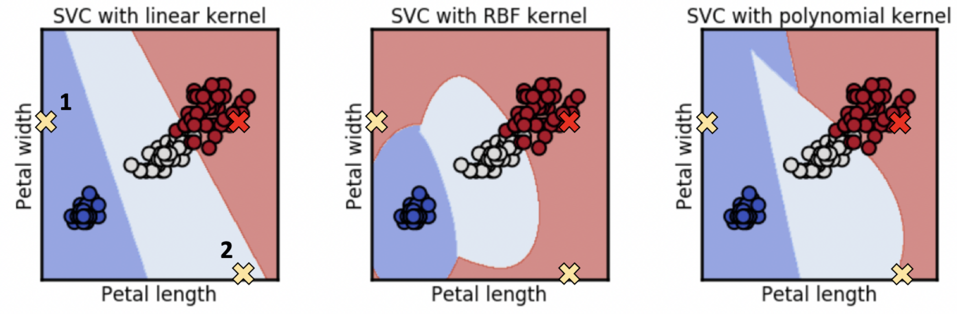


Figure 4.8: *Extrapolation behavior of similarly accurate models results in different explanations for the red point using yellow KernelSHAP perturbations, assuming a background distribution of zeros. Note that the queried labels will be (L to R): (Blue, White), (Red, Red), (Blue, Red). When used in the KernelSHAP formula, these labels will result in varied explanations for the prediction of the red point, even though model behavior is similar on-manifold across all three models.*

is not possible to disentangle the individual contributions, resulting in feature attributions that may not accurately represent the influence that changing a given feature would have. In particular, when features are correlated, it is possible for importance to be assigned to a feature that is not used by the model, as is also shown by Sundararajan and Najmi [SN20]. This occurs because the Shapley framework divides value among features when the contribution cannot be attributed to a single feature, and without sampling off-manifold it is impossible to identify which of two features, one used by the model and one merely correlated, is driving model behavior. Janzing, Minorics, and Blöbaum [JMB20] further develop these theoretical concerns, using a causal framework to argue in favor of the use of interventional methods.

More practically, the true conditional distributions are generally not known, making it difficult to use conditional methods even when desired. The issue of how best to approximate these conditional distributions is among the topics covered by Covert, Lundberg, and Lee [CLL20], who present a unified framework for describing 25 different explanation methods, including LIME and SHAP. These methods all share the same goal of calculating feature attributions by analyzing the value of a function after "removing" subsets of features. The authors develop a set of three design choices that describe the different methods: the *feature removal method*, the *model behavior* being analyzed (LIME and SHAP both consider model predictions), and the *summary presentation* (e.g., LIME presents a linear model; SHAP presents Shapley values). Among these, the choice of feature removal method determines whether the subset value calculations approximate conditional distributions. The authors take the position of a preference for conditional over interventional distributions and provide a novel justification for this stance, citing that the true conditional distributions are unique in forming a valid probability distribution over the outcome, a property they refer to as *consistency*. Covert, Lundberg, and Lee [CLL20] show both theoretically and empirically that the default implementations of LIME and KernelSHAP, which sample according to a background distribution and assume feature independence, unsurprisingly use poor approximations of the true conditional distribution under most circumstances. However, they find promising results for other proposed techniques for calculating conditional values, which are detailed in the following section.

While most of these works tend to advocate for the use of either conditional or interventional distributions, and Kumar et al. [Kum+20] present their irreconcilability as a critical flaw of Shapley values for explainability in general, Chen et al. [Che+20] argue that the existence of these two treatments can actually be beneficial, allowing users to tailor explanations to their specific application.

4.4.3 New Sampling Methods for LIME and SHAP

Because LIME and KernelSHAP use interventional rather than conditional distributions, most proposed methods for improving the sampling in LIME and SHAP emphasize the benefits of conditional sampling and attempt to provide approximations (note that sampling from the true conditional distributions tends to be infeasible due to being both ill defined and computationally intensive).

Aas, Jullum, and Løland [AJL19] propose methods for sampling from the true conditional distributions when the data approximates certain known distributions, such as conditional Gaussian, as well as a strategy for using a weighted empirical distribution for arbitrary non-parametric data distributions. Using known synthetic distributions for which they can calculate the true analytical Shapley values, they find that their method provides a closer approximation than KernelSHAP. However, the methods, and in particular the non-parametric approach, are computationally intensive.

Frye et al. [Fry+20] propose two methods for computing Shapley values using approximate

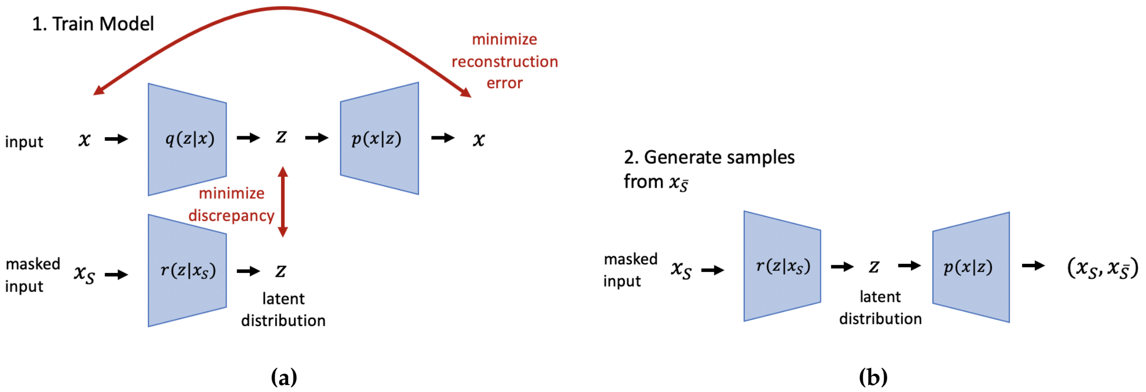


Figure 4.9: *Architecture of the masked encoder network used by Frye et al. [Fry+20]. Variational autoencoder q and decoder p discover the latent distribution of the data, z. The masked encoder r attempts to match outputs $p(z|x)$ using only a masked subset of x. To generate samples from the conditional distribution for a point, the masked encoder $r(z|x_S)$ is combined with the decoder $p(x|z)$.*

conditional distributions for the removed features where each involves training just one additional model. In the first, unsupervised method, the authors use a variational autoencoder [KW13], a model designed to discover a low dimensional latent representation of a data distribution, to create a generative model of conditional datapoints. The traditional autoencoder (with parameters determined by a neural network), which is composed of an encoder that maps data points to a latent space and a decoder that maps latent representations back to the original data points, attempting to minimize the reconstruction error, is augmented with a second encoder network that attempts to match latent representations using only masked subsets of the input (See Figure 4.9). Samples from the conditional distribution are then generated by sampling from the masked encoder network given a desired example and then sampling from the decoder network given the latent variables acquired in the previous step. The conditional value function can then be estimated by averaging the original function value over a number of these samples.

In the second, supervised method, the authors train a surrogate model to approximate the conditional value function directly by minimizing prediction error relative to the original function values using only masked subsets. In comparing against the empirical conditional distributions for a dataset with limited values (note that many continuous values make the empirical conditional distribution overly sparse with a fixed number of datapoints), the authors find that the supervised method is both more accurate and more efficient with respect to model evaluations, as it does not require sampling. Covert, Lundberg, and Lee [CLL20] also test the use of a similar surrogate model for generating conditional values and find that this method ranks feature importances such that the use of top-ranked features for prediction leads to lower losses than features selected when using background values and marginal samples (the approximations used by default LIME and SHAP, respectively).

Several works propose variations on the LIME sampling procedure that can increase its stability and local accuracy by using samples closer to the true conditional distribution. Zafar and Khan [ZK19] use a deterministic set of nearby points from the training data to generate explanations for any given example, which forces all perturbations to be on-manifold and removes variation in explanations across runs of the algorithm. However, the authors note that the accuracy of this approach is limited by the density of training data around a desired explanation. Shankaranarayana and Runje [SR19] draw perturbations according to a Gaussian distribution, as in the implementation of LIME for continuous data, but use an autoencoder (similar to Frye et al. [Fry+20]) to weight points according to their similarity in the latent space determined by the encoder, rather than the original data space. This is designed to encourage higher weightings for on-manifold points relative to off-manifold points. Saito et al. [Sai+20] propose training a conditional tabular GAN (generative adversarial network [Goo+14]) from which to sample points consistent with the data distribution when generating LIME explanations. The authors show that this results in explanations that are more robust to the adversarial attack described in the present chapter.

4.4.4 Adversarial Explanations

The work in this chapter concerns the development of a specific attack that exploits the perturbation strategies of LIME and KernelSHAP explanations to train an adversarial model. Additional adversarial attacks have been developed for other post hoc explanation methods and attack vectors (for example, adversarial model *inputs*).

Ghorbani, Abid, and Zou [GAZ19] show that some gradient-based explanation techniques for image-based deep neural networks can be highly sensitive to small perturbations in the input even though the underlying classifier's predictions remain unchanged. In particular, they develop a method for identifying an imperceptibly perturbed input for which the prediction remains the same while the regions of the input highlighted as contributing to that prediction vary substantially. Dombrowski et al. [Dom+19] build upon this idea, showing that not only can these feature maps be altered with small perturbations in the input image, they can be arbitrarily changed to resemble an explanation of the adversary's choosing. The attack proposed by Heo, Joo, and Moon [HJM19] considers both gradient-based and propagation-based explanations, and is more similar to the attack in this chapter in that the authors focus on fine-tuning the *model* rather than the input, such that explanation methods are inconsistent with the model's prediction across the entire test set.

Dimanov et al. [Dim+20] develop an adversarial method to reduce the appearance of a sensitive feature in a variety of feature importance-based explanations, including LIME and SHAP as well as gradient-based methods. The authors tune existing models with a regularization term on the norm of the gradient of the sensitive feature, encouraging models that flatten the gradient of the sensitive feature in the vicinity of training points. They show that this technique successfully lowers the perceived importance of a sensitive feature across all explanation models tested, while maintaining predictions that are fairly consistent with the original model. This suggests that the attack is effective in shifting gradients of the model away from regions that are expected to be audited, allowing it to obfuscate any use of sensitive features.

Anders et al. [And+20] provide a theoretical proof that models can be manipulated to provide arbitrary gradient and propagation-based explanations despite maintaining the same predictions for data in the distribution of the training set when the underlying structure of the input data (i.e. the dimension of the data manifold) is of significantly lower dimension than the input data, for example due to correlations. This occurs because model gradients are constrained only within the data manifold but are calculated over all input dimensions. Additional dimensionality beyond that of the latent data manifold allows for setting the gradients of directions orthogonal to the data manifold arbitrarily (See Figure 4.10), which leads to flexibility in manipulating resulting explanations. The authors develop an adversarial

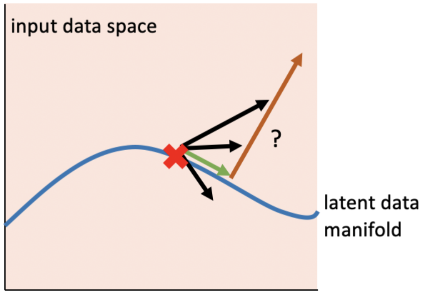


Figure 4.10: *The data manifold (in blue) is low-dimensional relative to the input space (in pink). Model explanations are constrained by training data only along the data manifold (green arrow). Explanation components orthogonal to the data manifold (orange arrow) may be controlled by an adversarial model without affecting predictions on the training data, allowing for many possible explanations of the same model (along black arrows), especially as dimensionality grows.*

model to provide such explanations and prove its effectiveness empirically. They additionally propose an explanation strategy that is robust to these attacks. This strategy reveals model gradients restricted to the data manifold, better representing the model behavior on training data points and tending to produce explanations similar to those in the absence of an attack. To achieve this, the authors use k nearest neighbors to estimate the tangent plane to the data manifold in the vicinity of a point to be explained and then project model explanations into this plane.

4.5 Conclusion

This chapter introduced a novel framework that can effectively hide discriminatory biases of any black box classifier. The approach exploits the fact that post hoc explanation techniques such as LIME and SHAP are perturbation-based to create a *scaffolding* around the biased classifier such that its predictions on input data distribution remain biased, but its behavior on the perturbed data points is controlled to make the post hoc explanations look completely innocuous. Extensive experimentation with real world data from criminal justice and credit scoring domains demonstrates that the approach is effective at generating adversarial classifiers that can fool post hoc explanation techniques, finding that LIME is more vulnerable than SHAP. These findings thus suggest the existing post hoc explanation techniques of LIME and SHAP are not sufficient for ascertaining discriminatory behavior of classifiers in sensitive applications.

This work has generated several follow-up research directions in machine learning explainability and adversarial considerations. Recent work has exposed other vulnerabilities in gradient-based and propagation-based techniques, as well as explored the theoretical basis for these vulnerabilities and proposed alternative sampling techniques. I hope that together this line of research can contribute to a better understanding of the limitations of explainability techniques, encouraging human users to interpret results more cautiously.

Chapter 5

Conclusion

In this thesis, I highlight the ways in which traditional machine learning fails to account for human behavior when incorporating humans in a variety of roles. I delineate the ways in which this oversight can lead to suboptimal model outcomes when humans play each of three roles: model user, model subject, and model auditor. In the case of humans as model users or model subjects, I propose solutions that allow for model optimization to consider and adapt to human behavior in common settings. In the case of humans as model auditors, I present an important deficiency in popular model explanation techniques that can lead to models with undesirable qualities (for example, racial or gender inequity) being able to pass certain proposed human audits intended to detect such behavior.

Chapter 2 concerns the setting of computer-assisted decision-making, in which a machine learning model provides a recommendation or risk score to a human decision-maker. I proposed and validated an alternative framework, in which machine learning models provide *representational advice*, allowing for greater flexibility in adapting to human users, and retaining human judgment and agency. In Chapter 3, I designed and tested a new form of model regularization that anticipates how model subjects will take action in response to the incentives created by a transparent model and ensures that these actions result in improved outcomes with high probability. Finally, in Chapter 4, I developed a framework for training adversarial models that is capable of fooling popular model explanation techniques and verified its efficacy on several real world datasets.

As machine learning is increasingly applied to problems in which human behavior determines input data and the use of a model in obtaining outcomes, the achievement of desired results will depend on the ability of humans and machines to predict, at least to some degree, each other's actions. Toward the goal of allowing machines to predict human actions, I presented novel work incorporating human behavioral models into machine learning training pipelines. Toward the goal of allowing humans to predict machine actions, I revealed a vulnerability in popular explainability methods which may lead humans to improperly predict machine behavior. I hope these contributions will lead to improved human-machine collaboration. Moving forward, there are many opportunities for extending the concepts explored in this thesis.

Humans as Model Users One of the greatest hurdles in optimizing for human behavior is minimizing the burden on humans to provide information during the training process. Future work should focus not only on reducing redundancy in human queries and optimizing for the collection of samples that lead to the largest resolution of uncertainty regarding human behavior but also on studying what other forms of feedback may be more informative than labels. This may reduce the human burden in two ways. First, more detailed feedback may allow for a greater level of model refinement per sample. Second, humans may prefer to provide more nuanced forms of feedback [Ame+14]. Thus, even providing a similar number of labels may feel less onerous. What these forms of feedback may be and how best to incorporate them into a system such as M○M are interesting questions for future work.

Explanations should not be ruled out as a useful tool to facilitate human-model collaboration, but more research should be done to understand *how* humans use explanations. In particular, several studies show that explanations can increase the frequency with which people accept model recommendations even when the model is wrong or the explanation is random [LT19; Ban+20]. This suggests a greater focus on helping humans to identify when models are wrong, especially if they are frequently right.

Humans as Model Subjects Current work on the effects of humans responding to model incentives, including the work presented in Chapter 3, as well as strategic classification [Har+16] and recourse [USL19], assume human decisions based on model gradients and cost functions. These cost functions are often taken to be distance functions over some transformation of feature space. It is likely that human behavior is more complicated than this, and that feature changes are correlated in ways that are not fully reflected in static data. A compelling future direction would be to design an experimental setting such that responses in feature space can be directly observed, or at least intended changes can be queried from human users. This would also allow for testing whether current methods are robust to human behavior, and, if not, developing models that will be more applicable in real world deployment.

Humans as Model Auditors Many errors of human auditors may arise as the result of a mismatch between the assumptions a human user has about a given model and its explanation, and the assumptions warranted by the model or explanation method. More work should be done to understand the assumptions users tend to exhibit and how to align user assumptions with those appropriate for the model and task.

Sokol and Flach [SF20] argue for providing *fact sheets* with explanation methods, which highlight functional requirements and assumptions among other qualities. However, it remains to be studied whether or not the addition of such information would deter users from improper inferences. Research in other fields, such as internet security, suggests that such passive warnings are generally ineffective [ECH08], indicating a potential need for active warnings embedded in explainability software.

Additionally, the accessibility and general purpose nature of tools such as LIME and SHAP may invite inappropriate uses and ill-informed users. Hancox-Li and Kumar [HLK21] encourage the development of task-specific tools, and, importantly, call for a greater use of task-specific human validation experiments in explainability research. They additionally suggest the use of multiple explanations or visualizations with more ambiguous implications, inviting a greater degree of human reasoning. It remains an open question whether this additional engagement of the human interpreter would, in fact, lead to fewer errors of overconfidence in machine learning explanations.

In conclusion, machine learning has great potential to extract new and useful insights across a wide range of applications. However, when humans play a role in creating data and interpreting and acting on model outputs, model efficacy will suffer unless models properly account for human behavior. To this end, I believe machine learning must continue to expand away from assumptions of static datasets and rational human behavior and engage with the Human-Computer Interaction and social science communities to cultivate a more human-centered approach to learning.

References

[Abr+20] Josh Abramson, Arun Ahuja, Iain Barr, Arthur Brussee, Federico Carnevale, Mary Cassin, Rachita Chhaparia, Stephen Clark, Bogdan Damoc, Andrew Dudzik, et al. "Imitating interactive intelligence". In: *arXiv preprint arXiv:2012.05672* (2020) (cit. on p. 23).

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein gan". In: *arXiv preprint arXiv:1701.07875* (2017) (cit. on pp. 29, 127).

[Ade+16] Julius A Adebayo et al. "FairML: ToolBox for diagnosing bias in predictive modeling". PhD thesis. Massachusetts Institute of Technology, 2016 (cit. on p. 8).

[Adv] *Introducing AIDA*. 2018. URL: https://advisorcafe.ca/winter-2018/introducing-aida (cit. on p. 3).

[Agr+19] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. "Differentiable convex optimization layers". In: *Advances in Neural Information Processing Systems*. 2019, pp. 9558–9570 (cit. on p. 50).

[Aïv+19] Ulrich Aïvodji, Hiromi Arai, Olivier Fortineau, Sébastien Gambs, Satoshi Hara, and Alain Tapp. "Fairwashing: the risk of rationalization". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 161–170 (cit. on pp. 7, 84).

[AJL19] Kjersti Aas, Martin Jullum, and Anders Løland. "Explaining individual predictions when features are dependent: More accurate approximations to Shapley values". In: *arXiv preprint arXiv:1903.10464* (2019) (cit. on p. 88).

[Alb19] Alex Albright. "If you give a judge a risk score: evidence from Kentucky bail decisions". In: *Harvard John M. Olin Fellow's Discussion Paper* 85 (2019) (cit. on p. 16).

[Alo+] Tal Alon, Magdalen Dobson, Ariel D Procaccia, Inbal Talgam-Cohen, and Jamie Tucker-Foltz. "Multiagent Evaluation Mechanisms". In: () (cit. on p. 47).

[Ame+14] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. "Power to the people: The role of humans in interactive machine learning". In: *Ai Magazine* 35.4 (2014), pp. 105–120 (cit. on pp. 24, 95).

[AMJ18] David Alvarez-Melis and Tommi S Jaakkola. "On the robustness of interpretability methods". In: *arXiv preprint arXiv:1806.08049* (2018) (cit. on p. 84).

[And+20] Christopher Anders, Plamen Pasliev, Ann-Kathrin Dombrowski, Klaus-Robert Müller, and Pan Kessel. "Fairwashing explanations with off-manifold detergent". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 314–323 (cit. on p. 91).

[Ang+16a] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. "Machine bias". In: *ProPublica* (2016) (cit. on p. 74).

[Ang+16b] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. "Machine bias". In: *ProPublica, May* 23.2016 (2016), pp. 139–159 (cit. on pp. 4, 8).

[Ang+17] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. "Learning certifiably optimal rule lists for categorical data". In: *arXiv preprint arXiv:1704.01701* (2017) (cit. on p. 6).

[Arc+19] Filippo Arcadu, Fethallah Benmansour, Andreas Maunz, Jeff Willis, Zdenka Haskova, and Marco Prunotto. "Deep learning algorithm predicts diabetic retinopathy progression in individual patients". In: *NPJ digital medicine* 2.1 (2019), pp. 1–9 (cit. on p. 2).

[Bac+15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". In: *PloS one* 10.7 (2015), e0130140 (cit. on p. 83).

[Ban10] Albert Bandura. "Self-efficacy". In: *The Corsini encyclopedia of psychology* (2010), pp. 1–3 (cit. on p. 21).

[Ban+19] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S Lasecki, Daniel S Weld, and Eric Horvitz. "Beyond Accuracy: The Role of Mental Models in Human-AI Team Performance". In: *Proc. AAAI Conf. on Human Comput. and Crowdsourcing*. 2019 (cit. on p. 23).

[Ban+20] Gagan Bansal, Tongshuang Wu, Joyce Zhu, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel S Weld. "Does the whole exceed its parts? the effect of ai explanations on complementary team performance". In: *arXiv preprint arXiv:2006.14779* (2020) (cit. on pp. 7, 23, 95).

[Ban+21] Gagan Bansal, Besmira Nushi, Ece Kamar, Eric Horvitz, and Daniel S Weld. "Is the Most Accurate AI the Best Teammate? Optimizing AI for Teamwork". In: (2021) (cit. on p. 15).

[Ban89] Albert Bandura. "Human agency in social cognitive theory." In: *American psychologist* 44.9 (1989), p. 1175 (cit. on p. 20).

[Bar+17] Chelsea Barabas, Karthik Dinakar, Joichi Ito, Madars Virza, and Jonathan Zittrain. "Interventions over predictions: Reframing the ethical debate for actuarial risk assessment". In: *arXiv preprint arXiv:1712.08238* (2017) (cit. on p. 20).

[Bar+19] Robert Bartlett, Adair Morse, Richard Stanton, and Nancy Wallace. *Consumer-lending discrimination in the FinTech era*. Tech. rep. National Bureau of Economic Research, 2019 (cit. on p. 4).

[BBS09] Steffen Bickel, Michael Brückner, and Tobias Scheffer. "Discriminative learning under covariate shift". In: *Journal of Machine Learning Research* 10.Sep (2009), pp. 2137–2155 (cit. on pp. 48, 55).

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828 (cit. on pp. 16, 21).

[Bec+20] Yahav Bechavod, Katrina Ligett, Zhiwei Steven Wu, and Juba Ziani. "Causal Feature Discovery through Strategic Modification". In: *arXiv preprint arXiv:2002.07024* (2020) (cit. on p. 47).

[Bel+18] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. "AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias". In: *arXiv preprint arXiv:1810.01943* (2018) (cit. on p. 8).

[Ber+20] Tobias Berg, Valentin Burg, Ana Gombović, and Manju Puri. "On the rise of fintechs: Credit scoring using digital footprints". In: *The Review of Financial Studies* 33.7 (2020), pp. 2845–2897 (cit. on p. 3).

[BKM99] C Blake, E Koegh, and CJ Mertz. "Repository of Machine Learning". In: *University of California at Irvine* (1999) (cit. on p. 75).

[BN99] Daniel Berdichevsky and Erik Neuenschwander. "Toward an ethics of persuasive technology". In: *Comm. ACM* 42.5 (1999), pp. 51–58 (cit. on pp. 28, 43).

[Boa19] Editorial Board. "Turns Out There's a Proper Way to Buy Your Kid a College Slot". In: *nytimes.com* (2019). URL: https://www.nytimes.com/2019/03/12/opinion/editorials/college-bribery-scandal-admissions.html (cit. on p. 12).

[Bou+19] David D Bourgin, Joshua C Peterson, Daniel Reichman, Thomas Griffiths, and Stuart J Russell. "Cognitive model priors for predicting human decisions". In: *arXiv preprint arXiv:1905.09397* (2019) (cit. on p. 23).

[Bre+01] Leo Breiman et al. "Statistical modeling: The two cultures (with comments and a rejoinder by the author)". In: *Statistical science* 16.3 (2001), pp. 199–231 (cit. on p. 45).

[Bro+13] Jeffrey R Brown, Jeffrey R Kling, Sendhil Mullainathan, and Marian V Wrobel. *Framing lifetime income*. Tech. rep. National Bureau of Economic Research, 2013 (cit. on p. 21).

[BS11] Michael Brückner and Tobias Scheffer. "Stackelberg games for adversarial prediction problems". In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2011, pp. 547–555 (cit. on pp. 13, 46).

[BS16] Solon Barocas and Andrew D Selbst. "Big data's disparate impact". In: *Calif. L. Rev.* 104 (2016), p. 671 (cit. on p. 4).

[Cai+19] Carrie J Cai, Emily Reif, Narayan Hegde, Jason Hipp, Been Kim, Daniel Smilkov, Martin Wattenberg, Fernanda Viegas, Greg S Corrado, Martin C Stumpe, et al. "Human-centered tools for coping with imperfect algorithms during medical decision-making". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–14 (cit. on p. 2).

[Car+15] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission". In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2015, pp. 1721–1730 (cit. on p. 4).

[Car+19] Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. "On the utility of learning about humans for human-ai coordination". In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 5174–5185 (cit. on p. 23).

[CDM16] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. "Learning with rejection". In: *International Conference on Algorithmic Learning Theory*. Springer. 2016, pp. 67–82 (cit. on p. 10).

[Cha+20] Stephanie Chan, Vidhatha Reddy, Bridget Myers, Quinn Thibodeaux, Nicholas Brownstone, and Wilson Liao. "Machine learning in dermatology: current applications, opportunities, and limitations". In: *Dermatology and therapy* 10.3 (2020), pp. 365–386 (cit. on p. 16).

[Che+18] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. “This looks like that: deep learning for interpretable image recognition”. In: *arXiv preprint arXiv:1806.10574* (2018) (cit. on p. 6).

[Che+19] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. “L-Shapley and C-Shapley: Efficient Model Interpretation for Structured Data”. In: *International Conference on Learning Representations*. 2019 (cit. on p. 84).

[Che+20] Hugh Chen, Joseph D Janizek, Scott Lundberg, and Su-In Lee. “True to the Model or True to the Data?” In: *arXiv preprint arXiv:2006.16234* (2020) (cit. on p. 88).

[Che73] Herman Chernoff. “The use of faces to represent points in k-dimensional space graphically”. In: *JASA* 68.342 (1973), pp. 361–368 (cit. on p. 35).

[Cho+18] Alexandra Chouldechova, Diana Benavides-Prado, Oleksandr Fialko, and Rhema Vaithianathan. “A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions”. In: *Conference on Fairness, Accountability and Transparency*. PMLR. 2018, pp. 134–148 (cit. on p. 16).

[CI04] Thomas M Campbell II. *The China study: the most comprehensive study of nutrition ever conducted and the startling implications for diet, weight loss and long-term health*. BenBella Books, Inc., 2004 (cit. on p. 12).

[Cli20] Mayo Clinic. *Heart Disease Risk Calculator*. https://www.mayoclinichealthsystem.org/locations/cannon-falls/services-and-treatments/cardiology/heart-disease-risk-calculator. 2020 (cit. on p. 45).

[CLL20] Ian Covert, Scott Lundberg, and Su-In Lee. “Explaining by Removing: A Unified Framework for Model Explanation”. In: *arXiv preprint arXiv:2011.14878* (2020) (cit. on pp. 86–88, 90).

[Cra+08] Henriette Cramer, Vanessa Evers, Satyan Ramlal, Maarten Van Someren, Lloyd Rutledge, Natalia Stash, Lora Aroyo, and Bob Wielinga. “The effects of transparency on trust in and acceptance of a content-based art recommender”. In: *User Modeling and User-adapted interaction* 18.5 (2008), p. 455 (cit. on p. 23).

[CS17] Alison Callahan and Nigam H Shah. “Machine learning in healthcare”. In: *Key Advances in Clinical Informatics*. Elsevier, 2017, pp. 279–291 (cit. on p. 44).

[CS19] Stephen Chan and Eliot L Siegel. “Will machine learning end the viability of radiology as a thriving medical specialty?” In: *The British journal of radiology* 92.1094 (2019), p. 20180416 (cit. on p. 10).

[CT92] Leda Cosmides and John Tooby. “Cognitive adaptations for social exchange”. In: *The adapted mind: Evolutionary psychology and the generation of culture* 163 (1992), pp. 163–228 (cit. on p. 21).

[DAFC20] Maria De-Arteaga, Riccardo Fogliato, and Alexandra Chouldechova. “A case for humans-in-the-loop: Decisions in the presence of erroneous algorithmic scores”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–12 (cit. on pp. 16, 34).

[DB20] Berkeley J Dietvorst and Soaham Bharti. “People reject algorithms in uncertain decision domains because they have diminishing sensitivity to forecasting error”. In: *Psychological science* 31.10 (2020), pp. 1302–1314 (cit. on p. 8).

[DG17] Dheeru Dua and Casey Graff. *UCI machine learning repository*. 2017 (cit. on pp. 58, 66).

[Dim+20] Botty Dimanov, Umang Bhatt, Mateja Jamnik, and Adrian Weller. “You Shouldn’t Trust Me: Learning Models Which Conceal Unfairness From Multiple Explanation Methods.” In: *SafeAI@ AAAI*. 2020 (cit. on pp. 84, 91).

[DMB16] William Dieterich, Christina Mendoza, and Tim Brennan. “COMPAS risk scales: Demonstrating accuracy equity and predictive parity”. In: *Northpoint Inc* 7.7.4 (2016), p. 1 (cit. on p. 8).

[DND20] Amir Dezfouli, Richard Nock, and Peter Dayan. “Adversarial vulnerabilities of human decision-making”. In: *Proceedings of the National Academy of Sciences* 117.46 (2020), pp. 29221–29228 (cit. on p. 23).

[Dom+19] Ann-Kathrin Dombrowski, Maximilian Alber, Christopher J Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. "Explanations can be manipulated and geometry is to blame". In: *arXiv preprint arXiv:1906.07983* (2019) (cit. on p. 91).

[Don+18] Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. "Strategic classification from revealed preferences". In: *Proceedings of the 2018 ACM Conference on Economics and Computation*. 2018, pp. 55–70 (cit. on p. 46).

[DSM15] Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. "Algorithm aversion: People erroneously avoid algorithms after seeing them err." In: *Journal of Experimental Psychology: General* 144.1 (2015), p. 114 (cit. on pp. 8, 21).

[DSM16] Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. "Overcoming algorithm aversion: People will use imperfect algorithms if they can (even slightly) modify them". In: *Management Science* 64.3 (2016), pp. 1155–1170 (cit. on pp. 21, 23).

[DSM18] Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. "Overcoming algorithm aversion: People will use imperfect algorithms if they can (even slightly) modify them". In: *Management Science* 64.3 (2018), pp. 1155–1170 (cit. on p. 8).

[DT16] LM DeBruine and BP Tiddeman. *Webmorph*. 2016 (cit. on pp. 36, 132).

[DTM14] Shichuan Du, Yong Tao, and Aleix M Martinez. "Compound facial expressions of emotion". In: *Proceedings of the National Academy of Sciences* 111.15 (2014), E1454–E1462 (cit. on pp. 35, 131, 132).

[DVK17] Finale Doshi-Velez and Been Kim. "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint arXiv:1702.08608* (2017) (cit. on pp. 6, 65, 84).

[EAMS19] Radwa Elshawi, Mouaz H Al-Mallah, and Sherif Sakr. "On the interpretability of machine learning-based model for predicting hypertension". In: *BMC medical informatics and decision making* 19.1 (2019), p. 146 (cit. on p. 66).

[ECH08] Serge Egelman, Lorrie Faith Cranor, and Jason Hong. “You’ve been warned: an empirical study of the effectiveness of web browser phishing warnings”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2008, pp. 1065–1074 (cit. on p. 96).

[Efr+04] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. “Least angle regression”. In: *The Annals of statistics* 32.2 (2004), pp. 407–499 (cit. on p. 61).

[Elm+15] Avshalom Elmalech, David Sarne, Avi Rosenfeld, and Eden Shalom Erez. “When suboptimal rules”. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015 (cit. on p. 21).

[Eng62] Douglas C Engelbart. “Augmenting human intellect: A conceptual framework”. In: *Menlo Park, CA* (1962) (cit. on p. 21).

[Ens+18] Danielle Ensign, Sorelle A Friedler, Scott Neville, Carlos Scheidegger, and Suresh Venkatasubramanian. “Runaway feedback loops in predictive policing”. In: *Conference on Fairness, Accountability and Transparency*. PMLR. 2018, pp. 160–171 (cit. on p. 3).

[Est+17] 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* 542.7639 (2017), pp. 115–118 (cit. on pp. 2, 20).

[ET94] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994 (cit. on p. 56).

[FJ16] Jonathan B Freeman and Kerri L Johnson. “More than meets the eye: Split-second social perception”. In: *Trends in cognitive sciences* 20.5 (2016), pp. 362–374 (cit. on p. 35).

[Fog98] Brian J Fogg. “Persuasive computers: perspectives and research directions”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1998, pp. 225–232 (cit. on p. 43).

[Fry+20] Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. “Shapley explainability on the data manifold”. In: *arXiv preprint arXiv:2006.01272* (2020) (cit. on pp. 88–90).

[FSV16] Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. “On the (im) possibility of fairness”. In: *arXiv preprint arXiv:1609.07236* (2016) (cit. on p. 8).

[GAZ19] Amirata Ghorbani, Abubakar Abid, and James Zou. “Interpretation of neural networks is fragile”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 3681–3688 (cit. on p. 90).

[GC19a] Ben Green and Yiling Chen. “Disparate interactions: An algorithm-in-the-loop analysis of fairness in risk assessments”. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 2019, pp. 90–99 (cit. on pp. 9, 23).

[GC19b] Ben Green and Yiling Chen. “The principles and limits of algorithm-in-the-loop decision making”. In: *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (2019), pp. 1–24 (cit. on p. 21).

[GG16] Yarin Gal and Zoubin Ghahramani. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: *international conference on machine learning*. 2016, pp. 1050–1059 (cit. on p. 47).

[GH95] Gerd Gigerenzer and Ulrich Hoffrage. “How to improve Bayesian reasoning without instruction: frequency formats”. In: *Psychological review* 102.4 (1995), p. 684 (cit. on p. 21).

[GL20] Damien Garreau and Ulrike von Luxburg. “Looking deeper into LIME”. In: *arXiv preprint arXiv:2008.11092* (2020) (cit. on p. 85).

[GLW06] Uri Gneezy, John A List, and George Wu. “The uncertainty effect: When a risky prospect is valued less than its worst possible outcome”. In: *The Quarterly Journal of Economics* 121.4 (2006), pp. 1283–1309 (cit. on p. 15).

[Goo+14] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial networks”. In: *arXiv preprint arXiv:1406.2661* (2014) (cit. on p. 90).

[GP17] Pratik Gajane and Mykola Pechenizkiy. “On formalizing fairness in prediction with machine learning”. In: *arXiv preprint arXiv:1710.03184* (2017) (cit. on p. 8).

[Gre+09] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. “Covariate shift by kernel mean matching”. In: *Dataset shift in machine learning* 3.4 (2009), p. 5 (cit. on p. 46).

[Guo+17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. “On calibration of modern neural networks”. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1321–1330 (cit. on p. 47).

[HAB17] Sohaib Haseeb, Bryce Alexander, and Adrian Baranchuk. “Wine and cardiovascular health: A comprehensive review”. In: *Circulation* 136.15 (2017), pp. 1434–1448 (cit. on p. 45).

[Hag+20] Nika Haghtalab, Nicole Immorlica, Brendan Lucier, and Jack Wang. “Maximizing Welfare with Incentive-Aware Evaluation Mechanisms”. In: *Proceedings of the 29th International Joint Conference on Artificial Intelligence*. 2020 (cit. on p. 47).

[Har+16] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. “Strategic classification”. In: *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*. 2016, pp. 111–122 (cit. on pp. 13, 46, 49, 96).

[Har21] Beverly Harzog. *5 Sneaky Ways to Improve your Credit Score*. 2021. URL: https://clark.com/credit/5-sneaky-ways-to-increase-your-credit-score/ (cit. on p. 11).

[Her+11] Hal E Hershfield, Daniel G Goldstein, William F Sharpe, Jesse Fox, Leo Yeykelis, Laura L Carstensen, and Jeremy N Bailenson. “Increasing saving behavior through age-progressed renderings of the future self”. In: *Journal of Marketing Research* 48.SPL (2011), S23–S37 (cit. on p. 38).

[HIV19] Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. "The disparate effects of strategic manipulation". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 2019, pp. 259–268 (cit. on pp. 13, 47).

[HJM19] Juyeon Heo, Sunghwan Joo, and Taesup Moon. "Fooling Neural Network Interpretations via Adversarial Model Manipulation". In: *Advances in Neural Information Processing Systems 32*. 2019, pp. 2921–2932 (cit. on p. 91).

[HLA15] José Miguel Hernández-Lobato and Ryan Adams. "Probabilistic backpropagation for scalable learning of bayesian neural networks". In: *International Conference on Machine Learning*. 2015, pp. 1861–1869 (cit. on p. 47).

[HLK21] Leif Hancox-Li and I Elizabeth Kumar. "Epistemic values in feature importance methods: Lessons from feminist epistemology". In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 2021, pp. 817–826 (cit. on p. 96).

[HM19] Giles Hooker and Lucas Mentch. "Please stop permuting features: An explanation and alternatives". In: *arXiv preprint arXiv:1905.03151* (2019) (cit. on p. 86).

[Ibr+19] Mark Ibrahim, Melissa Louie, Ceena Modarres, and John Paisley. "Global Explanations of Neural Networks: Mapping the Landscape of Predictions". In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. AIES '19. Honolulu, HI, USA, 2019, pp. 279–287. ISBN: 978-1-4503-6324-2 (cit. on p. 66).

[Iza94] Carroll E. Izard. "Innate and universal facial expressions: Evidence from developmental and cross-cultural research". In: *Psychological Bulletin* 115.2 (1994), 288–299 (cit. on p. 35).

[JMB20] Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. "Feature relevance quantification in explainable AI: A causal problem". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 2907–2916 (cit. on p. 87).

[Jun+20] Jongbin Jung, Connor Concannon, Ravi Shroff, Sharad Goel, and Daniel G Goldstein. "Simple rules to guide expert classifications". In: *Journal of the Royal Sta-*

tistical Society: Series A (Statistics in Society) 183.3 (2020), pp. 771–800 (cit. on p. 2).

[Kah11] Daniel Kahneman. *Thinking, fast and slow*. Macmillan, 2011 (cit. on p. 21).

[KAH19] Vladimir Yu Kiselev, Tallulah S Andrews, and Martin Hemberg. “Challenges in unsupervised clustering of single-cell RNA-seq data”. In: *Nature Reviews Genetics* 20.5 (2019), pp. 273–282 (cit. on p. 31).

[Kah20] Jeremy Kahn. *Can an A.I. algorithm help end unfair lending? This company says yes*. https://fortune.com/2020/10/20/artificial-intelligence-unfair-lending/. 2020 (cit. on p. 3).

[Kau+20] Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. “Interpreting Interpretability: Understanding Data Scientists’ Use of Interpretability Tools for Machine Learning”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–14 (cit. on p. 7).

[KD99] Justin Kruger and David Dunning. “Unskilled and unaware of it: how difficulties in recognizing one’s own incompetence lead to inflated self-assessments.” In: *Journal of personality and social psychology* 77.6 (1999), p. 1121 (cit. on p. 15).

[KH01] Roger Koenker and Kevin F Hallock. “Quantile regression”. In: *Journal of economic perspectives* 15.4 (2001), pp. 143–156 (cit. on p. 56).

[KHS09] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. “A least-squares approach to direct importance estimation”. In: *Journal of Machine Learning Research* 10.Jul (2009), pp. 1391–1445 (cit. on p. 55).

[Kil20] David Killock. “AI outperforms radiologists in mammographic screening”. In: *Nature Reviews Clinical Oncology* 17.3 (2020), pp. 134–134 (cit. on p. 10).

[Kim+18] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory S ayres. “Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)”. In: *Proceedings of the*

35th International Conference on Machine Learning. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, 2018, pp. 2668–2677 (cit. on p. 65).

[Kle+15] Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ziad Obermeyer. “Prediction policy problems”. In: *American Economic Review* 105.5 (2015), pp. 491–95 (cit. on p. 25).

[Kle+18] Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. “Human decisions and machine predictions”. In: *The quarterly journal of economics* 133.1 (2018), pp. 237–293 (cit. on pp. 2, 16).

[KLW94] Augustine Kong, Jun S Liu, and Wing Hung Wong. “Sequential imputations and Bayesian missing data problems”. In: *Journal of the American statistical association* 89.425 (1994), pp. 278–288 (cit. on p. 142).

[KMR16] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. “Inherent trade-offs in the fair determination of risk scores”. In: *arXiv preprint arXiv:1609.05807* (2016) (cit. on p. 8).

[KR19] Jon Kleinberg and Manish Raghavan. “How Do Classifiers Induce Agents to Invest Effort Strategically?” In: *Proceedings of the 2019 ACM Conference on Economics and Computation*. 2019, pp. 825–844 (cit. on p. 47).

[Kra05] M Kratz. “Dietary cholesterol, atherosclerosis and coronary heart disease”. In: *Atherosclerosis: Diet and Drugs* (2005), pp. 195–213 (cit. on p. 12).

[Kro+16] Joshua A Kroll, Solon Barocas, Edward W Felten, Joel R Reidenberg, David G Robinson, and Harlan Yu. “Accountable algorithms”. In: *U. Pa. L. Rev.* 165 (2016), p. 633 (cit. on p. 84).

[KRS14] Been Kim, Cynthia Rudin, and Julie A Shah. “The bayesian case model: A generative approach for case-based reasoning and prototype classification”. In: *Advances in neural information processing systems*. 2014, pp. 1952–1960 (cit. on pp. 6, 7).

[KT13] Daniel Kahneman and Amos Tversky. “Prospect theory: An analysis of decision under risk”. In: *Handbook of the fundamentals of financial decision making: Part I*. World Scientific, 2013, pp. 99–127 (cit. on p. 21).

[Kum+20] I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. “Problems with Shapley-value-based explanations as feature importance measures”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5491–5500 (cit. on pp. 86, 88).

[KW13] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013) (cit. on p. 89).

[Lag+18] Isaac Lage, Andrew Ross, Samuel J Gershman, Been Kim, and Finale Doshi-Velez. “Human-in-the-loop interpretability prior”. In: *Adv. in Neural Info. Proc. Sys.* 2018, pp. 10159–10168 (cit. on pp. 24, 129).

[Lag+19] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Samuel J Gershman, and Finale Doshi-Velez. “Human evaluation of models built for interpretability”. In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*. Vol. 7. 1. 2019, pp. 59–67 (cit. on pp. 7, 24).

[Lar+16] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. “How we analyzed the COMPAS recidivism algorithm”. In: *ProPublica (5 2016)* 9 (2016) (cit. on p. 66).

[LB20] Himabindu Lakkaraju and Osbert Bastani. “" How do I fool you?" Manipulating User Trust via Misleading Black Box Explanations”. In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020, pp. 79–85 (cit. on p. 7).

[LBL16] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. “Interpretable decision sets: A joint framework for description and prediction”. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1675–1684 (cit. on pp. 6, 7, 20).

[LCT20] Vivian Lai, Samuel Carton, and Chenhao Tan. “Harnessing Explanations to Bridge AI and Humans”. In: *arXiv preprint arXiv:2003.07370* (2020) (cit. on pp. 23, 34).

[LD90] John A Lott and Timothy C Durbridge. "Use of Chernoff faces to follow trends in laboratory data". In: *Journal of clinical laboratory analysis* 4.1 (1990), pp. 59–63 (cit. on p. 35).

[LDP10] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. "Personalized news recommendation based on click behavior". In: *Proceedings of the 15th international conference on Intelligent user interfaces*. 2010, pp. 31–40 (cit. on p. 3).

[Lee+19] Eunjin Lee, David Braines, Mitchell Stiffler, Adam Hudler, and Daniel Harborne. "Developing the sensitivity of LIME for better machine learning explanation". In: *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*. Vol. 11006. International Society for Optics and Photonics. 2019, p. 1100610 (cit. on p. 84).

[LHS10] Dan Lockton, David Harrison, and Neville A Stanton. "The Design with Intent Method: A design tool for influencing user behaviour". In: *Applied ergonomics* 41.3 (2010), pp. 382–392 (cit. on p. 27).

[Lic60] Joseph Carl Robnett Licklider. "Man-computer symbiosis". In: *IRE transactions on human factors in electronics* (1960), pp. 4–11 (cit. on p. 21).

[Lin+20] Jimmy Lin, Chudi Zhong, Diane Hu, Cynthia Rudin, and Margo Seltzer. "Generalized and scalable optimal sparse decision trees". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 6150–6160 (cit. on p. 6).

[Lip16] Zachary C Lipton. "The mythos of model interpretability". In: *arXiv preprint arXiv:1606.03490* (2016) (cit. on pp. 65, 84).

[Lip18] Zachary C Lipton. "The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3 (2018), pp. 31–57 (cit. on p. 6).

[Liu+19a] Jeremiah Liu, John Paisley, Marianthi-Anna Kioumourtzoglou, and Brent Coull. "Accurate Uncertainty Estimation and Decomposition in Ensemble Learning". In: *Advances in Neural Information Processing Systems*. 2019, pp. 8950–8961 (cit. on p. 47).

[Liu+19b] Xiaoxuan Liu, Livia Faes, Aditya U Kale, Siegfried K Wagner, Dun Jack Fu, Alice Bruynseels, Thushika Mahendiran, Gabriella Moraes, Mohith Shamdas, Christoph Kern, et al. "A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis". In: *The lancet digital health* 1.6 (2019), e271–e297 (cit. on pp. 2, 20).

[Liu+20] Lydia T Liu, Ashia Wilson, Nika Haghtalab, Adam Tauman Kalai, Christian Borgs, and Jennifer Chayes. "The disparate equilibria of algorithmic decision making when individuals invest rationally". In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020, pp. 381–391 (cit. on pp. 12, 47).

[LL17] Scott Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *arXiv preprint arXiv:1705.07874* (2017) (cit. on pp. 6, 19, 20, 66, 67, 70, 75).

[LM07] Nicholas H Lurie and Charlotte H Mason. "Visual representation: Implications for decision making". In: *Journal of marketing* 71.1 (2007), pp. 160–177 (cit. on p. 28).

[Log17] Jennifer Marie Logg. "Theory of machine: When do people rely on algorithms?" In: *Harvard Business School working paper series# 17-086* (2017) (cit. on pp. 8, 21).

[LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in neural information processing systems*. 2017, pp. 6402–6413 (cit. on p. 47).

[LSY03] Greg Linden, Brent Smith, and Jeremy York. "Amazon. com recommendations: Item-to-item collaborative filtering". In: *IEEE Internet computing* 7.1 (2003), pp. 76–80 (cit. on p. 3).

[LT19] Vivian Lai and Chenhao Tan. "On human predictions with explanations and predictions of machine learning models: A case study on deception detection". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 2019, pp. 29–38 (cit. on pp. 7, 8, 20, 23, 34, 95).

[Man+20] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. “Feedback loop and bias amplification in recommender systems”. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 2145–2148 (cit. on p. 11).

[Mei18] Nicolai Meinshausen. “Causality from a distributional robustness point of view”. In: *2018 IEEE Data Science Workshop (DSW)*. IEEE. 2018, pp. 6–10 (cit. on p. 48).

[Mel18] Steven Melendez. *Insurers turn to artificial intelligence in war on fraud*. `https://www.fastcompany.com/40585373/to-combat-fraud-insurers-turn-to-artificial-intelligence`. 2018 (cit. on p. 3).

[Mil+19] Smitha Milli, John Miller, Anca D Dragan, and Moritz Hardt. “The social cost of strategic classification”. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 2019, pp. 230–239 (cit. on pp. 14, 47).

[Mil56] George A Miller. “The magical number seven, plus or minus two: Some limits on our capacity for processing information.” In: *Psychological review* 63.2 (1956), p. 81 (cit. on p. 21).

[Mla+20] Martin Mladenov, Elliot Creager, Omer Ben-Porat, Kevin Swersky, Richard Zemel, and Craig Boutilier. “Optimizing long-term social welfare in recommender systems: A constrained matching approach”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 6987–6998 (cit. on p. 11).

[MMH20] John Miller, Smitha Milli, and Moritz Hardt. “Strategic Classification is Causal Modeling in Disguise”. In: *Proceedings of the 37th International Conference on Machine Learning*. 2020 (cit. on p. 47).

[MPZ18] David Madras, Toni Pitassi, and Richard Zemel. “Predict Responsibly: Improving Fairness and Accuracy by Learning to Defer”. In: *Adv. in Neural Info. Proc. Sys. 31*. 2018, pp. 6147–6157 (cit. on pp. 10, 24, 25).

[MRW19] Brent Mittelstadt, Chris Russell, and Sandra Wachter. “Explaining explanations in AI”. In: *Proceedings of the conference on fairness, accountability, and transparency*. ACM. 2019, pp. 279–288 (cit. on p. 72).

[MS20] Hussein Mozannar and David Sontag. "Consistent estimators for learning to defer to an expert". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 7076–7087 (cit. on pp. 10, 24).

[Mue+17] Jonas Mueller, David N Reshef, George Du, and Tommi Jaakkola. "Learning optimal interventions". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2017 (cit. on pp. 48, 51, 64).

[MW19] Albert Meijer and Martijn Wessels. "Predictive policing: Review of benefits and drawbacks". In: *International Journal of Public Administration* 42.12 (2019), pp. 1031–1039 (cit. on p. 1).

[Neu17] Adam Neufeld. *In Defense of Risk-Assessment Tools*. `https://www.themarshallproject.org/2017/10/22/in-defense-of-risk-assessment-tools`. 2017 (cit. on p. 10).

[Nie+19] Katrine B Nielsen, Mie L Lautrup, Jakob KH Andersen, Thiusius R Savarimuthu, and Jakob Grauslund. "Deep learning–based algorithms in screening of diabetic retinopathy: A systematic review of diagnostic performance". In: *Ophthalmology Retina* 3.4 (2019), pp. 294–304 (cit. on p. 2).

[NR14] David W Nickerson and Todd Rogers. "Political campaigns and big data". In: *J. Econ. Persp.* 28.2 (2014), pp. 51–74 (cit. on p. 20).

[OT08] Nikolaas N Oosterhof and Alexander Todorov. "The functional basis of face evaluation". In: *Proceedings of the National Academy of Sciences* 105.32 (2008), pp. 11087–11092 (cit. on p. 132).

[PBM16] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. "Causal inference by using invariant prediction: identification and confidence intervals". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 78.5 (2016), pp. 947–1012 (cit. on pp. 48, 50, 51, 64).

[Pea+09] Judea Pearl et al. "Causal inference in statistics: An overview". In: *Statistics surveys* 3 (2009), pp. 96–146 (cit. on p. 45).

[Pea10] Beth Pearsall. “Predictive policing: The future of law enforcement”. In: *National Institute of Justice Journal* 266.1 (2010), pp. 16–19 (cit. on p. 1).

[Per13] Walt L Perry. *Predictive policing: The role of crime forecasting in law enforcement operations*. Rand Corporation, 2013 (cit. on p. 2).

[Per+20] Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. “Performative prediction”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 7599–7609 (cit. on pp. 13, 47, 49).

[PON19] Ravi B Parikh, Ziad Obermeyer, and Amol S Navathe. “Regulation of predictive analytics in medicine”. In: *Science* 363.6429 (2019), pp. 810–812 (cit. on p. 20).

[PS+18] Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. “Manipulating and measuring model interpretability”. In: *arXiv preprint arXiv:1802.07810* (2018) (cit. on pp. 7, 8, 23).

[QC+09] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009 (cit. on p. 51).

[Rag+19] Maithra Raghu, Katy Blumer, Greg Corrado, Jon Kleinberg, Ziad Obermeyer, and Sendhil Mullainathan. “The algorithmic automation problem: Prediction, triage, and human effort”. In: *arXiv preprint arXiv:1903.12220* (2019) (cit. on p. 10).

[Raj+17] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. “Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning”. In: *arXiv preprint arXiv:1711.05225* (2017) (cit. on p. 2).

[RB02] Michael Redmond and Alok Baveja. “A data-driven software tool for enabling cooperative information sharing among police departments”. In: *European Journal of Operational Research* 141.3 (2002), pp. 660–678 (cit. on p. 74).

[RC+18] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. “Invariant models for causal transfer learning”. In: *The Journal of Machine Learning Research* 19.1 (2018), pp. 1309–1342 (cit. on p. 51).

[Red11] M Redmond. “Communities and crime unnormalized data set”. In: *UCI Machine Learning Repository.* (2011) (cit. on p. 66).

[Reg16] General Data Protection Regulation. “Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46”. In: *Official Journal of the European Union (OJ)* 59.1-88 (2016), p. 294 (cit. on p. 71).

[RHDV17] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. “Right for the right reasons: Training differentiable models by constraining their explanations”. In: *arXiv preprint arXiv:1703.03717* (2017) (cit. on p. 24).

[Rib+20] Manoel Horta Ribeiro, Raphael Ottoni, Robert West, Virgílio AF Almeida, and Wagner Meira Jr. “Auditing radicalization pathways on youtube”. In: *Proc. ACM Conf. FAT*. 2020, pp. 131–141 (cit. on p. 43).

[Ric+09] Jennifer J Richler, Michael L Mack, Isabel Gauthier, and Thomas J Palmeri. “Holistic processing of faces happens at a glance”. In: *Vision research* 49.23 (2009), pp. 2856–2861 (cit. on p. 35).

[RMYT18] Nir Rosenfeld, Yishay Mansour, and Elad Yom-Tov. “Discriminative learning of prediction intervals”. In: *International Conference on Artificial Intelligence and Statistics*. 2018, pp. 347–355 (cit. on p. 55).

[Rot+18] Dominik Rothenhäusler, Nicolai Meinshausen, Peter Bühlmann, and Jonas Peters. “Anchor regression: heterogeneous data meets causality”. In: *arXiv preprint arXiv:1801.06229* (2018) (cit. on p. 48).

[RSC19] Rashida Richardson, Jason M Schultz, and Kate Crawford. “Dirty data, bad predictions: How civil rights violations impact police data, predictive policing systems, and justice”. In: *NYUL Rev. Online* 94 (2019), p. 15 (cit. on p. 3).

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “" Why should i trust you?" Explaining the predictions of any classifier”. In: *Proceedings of the 22nd*

ACM SIGKDD international conference on knowledge discovery and data mining. 2016, pp. 1135–1144 (cit. on pp. 6, 7, 19, 20, 65, 67, 70, 82).

[RSG18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Anchors: High-precision model-agnostic explanations". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018 (cit. on p. 82).

[Rub05] Donald B Rubin. "Causal inference using potential outcomes: Design, modeling, decisions". In: *Journal of the American Statistical Association* 100.469 (2005), pp. 322–331 (cit. on p. 45).

[Rud19] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature Machine Intelligence* 1.5 (2019), pp. 206–215 (cit. on pp. 7, 84).

[Sah+15] Amirhossein Sahebkar, Corina Serban, Sorin Ursoniu, Nathan D Wong, Paul Muntner, Ian M Graham, Dimitri P Mikhailidis, Manfredi Rizzo, Jacek Rysz, Laurence S Sperling, et al. "Lack of efficacy of resveratrol on C-reactive protein and selected cardiovascular risk factors—Results from a systematic review and meta-analysis of randomized controlled trials". In: *International journal of cardiology* 189 (2015), pp. 47–55 (cit. on p. 45).

[Sai+20] Sean Saito, Eugene Chua, Nicholas Capel, and Rocco Hu. "Improving LIME Robustness with Smarter Locality Sampling". In: *arXiv preprint arXiv:2006.12302* (2020) (cit. on p. 90).

[Sax+19] Nripsuta Ani Saxena, Karen Huang, Evan DeFilippis, Goran Radanovic, David C Parkes, and Yang Liu. "How do fairness definitions fare? Examining public attitudes towards algorithmic definitions of fairness". In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019, pp. 99–106 (cit. on p. 8).

[SB18] Andrew D Selbst and Solon Barocas. "The intuitive appeal of explainable machines". In: *Fordham L. Rev.* 87 (2018), p. 1085 (cit. on p. 84).

[Sch+09] Gabriele Schweikert, Gunnar Rätsch, Christian Widmer, and Bernhard Schölkopf. "An empirical analysis of domain adaptation algorithms for genomic sequence analysis". In: *Advances in neural information processing systems*. 2009, pp. 1433–1440 (cit. on p. 51).

[SD19] Megan T Stevenson and Jennifer L Doleac. "Algorithmic risk assessment in the hands of humans". In: *Available at SSRN 3489440* (2019) (cit. on pp. 9, 16, 21, 34).

[SEA20] Yonadav Shavit, Benjamin Edelman, and Brian Axelrod. "Learning From Strategic Agents: Accuracy, Improvement, and Causality". In: *Proceedings of the 37 th International Conference on Machine Learning*. 2020 (cit. on p. 47).

[SF20] Kacper Sokol and Peter Flach. "Explainability fact sheets: a framework for systematic assessment of explainable approaches". In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020, pp. 56–67 (cit. on p. 96).

[SGK17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. "Learning important features through propagating activation differences". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3145–3153 (cit. on p. 82).

[Sha51] Lloyd S Shapley. *Notes on the N-person Game–II: The Value of an N-person Game*. Rand Corporation, 1951 (cit. on p. 69).

[Shi00] Hidetoshi Shimodaira. "Improving predictive inference under covariate shift by weighting the log-likelihood function". In: *Journal of statistical planning and inference* 90.2 (2000), pp. 227–244 (cit. on pp. 46, 50, 51, 55).

[Sid12] Naeem Siddiqi. *Credit risk scorecards: developing and implementing intelligent credit scoring*. Vol. 3. John Wiley & Sons, 2012 (cit. on p. 44).

[Sim09] Uri Simonsohn. "Direct risk aversion: Evidence from risky prospects valued below their worst outcome". In: *Psychological Science* 20.6 (2009), pp. 686–692 (cit. on p. 15).

[SM19] Yonadav Shavit and William S Moses. "Extracting Incentives from Black-Box Decisions". In: *arXiv preprint arXiv:1910.05664* (2019) (cit. on p. 57).

[Smi+17] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. "Smoothgrad: removing noise by adding noise". In: *arXiv preprint arXiv:1706.03825* (2017) (cit. on p. 6).

[SN20] Mukund Sundararajan and Amir Najmi. "The many Shapley values for model explanation". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9269–9278 (cit. on pp. 85, 87).

[SND18] Aman Sinha, Hongseok Namkoong, and John Duchi. "Certifying some distributional robustness with principled adversarial training". In: *International Conference on Learning Representations*. 2018 (cit. on p. 47).

[Sno+19] Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D Sculley, Joshua Dillon, Jie Ren, and Zachary Nado. "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift". In: *Advances in Neural Information Processing Systems*. 2019, pp. 13969–13980 (cit. on p. 47).

[Spa+19] Douglas Spangler, Thomas Hermansson, David Smekal, and Hans Blomberg. "A validation of machine learning-based risk scores in the prehospital setting". In: *PloS one* 14.12 (2019), e0226518 (cit. on p. 2).

[SR19] Sharath M Shankaranarayana and Davor Runje. "ALIME: Autoencoder based approach for local interpretability". In: *International conference on intelligent data engineering and automated learning*. Springer. 2019, pp. 454–463 (cit. on p. 90).

[SRP19] Lesia Semenova, Cynthia Rudin, and Ronald Parr. "A study in Rashomon curves and volumes: A new perspective on generalization and model simplicity in machine learning". In: *arXiv preprint arXiv:1908.01755* (2019) (cit. on p. 7).

[Str+17] Aaron F Struck, Berk Ustun, Andres Rodriguez Ruiz, Jong Woo Lee, Suzette M LaRoche, Lawrence J Hirsch, Emily J Gilmore, Jan Vlachy, Hiba Arif Haider, Cynthia Rudin, et al. "Association of an electroencephalography-based risk score with seizure probability in hospitalized patients". In: *JAMA neurology* 74.12 (2017), pp. 1419–1424 (cit. on p. 2).

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3319–3328 (cit. on p. 82).

[Sug+08] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau, and Motoaki Kawanabe. "Direct importance estimation with model selection and its application to covariate shift adaptation". In: *Advances in neural information processing systems*. 2008, pp. 1433–1440 (cit. on pp. 46, 51).

[Sut+20] Reed T Sutton, David Pincock, Daniel C Baumgart, Daniel C Sadowski, Richard N Fedorak, and Karen I Kroeker. "An overview of clinical decision support systems: benefits, risks, and strategies for success". In: *NPJ Digital Medicine* 3.1 (2020), pp. 1–10 (cit. on p. 21).

[SV08] Glenn Shafer and Vladimir Vovk. "A Tutorial on Conformal Prediction." In: *Journal of Machine Learning Research* 9.3 (2008) (cit. on p. 48).

[SVZ13] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034* (2013) (cit. on p. 82).

[Tab+19a] Behzad Tabibian, Stratis Tsirtsis, Moein Khajehnejad, Adish Singla, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. "Optimal decision making under strategic behavior". In: *arXiv preprint arXiv:1905.09239* (2019) (cit. on p. 47).

[Tab+19b] Behzad Tabibian, Utkarsh Upadhyay, Abir De, Ali Zarezade, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. "Enhancing human learning via spaced repetition optimization". In: *Proc. Nat. Acad. of Sci.* 116.10 (2019), pp. 3988–3993 (cit. on p. 20).

[Tan+18] Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. "Distill-and-compare: auditing black-box models using transparent model distillation". In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. ACM. 2018, pp. 303–310 (cit. on p. 65).

[The] *AI, radiology and the future of work*. 2018. URL: https://www.economist.com/leaders/2018/06/07/ai-radiology-and-the-future-of-work (cit. on p. 2).

[Tho80] Peter Thompson. "Margaret Thatcher: A new illusion". In: *Perception* (1980) (cit. on p. 21).

[Tib+19] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel J Candès, and Aaditya Ramdas. "Conformal prediction under covariate shift". In: *arXiv preprint arXiv:1904.06019* (2019) (cit. on p. 48).

[Tib96] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288 (cit. on p. 6).

[Tif19] Kaitlyn Tiffany. "The Tinder algorithm, explained". In: *Vox. Retrieved April* 1 (2019), p. 2019 (cit. on p. 3).

[TK74] Amos Tversky and Daniel Kahneman. "Judgment under uncertainty: Heuristics and biases". In: *science* 185.4157 (1974), pp. 1124–1131 (cit. on p. 15).

[TK81] Amos Tversky and Daniel Kahneman. "The framing of decisions and the psychology of choice". In: *science* 211.4481 (1981), pp. 453–458 (cit. on p. 15).

[TLP19] Natasa Tagasovska and David Lopez-Paz. "Single-Model Uncertainties for Deep Learning". In: *Advances in Neural Information Processing Systems*. 2019, pp. 6414–6425 (cit. on p. 47).

[Tod+08] Alexander Todorov, Chris P Said, Andrew D Engell, and Nikolaas N Oosterhof. "Understanding evaluation of faces on social dimensions". In: *Trends in cognitive sciences* 12.12 (2008), pp. 455–460 (cit. on p. 35).

[Tsc+19] Philipp Tschandl, Cliff Rosendahl, Bengu Nisa Akay, Giuseppe Argenziano, Andreas Blum, Ralph P Braun, Horacio Cabo, Jean-Yves Gourhant, Jürgen Kreusch, Aimilios Lallas, et al. "Expert-level diagnosis of nonpigmented skin cancer by combined convolutional neural networks". In: *JAMA dermatology* 155.1 (2019), pp. 58–65 (cit. on p. 2).

[UR16] Berk Ustun and Cynthia Rudin. "Supersparse linear integer models for optimized medical scoring systems". In: *Machine Learning* 102.3 (2016), pp. 349–391 (cit. on pp. 6, 45).

[USL19] Berk Ustun, Alexander Spangher, and Yang Liu. "Actionable recourse in linear classification". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 2019, pp. 10–19 (cit. on p. 96).

[Ven+03] Viswanath Venkatesh, Michael G Morris, Gordon B Davis, and Fred D Davis. "User acceptance of information technology: Toward a unified view". In: *MIS quarterly* (2003), pp. 425–478 (cit. on p. 21).

[WC19] M. West S.M. Whittaker and K. Crawford. *Discriminating Systems: Gender, Race and Power in AI*. https://ainowinstitute.org/discriminating systems.html . 2019 (cit. on p. 22).

[WGH16] Leanne S Whitmore, Anthe George, and Corey M Hudson. "Mapping chemical performance on molecular structures using locally interpretable explanations". In: *arXiv preprint arXiv:1611.07443* (2016) (cit. on p. 66).

[WHK20] Bryan Wilder, Eric Horvitz, and Ece Kamar. "Learning to complement humans". In: *arXiv preprint arXiv:2005.00582* (2020) (cit. on pp. 10, 24).

[Wue+16] Thorsten Wuest, Daniel Weimer, Christopher Irgens, and Klaus-Dieter Thoben. "Machine learning in manufacturing: advantages, challenges, and applications". In: *Production & Manufacturing Research* 4.1 (2016), pp. 23–45 (cit. on p. 44).

[Yal+19] Adam Yala, Constance Lehman, Tal Schuster, Tally Portnoi, and Regina Barzilay. "A deep learning mammography-based model for improved breast cancer risk prediction". In: *Radiology* 292.1 (2019), pp. 60–66 (cit. on p. 2).

[Yeo+17] Michael Yeomans, Anuj Shah, Sendhil Mullainathan, and Jon Kleinberg. “Making sense of recommendations”. In: *Journal of Behavioral Decision Making* (2017) (cit. on p. 21).

[You85] H Peyton Young. “Monotonic solutions of cooperative games”. In: *International Journal of Game Theory* 14.2 (1985), pp. 65–72 (cit. on p. 70).

[Yu+13] Danxia Yu, Xiao-Ou Shu, Honglan Li, Yong-Bing Xiang, Gong Yang, Yu-Tang Gao, Wei Zheng, and Xianglan Zhang. “Dietary carbohydrates, refined grains, glycemic load, and risk of coronary heart disease in Chinese adults”. In: *American journal of epidemiology* 178.10 (2013), pp. 1542–1549 (cit. on p. 12).

[YWVW19] Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. “Understanding the effect of accuracy on trust in machine learning models”. In: *Proceedings of the 2019 chi conference on human factors in computing systems*. 2019, pp. 1–12 (cit. on pp. 8, 21, 23, 34).

[Zec+18] John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric K Oermann. “Confounding variables can degrade generalization performance of radiological deep learning models”. In: *arXiv preprint arXiv:1807.00431* (2018) (cit. on p. 4).

[Zha+19] Yujia Zhang, Kuangyan Song, Yiming Sun, Sarah Tan, and Madeleine Udell. “" Why Should You Trust My Explanation?" Understanding Uncertainty in LIME Explanations”. In: *arXiv preprint arXiv:1904.12991* (2019) (cit. on pp. 7, 84).

[ZK19] Muhammad Rehman Zafar and Naimul Mefraz Khan. “DLIME: a deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems”. In: *arXiv preprint arXiv:1906.10263* (2019) (cit. on pp. 84, 90).

Appendix A

Appendix to Chapter 2

A.1 General Optimization Issues

A.1.1 Initialization

Because acquiring human labels is expensive, it is important to initialize ϕ to map to a region of the representation space in which there is variation and consistency in human reports, such that gradients lead to progress in subsequent rounds.

In some representation spaces, such as our 2D projections of noisy 3D rotated images, this is likely to be the case (almost any 3D slice will retain some signal from the original 2D image). However, in 4+ dimensions, as well as with the subset selection and avatar tasks, there are no such guarantees.

To minimize non-informative queries, we adopt two initialization strategies:

1. **Initialization with a computer-only model:** In scenarios in which the representation space is a (possibly discrete) subset of input space, such as in subset selection, the initialization problem is to isolate the region of the input space that is important for decision-making. In this situation, it can be useful to initialize with a computer-only classifier. This classifier should share a representation-learning architecture with ϕ but can have any other classifying architecture appended (although simpler is likely better for this purpose). This should result in some ϕ which at least focuses on the features

relevant for classification, if not necessarily in a human-interpretable format.

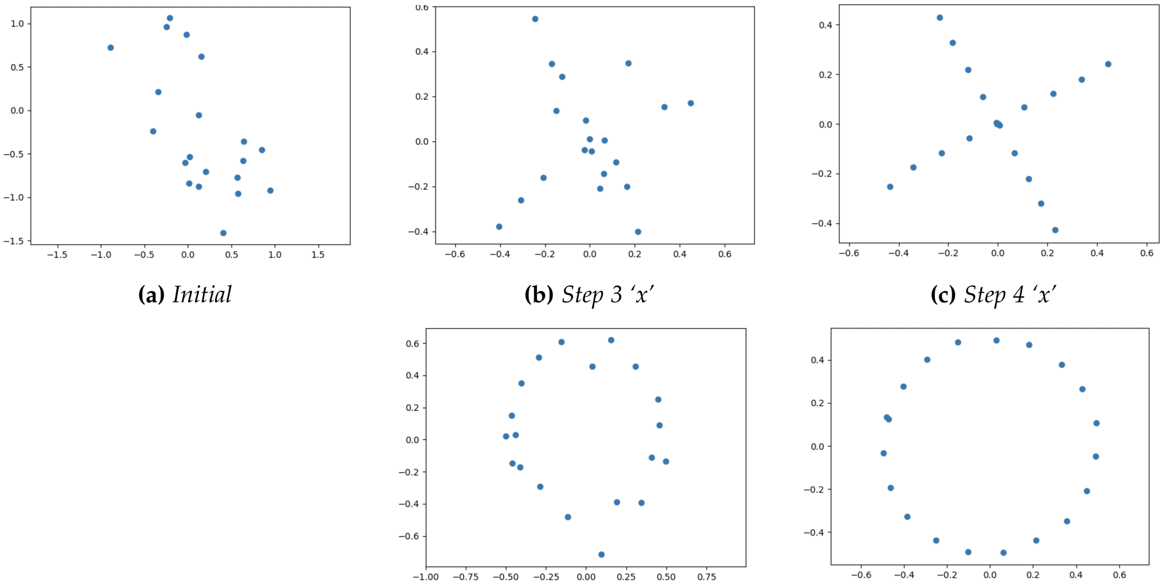
2. **Initialization to a desired distribution with a WGAN:** In scenarios in which the initialization problem is to isolate a region of representation space into which to map all inputs, as in the avatar example, in which we wish to test a variety of expressions without creating expression combinations which will appear overly strange to participants, it can be useful to hand-design a starting distribution over representation space and initialize ϕ with a Wasserstein GAN [ACB17]. In this case, we use a Generator Network with the same architecture as ϕ but allow the Discriminator Network to be of any effective architecture. As with the previous example, this results in an ϕ in which the desired distribution is presented to users, but not necessarily in a way that reflects any human intuitive concept.

A.1.2 Convergence

As is true in general of gradient descent algorithms, the M∘M framework is not guaranteed to find a global optimum but rather is likely to end up at a local optimum dependent on both the initialization of ϕ and $\hat{h}$. In our case, however, the path of gradient descent is also dependent on the inherently stochastic selection and behavior of human users. If users are inconsistent or user groups at different iterations are not drawn from the same behavior distribution, it is possible that learning at one step of the algorithm could result in convergence to a suboptimal distribution for future users. It remains for future work to test how robust machine learning methods might be adapted to this situation to mitigate this issue.

A.1.3 Regularization/Early Stopping

As mentioned in Section 2.3, training ϕ will in general shift the distribution of the representation space away from the region on which we have collected labels for $\hat{h}$ in the previous iterations, resulting in increasing uncertainty in the predicted outcomes. We test a variety of methods to account for this, but developing a consistent scheme for choosing how best to maximize the information in human labels remains future work.



(a) *Initial*

(b) *Step 3 'x'*

(c) *Step 4 'x'*

(d) *Step 3 'o'*

(e) *Step 4 'o'*

Figure A.1: *Images of x-o interface*

- **Regularization of $\hat{h}$:** We test regularization of $\hat{h}$ both with Dropout and L2 regularization, both of which help in preventing overfitting, especially in early stages of training, when the representation distribution is not yet refined. As training progresses and the distribution $\phi_\theta(x)$ becomes more tightly defined, decreasing these regularization parameters increases performance.

- **Training $\hat{h}$ with samples from previous iterations**: We also found it helpful in early training iterations to reuse samples from the previous human labeling round in training $\hat{h}$, as inspired by [Bobu et al. 2018]. [1] We weight these samples equally and use only the previous round, but it may be reasonable in other applications to alter the weighting scheme and number of rounds used.

- **Early stopping based on Bayesian Linear Regression:** In an attempt to quantify how the prediction uncertainly changes as θ changes, we also implement Bayesian Linear Regression, found in [Riquelme et al., 2018] [2] to be a simple but effective measure of

[1] Bobu, Andreea, et al. "Adapting to continuously shifting domains." (2018).

[2] Riquelme, Carlos, George Tucker, and Jasper Snoek. "Deep bayesian bandits showdown." *International Conference*

uncertainty, over the last layer of $\hat{h}(\phi_\theta)$ as we vary θ through training. We find that in early iterations of training, this can be an effective stopping criterion for training of ϕ. Again, as training progresses, we find that this mostly indicates only small changes in model uncertainty.

A.1.4 Human Input

Testing on mTurk presents various challenges for testing the M$\circ$M framework:

- In some applications, such as loan approval, mTurk users are not experts. This makes it difficult to convince them that anything is at stake (we found that bonuses did not meaningfully affect performance). It is also difficult to directly measure effort, agency, trust, or autonomy, all of which result in higher variance in responses.
- In many other applications, the ground truth is generated by humans to begin with (for example, sentiment analysis). Since we require ground truth for training, in these task it cannot be expected of humans to outperform machines.
- As the researchers found in [Lag+18], there can be a large variance in the time users take to complete a given task. Researchers have found that around 25% of mTurk users complete several tasks at once or take breaks during HITs [Moss and Litman, 2019].[3] making it difficult to determine how closely Turkers are paying attention to a given task. We use requirements of HIT approval rate greater than 98%, US only, and at least 5,000 HITs approved, as well as a simple comprehension check.
- Turker populations can vary over time and within time periods, again leading to highly variable responses, which can considerably effect the performance of learning.
- Recently, there have been concerns regarding the usage of automated bots within the mTurk communiy. Towards this end, we incorporated in the experimental survey a

on Learning Representations. 2018.

[3] A. J. Moss and L. Litman. How do most mturk workers work?, Mar 2019.

required reading comprehension task and as well as a CAPTCHA task, and filtered users that did not succeed in these.

A.2 Experimental Details

A.2.1 Decision-compatible 2D projections

In the experiment, we generate 1,000 examples of these point clouds in 3D. The class of ϕ is a 3x3 linear layer with no bias, where we add a penalization term on $\phi^T\phi - \mathbb{I}$ during training to constrain the matrix to be orthogonal. Humans are shown the result of passing the points through this layer and projecting onto the first two dimensions. The class of $\hat{h}$ is a small network with 1 3x3 convolutional layer creating 3 channels, 2x2 max pooling, and a sigmoid over a final linear layer. The input to this network is a soft (differentiable) 6x6 histogram over the 2D projection shown to the human user.

We tested an interactive command line query and response game on 12 computer science students recruited on Slack and email. Users filled out a consent form online, watched an instructional video, and then completed a training and testing round, each with up to 5 rounds of 15 responses. Due to the nature of the training process, achieving 100% accuracy results in ϕ not updating in the following round. With this in mind, if a user reached 100% accuracy in training, they immediately progressed to testing. If a user reached 100% accuracy in testing, the program exited. ϕ was able to find a representation that allowed for 100% accuracy 75% of the time, with an average 5 round improvement of 23% across all participants. Many times the resulting projection appeared to be an ‘x’ and ‘o’, as in Figure A.1, but occasionally it was user-specific. For example, a user who associates straight lines with the ‘x’ may train the network to learn any projection for ‘x’ that includes many points along a straight line.

The architecture of ϕ and $\hat{h}$ are described in Section 2.4. For training, we use a fixed number of epochs (500 for $\hat{h}$ and 300 for ϕ) with base learning rates of .07 and .03, respectively, that increase with lower accuracy scores and decrease with each iteration. We have found these parameters to work well in practice, but observed that results were not sensitive to their selection.

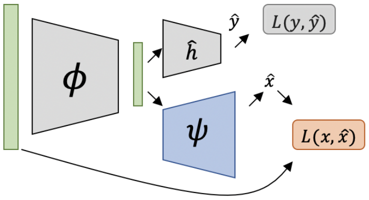


Figure A.2: *Visualization of reconstruction component*

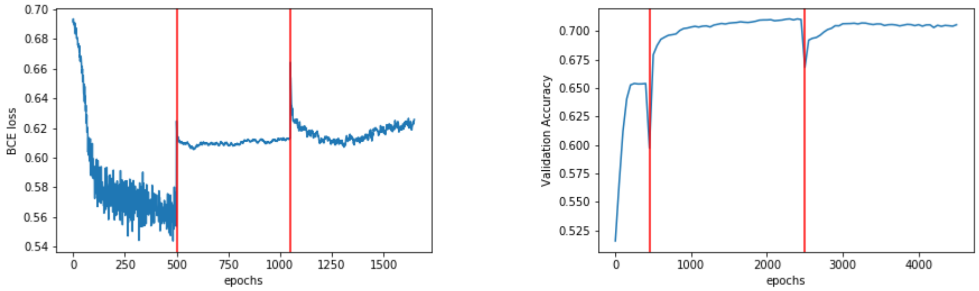
A.2.2 Decision-compatible algorithmic avatars

Data Preprocessing.

We use the *Lending Club* dataset, which we filter to include only loans for which we know the resolution (either default or paid in full, not loans currently in progress) and to remove all features that would not have been available at funding time. We additionally drop loans that were paid off in a single lump sum payment of at least 5 times the normal installment. This results in a dataset that is 49% defaulted and 51% repaid loans. Categorical features are transformed to one-hot variables. There are roughly 95,000 examples remaining in this dataset, of which we split 20% into the test set.

Learning architecture and pipeline.

The network ϕ takes as input the standardized loan data. Although the number of output dimension are $\mathbb{R}^9$, ϕ outputs vectors in $\mathbb{R}^{11}$. This is because the some facial expressions do not naturally coexist as compound emotions, i.e., happiness and sadness [DTM14]. Hence, we must add some additional constraints to the output space, encoded in the extra dimensions. For example, happiness and sadness are split into two separate parameters (rather than using one dimension with positive for happiness and negative for sadness). The same is true of “happy surprise", which is only allowed to coincide with happiness, as opposed to “sad surprise." For parameters which have positive and negative versions, we use a tanh function as the final nonlinearity, and for parameters which are positive only, we use a sigmoid function as the final nonlinearity.



(a) *Loss in training $\hat{h}$ over 3 rounds*

(b) *Validation Accuracy in training ϕ over 3 rounds*

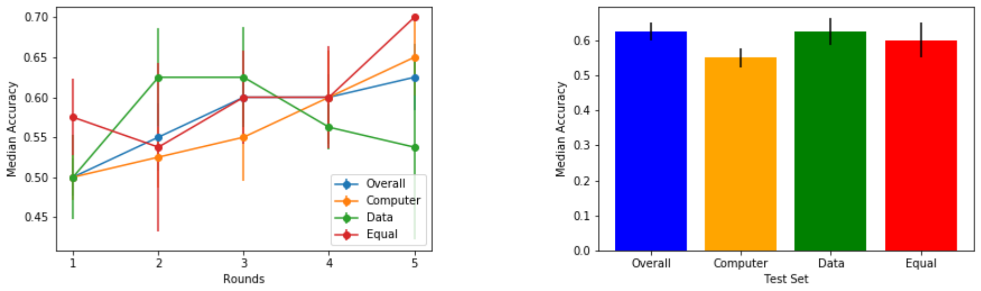
Figure A.3: *$\hat{h}$ does not necessarily have to match h well to lead to an increase in accuracy*

These parameters are programmatically mapped to a series of Webmorph [DT16] transformation text files, which are manually loaded into the batch transform/batch edit functions of Webmorph. We use base emotion images from the CFEE database [DTM14] and trait identities from [OT08]. This forms ρ for this experiment.

The network ϕ is initialized with a WGAN to match a distribution of parameters chosen to output a fairly uniform distribution of feasible faces. To achieve this, each parameter was chosen to be distributed according to one of the following: a clipped $\mathcal{N}(0,4)$, $\mathcal{U}[0,1]$, or Beta(1,2). The choice of distribution was based on inspection as to what would give reasonable coverage over the set of emotional representations we were interested in testing. In this initial version of ϕ, x values end up mapped randomly to representations, as the WGAN has no objective other than distribution matching.

The hidden layer sizes of ϕ and $\hat{h}$ were chosen via cross validation. For ϕ, we use the smallest architecture out of those tested capable of recreating a wide distribution of representations z as the generator of the WGAN. For $\hat{h}$, we use the smallest architecture out of those tested that achieves low error both in the computer-only simulation and with the first round of human responses.

In the first experiment, we collect approximately 5 labels each (with minor variation due to a few mTurk users dropping out mid-experiment) for the LASSO feature subset of 400 training set x points and their ϕ_0 mappings (see Figure A.5). a is taken to be the percentage of users



(a) *Training Rounds ('Overall' here is average* per user *score, rather than the score of the average response per question)*

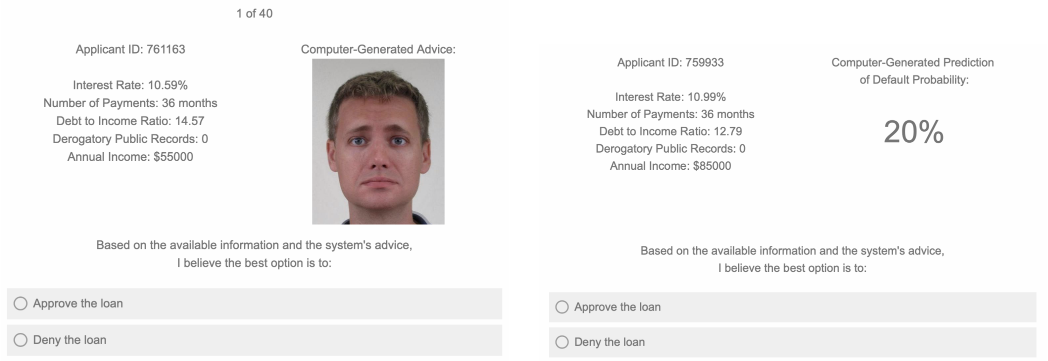
(b) *Test Round*

Figure A.4: *Results by Reported User Type*

responding "approve" for each point.

To train $\hat{h}$, we generate 15 different training-test splits of the collected $\{z, a\}$ pairs and compare the performance of variations of $\hat{h}$ in which it is either initialized randomly or with the $\hat{h}$ from the previous iteration, trained with or without adding the samples from the previous iteration, and ranging over different regularization parameters. We choose the training parameters and number of training epochs which result in the lowest average error across the 15 random splits. In the case of random initialization, we choose the best out of 30 random seeds over the 15 splits.

To train ϕ, we fix $\hat{h}$ and use batches of 30,000 samples per epoch from the training set,



1 of 40

Applicant ID: 761163

Computer-Generated Advice:

Interest Rate: 10.59%
Number of Payments: 36 months
Debt to Income Ratio: 14.57
Derogatory Public Records: 0
Annual Income: $55000

Based on the available information and the system's advice,
I believe the best option is to:

○ Approve the loan

○ Deny the loan

Applicant ID: 759933

Computer-Generated Prediction
of Default Probability:

Interest Rate: 10.99%
Number of Payments: 36 months
Debt to Income Ratio: 12.79
Derogatory Public Records: 0
Annual Income: $85000

20%

Based on the available information and the system's advice,
I believe the best option is to:

○ Approve the loan

○ Deny the loan

(a)

(b)

Figure A.5: *Images from mTurk questionnaire*

which has 75,933 examples in total. To prevent mode collapse, wherein faces “binarize" to two prototypical exemplars, we add a reconstruction regularization term $R(x) = \|x - \psi(\phi(x))\|_2^2$ to the binary cross entropy accuracy loss, where ψ is a decoder implemented by an additional neural network (see Figure A.2). ϕ here also features a constraint penalty that prevents co-occurrence of incompatible emotions.

We train ϕ for 2,000 epochs with the Adam optimizer for a variety of values of α, where we use α to balance reconstruction and accuracy loss in the form $\mathcal{L}_{total} = \alpha\mathcal{L}_{acc} + (1 - \alpha)\mathcal{L}_{rec}$. We choose the value of α per round that optimally retains x information while promoting accuracy by inspecting the accuracy vs. reconstruction MSE curve. We then perform Bayesian Linear Regression over the final layer of the current $\hat{h}$ for every 50th epoch of ϕ training and select the number of epochs to use by the minimum of either 2,000 epochs or the epoch at which accuracy uncertainty has doubled. In all but the first step, this resulted in using 2,000 epochs. At each of the 2-5th epochs, we choose only 200 training points to query. In the 6th epoch we use 200 points from the test set.

Self-reported user type.

In the end of the survey, we ask users to report their decision method from among the following choices:

- I primarily relied on the data available
- I used the available data unless I had a strong feeling about the advice of the computer system
- I used both the available data and the advice of the computer system equally
- I used the advice of the computer system unless I had a strong feeling about the available data
- I primarily relied on the advice of the computer system
- Other

The percentage of users in each of these groups varied widely from round to round.

We consider the first two conditions to be the 'Data' group, the third to be the 'Equal' group, and the next two to be the 'Computer Advice' group. Although the trend is not statistically significant (at $p = 0.05$), likely due to the small number of subjects per type per round, we find it interesting that the performance improved on average over training rounds for all three types, of which the equal-consideration type performed best. For the data-inclined users, whose performance improved to surpass that of the no-advice condition in as early as round two, this implies at least one of the following: users misreport their decision method; users believe they are not influenced by the advice but are in fact influenced; or, as the algorithmic evidence becomes apparently better, only the population of users who are comparatively skilled at using the data continue to do so.

Diversity in avatar representation.

Figure A.6 presents examples of visualized avatars. Avatars correspond to examples having either low or high human-predicted probability (averaged across users) (top figure), and either low or high machine-predicted probability (lower figure). For visualization purposes, avatars are aligned according to a uni-dimensional PCA projection of the inputs, so that their spatial positioning captures the variance in the data. As can be seen, avatars are different for each predictive category (positive or negative; human or machine), but also vary considerably within each predictive category, with variance eminent across multiple facial dimensions.

We believe the additional dimensionality of the avatar representation relative to a numerical or binary prediction of default is useful for two reasons. Most importantly, high dimensionality allows users to retain an ability to reason about their decisions. In particular, avatars are useful because people likely have shared, mental reference points for faces. Moreover, users with a more sophisticated mental reference space may be able to teach the advising system over time to match specific reasoning patterns to specific characteristics. Additionally, when the advising system does not have a strong conviction about a prediction, presenting neutral advice should encourage the user to revisit the data, whereas percentages above or below the base rate of default (or 50%) may suffer from the anchoring effect.

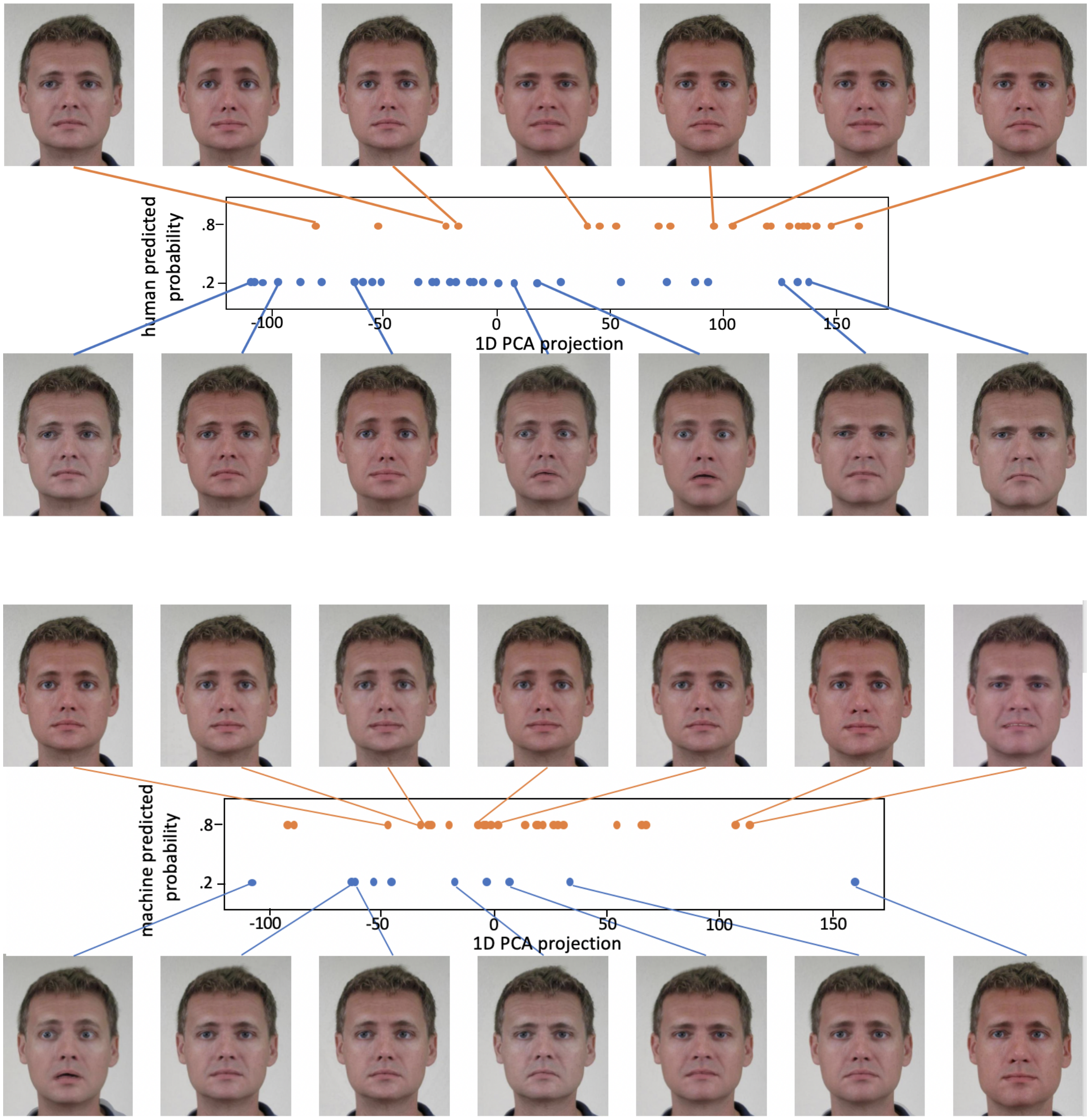


Figure A.6: *Richness of avatar representation. A visualization of 200 avatars randomly sampled from the held-out test set, grouped by either human (top) or machine (bottom) predictive probability (0.2 in blue, 0.8 in orange, with a tolerance of 0.05). Avatars are positioned based on a 1D PCA dimensionality reduction of their corresponding feature vectors z, along which a 'gradient' of facial changes can be observed.* ***Top****: Here avatars are grouped by human predictive probability. The figure shows how for the same human decisions, learning results in avatars of varied and complex facial expressions, conveying rich high-dimensional information. Interestingly, avatars corresponding to loan denial exhibit more variance, suggesting that there may be more 'reasons' for denying a loan than for approving one.* ***Bottom****: Here avatars are grouped by machine predictive probability. Since all examples in each group have the same predictive probability, they are equally similar, which does not facilitate a clear notion for reasoning. In contrast, avatars maintain richness in variation, and can be efficiently used for reasoning (e.g., via similarity arguments) and other downstream tasks.*

Further Details on Information Learned by z.

Using cross-validated ridge regression to predict individual x variables from individual z variables results in the coefficients of determination R^2 (to 2 significant figures) shown in Table A.1.

Using cross-validated ridge regression to predict individual x variables from all z variables (both standardized to mean 0, std 1) results in the *variable coefficients* (to 2 significant figures) shown in Table A.2.

A.2.3 Incorporating Side Information

$y = x_c + x_d + s; \; ; \; y_{bin} = \mathbb{1}\{y > 3\}$

Learning Architecture.

The network ϕ contains a single linear layer with no bias which takes a constant (1) as an input and outputs a number z_i for each data dimension i.

Table A.1: *Coefficients of Determination R^2, predicting each x variable from each final z variable.*

	RATE	TERM	DT	REC	INC	EMP
happiness	0.00	-0.15	-0.14	0.00	-0.01	0.00
sadness	-0.01	-0.06	-0.10	0.00	-0.04	-0.07
trustworthiness	0.57	0.17	0.01	0.00	-0.01	-0.01
dominance	0.00	-0.01	0.03	-0.01	0.01	-0.01
hue	0.48	0.29	-0.02	0.00	-0.04	-0.02
eye gaze	0.42	0.46	-0.04	-0.40	-0.04	-0.17
age	0.23	0.22	-0.12	-0.21	0.17	0.04
anger	-0.01	-0.02	-0.05	-0.02	-0.01	0.00
fear	0.04	0.00	-0.03	0.00	-0.01	-0.01
surprise	-0.18	0.04	-0.01	-0.02	0.00	-0.04

Table A.2: *Coefficients of Ridge Regression, predicting each x variable from all final z variables.*

	RATE	TERM	DT	REC	INC	EMP
happiness	-0.07	-0.29	-0.10	-0.06	0.21	-0.07
sadness	0.16	0.07	0.07	-0.01	0.13	0.07
trustworthiness	-0.62	-0.28	-0.05	-0.23	0.31	0.16
dominance	0.05	0.16	0.12	-0.13	-0.02	0.04
hue	0.27	0.20	0.19	0.03	0.01	-0.08
eye gaze	0.13	0.28	-0.10	0.13	-0.29	-0.04
age	-0.09	0.14	0.12	-0.09	0.67	0.40
anger	0.00	0.00	0.00	0.00	0.00	0.00
fear	0.19	0.12	0.08	-0.07	0.04	0.00
surprise	0.07	0.12	0.03	-0.07	-0.06	0.13

The network $\hat{h}$ takes as input (x, w, y). It contains one linear layer with no bias which takes as input $[x, y]$ and outputs a single number $\hat{s}$. The second linear layer (with bias) takes as input w and outputs the sigmoid activation of a single number, *switch*, representing the propensity to incorporate s at w. It then outputs $w^\intercal x + switch \cdot \hat{s}$.

Baselines.

- **Machine Only**: The best possible linear model (with bias) trained to predict y from $x_1 \ldots x_4$.
- h(**Machine)**: The human model h applied to the best possible linear model (with bias) trained to predict y from $x_1 \ldots x_4$.

$$h(\text{Machine}) = \beta_0 + h(x, \beta_1, \ldots, \beta_4, s)$$

  where β are the coefficients selected by the machine-only regression.

Human Models

- **Always**: The human always fully incorporates the side information,

$$h(x, w, s) = w^\intercal x + s$$

- **Never**: The human never incorporates the side information,

$$h(x, w, s) = w^\intercal x$$

- **Or**: The human becomes less likely to incorporate side information as weight is put on x_i, x_r,

$$h(x, w, s) = w^\intercal x + \sigma(1/\max(\max(x_i, x_r), .0001) - 2). \cdot s$$

  Note that max(.0001) is required to prevent numerical overflow, and -2 recenters the sigmoid to allow for values $< .5$.

- **Coarse**: The human incorporates s as in Or, but uses a coarse, noisy version of s, $s' = 2 \cdot \mathbb{1}\{s \geqslant 2\}$

$$h(x, w, s) = w^{\intercal}x + \sigma(1/\max(\max(x_i, x_r), .0001) - 2). \cdot s'$$

A.3 Select Turker quotes

- "I wasn't always looking at just happiness or sadness. Sometimes the expressions seemed disingenuously happy, and that also threw me off. I don't know if that was intentional but it definitely effected my gut feeling and how I chose."
- "In my opinion, the level of happiness or sadness, the degree of a smile or a frown, was used to represent applications who were likely to be payed back. The more happy one looks, the better the chances of the client paying the loan off (or at least what the survey information lead me to believe)."
- "I was more comfortable with facial expressions than numbers. I felt like a computer and I didn't feel human anymore. Didn't like it at all."

Appendix B

Appendix to Chapter 3

B.1 Pseudocode

Our algorithm alternates between optimizing the three components of the framework: a predictive model, a propensity model, and an uncertainty model. Here we give pseudocode for the following per-component objectives:

1. A predictive model $\hat{y} = f(x)$, optimizing the squared loss:

$$\ell_{\text{pred}}(f; \mathcal{S}) = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

2. A propensity weight model $w = e^{h(x)}$, optimizing the log-loss:

$$\ell_{\text{prop}}(h; \mathcal{S}, \mathcal{S}') = \sum_{i=1}^{m} \log(1 + e^{h(x_i)}) + \log(1 + e^{-h(x_i')})$$

3. An uncertainty interval model $[\ell, u] = g_\tau(x)$, optimizing the τ-quantile loss:

$$\ell_{\text{uncert}}^{(\tau)}(g; \mathcal{S}, w) = \sum_{i=1}^{m} w(x_i) \max\{(\tau - 1)(y_i - \ell_i), \tau(y_i - \ell_i)\}$$

but note that others can be plugged in. The pseudocode is given below.

Algorithm B.1: Lookahead($\mathcal{S}, T, \lambda, \eta, \tau$)

```
$f^{(0)} \leftarrow \operatorname{argmin}_{f \in F} \ell_{\text{pred}}(f; \mathcal{S})$
for $t = 1, \dots, T$ do
    $x'_i \leftarrow d_\eta(x_i; f^{(t-1)})$ for all $i = 1, \dots, m$ {e.g., $d_\eta(x; f) = x + \eta\Gamma(\nabla_f(x))$}
    $\mathcal{S}' \leftarrow \{x'_i\}_{i=1}^m$
    $h^{(t)} \leftarrow \operatorname{argmin}_{h \in H} \ell_{\text{prop}}(h; \mathcal{S}, \mathcal{S}')$
    $w \leftarrow e^{h^{(t)}}$
    $g^{(t)} \leftarrow \operatorname{argmin}_{g \in G} \ell^{(\tau)}_{\text{uncert}}(g; \mathcal{S}, w)$
    $f^{(t)} \leftarrow \operatorname{argmin}_{f \in F} \ell_{\text{pred}}(f; \mathcal{S}) + \lambda R(g^{(t)}; \mathcal{S})$
end for
return $f^{(T)}$
```

B.2 Uncertainty models

Here we describe the two uncertainty methods used in our paper and how they apply to our setting.

B.2.1 Bootstrapping

Bootstrapping produces uncertainty intervals by combining the outputs of a collection of k models $\{g^{(i)}\}_{i=1}^k$, each trained independently for *prediction* on a random subset of the data. There are many approaches to bootstrapping, and here we describe two:

- **Vanilla bootstrapping**: Each $g^{(i)}$ is trained using a predictive objective (e.g., squared loss) on a sample set $\mathcal{S}^{(i)} = \{(x_j^{(i)}, y_j^{(i)})\}_{j=1}^m$ where $(x_j^{(i)}, y_j^{(i)})$ are sampled with replacement from $\mathcal{S}$. The sub-models are then combined using:

$$g(x) = [\mu(x) - z\sigma(x), \mu(x) + z\sigma(x)]$$

where:

$$\mu(x) = \frac{1}{k}\sum_{i=1}^k g^{(i)}(x), \qquad \sigma(x) = \frac{1}{k}\sum_{i=1}^k (\mu(x) - g^{(i)}(x))^2$$

and z is the z-score corresponding to the confidence parameter τ under a normal distribution.

- **Bootstrapping residuals**: First, a predictive model $\bar{g}$ is fit to the data, and residuals $r = y - \bar{g}(x)$ are computed. Then, each $g^{(i)}$ is trained on the original sample data but with ground truth-labels y_i replaced with random pseudo-labels:

$$\mathcal{S}^{(i)} = \{(x_j, \bar{y}_j^{(i)})\}_{j=1}^m \qquad \bar{y}_j^{(i)} = y_i + r_j$$

where r_j are sampled with replacement from $\{r_j\}_{j=1}^m$.

In our framework, because g must apply to p', each $g^{(i)}$ is trained with propensity weights w. To account for cases where p and p' differ, the $g^{(i)}$ are trained not on sample sets of size m, but rather, of size $\tilde{m}(w)$, where $\tilde{m}(w)$ is the *effective sample size* [KLW94] given by:

$$\tilde{m}(w) = \frac{\text{mean}(\{w_i\}_{i=1}^m)}{\text{var}(\{w_i\}_{i=1}^m)}, \qquad w_i = w(x_i) \;\; \forall\, i = 1, \ldots, m$$

B.2.2 Quantile regression

Quatile regression is a learning framework for training models to predict the τ-quantile of the conditional label distribution $p(y|x)$. Just as training with the squared loss is aimed at predicting the mean of $p(y|x)$, training with the absolute loss $|y - \hat{y}|$ is aimed at the median. Quantile regression generalizes the absolute loss by considering a 'tilted' variant with slopes $\tau - 1$ and τ:

$$Q_\tau(y, \hat{y}) = \max\{(1-\tau)(y-\hat{y}), \tau(y-\hat{y})\}$$

B.3 Experimental details

B.3.1 Experiment 1: Quadratic curves

Here we set $f^*(x) = -0.8x^2 + 0.5x + 0.1$. F and G include quadratic functions, and H to include linear functions. For uncertainty estimation we used vanilla bootstrap, and for propensity scores we used logistic regression. For lookahead, we set $\lambda = 4$, $\tau = 0.95$, use $k = 10$ bootstrapped models, and train for $T = 5$ rounds. The data includes $m = 25$ samples x drawn from $N(-0.8, 0.5)$, and $y = f^*(x) + \epsilon$ where $\epsilon \sim N(0, 0.25)$. We use a $75 : 25$ train-test split. The three conditions vary only in η with values $\eta = 0.75, 1.25$, and 3.5.

Quantitative results are given in the table below:

		RMSE	Imp. rate	Imp. mag.
$\eta = 0.75$	baseline	0.349	0.857	1.109
	lookahead	0.351	0.857	1.108
$\eta = 1.25$	baseline	0.342	0.143	-0.261
	lookahead	0.424	0.714	1.065
$\eta = 3.5$	baseline	0.342	0	-35.13
	lookahead	0.675	0.571	0.604

B.3.2 Experiment 2: Wine quality

The wine dataset includes $m = 178$ examples and $d = 13$ features. We learn a quadratic $f^*(x) = \sum_i \theta_i x_i + \sum_i \theta'_i x_i^2$. F, G, and H include linear functions. For uncertainty estimation we used residuals bootstrap, and for propensity scores we used logistic regression. For lookahead, we set $\tau = 0.95$, use $k = 20$ bootstrapped models, and train for $T = 10$ rounds. For f, we use SGD with a learning rate of 0.1 and 1000 epochs for initialization and 100 additional epochs per round. For g, each sub-model was trained with SGD using a learning rate of 0.1 and for 500 epochs. We set $\eta = 0.5$ and $\eta = 2$ for the fully and partially mutable settings, respectively.

B.3.3 Experiment 3: Diabetes

The diabetes dataset includes $m = 442$ examples and $d = 10$ features. We set $f^*(x)$ to be a generalized additive model (GAM) with splines of degree 10 trained on the entire dataset and tuned using cross-validation. In the first setting, F, G, and H include linear functions. In the second setting, F, G are quadratic functions (i.e., linear in x_i and in x_i^2) and H remains linear. For uncertainty estimation we used quantile regression, and for propensity scores we used logistic regression. For lookahead, we set $\tau = 0.8$ and train for $T = 10$ rounds. For f, we use SGD with a learning rate of 0.05 and 1000 epochs for initialization and 100 additional epochs per round. For g, we use SGD with a learning rate of 0.05 and for 500 epochs. For both linear and non-linear settings we set $\eta = 5$, and normalize y to be in $[0, 1]$.

B.3.4 Sensitivity analysis

The experiments in the paper assume models are trained with the same η used in evaluation. Here we evaluate the sensitivity of our method to the misspecification of η. We use the diabetes

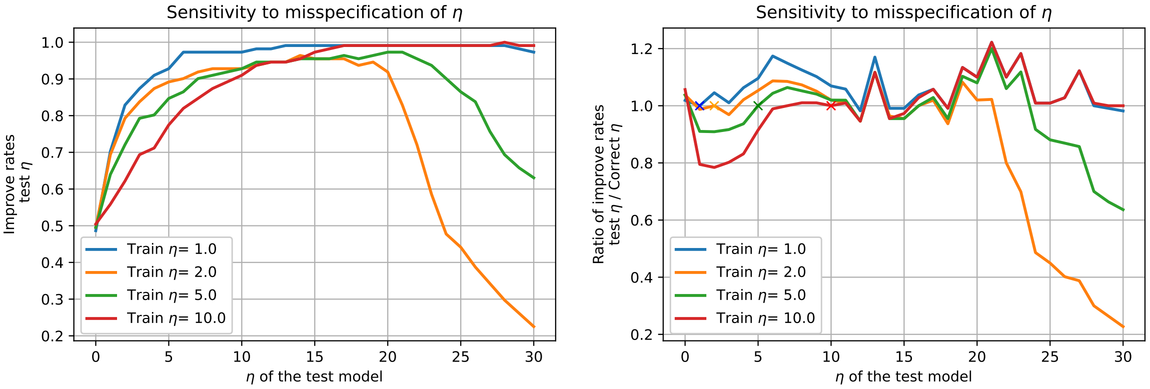


Figure B.1: *Sensitivity analysis of learning with misspecified η. (Left) Improvement rate of misspecified models, trained on a single η and evaluated on data generated with varying values of η. (Right) The ratio between the improvement rate of misspecified and correctly-specified models (i.e., trained on the same η on which they are evaluated).*

experimental setup, train four different models with $\eta \in \{1, 2, 5, 10\}$, and evaluate each on action outcomes generated with $\eta' \in \{0, \ldots, 30\}$. Figure B.1 (left) shows the improvement rate of each model evaluated on varying test-time η'. As can be seen, improvement rates across η' show an inverse-U patter. In most regimes performance is robust, although for large deviations between η and η' improvement rates deteriorate. To investigate this, in Figure B.1 (right) we compare the improvement rate of the misspecified model (i.e., trained on a fixed η) to that of a correctly-specified model, and report the ratio.[1] The correctly-specified model serves as a benchmark on performance, and results show that misspecified models are competitive with this benchmark (except for extremely large values of η').

[1] Due to randomness in experimentation, the ratio can be larger than one.

Appendix C

Appendix to Chapter 4

Symbol	Description
x_i	Observed attributes of a data point i
y_i	ground truth class label of a data point i
$\mathcal{X}$	Set of all the observed attributes of input data points i.e., $\mathcal{X} = \{x_1, x_2, \cdots x_N\}$
$\boldsymbol{y}$	Set of all the ground truth class labels of input data points i.e., $\boldsymbol{y} = \{y_1, y_2, \cdots y_N\}$
$\mathcal{D}$	Set of input data points $\mathcal{D} = (\mathcal{X}, \boldsymbol{y}) = \{(x_1, y_1), (x_2, y_2) \cdots (x_N, y_N)\}$
$\mathcal{C}$	Set of class labels in $\mathcal{D}$ i.e., $\forall i$, $y_i \in \mathcal{C}$
$\mathcal{X}_{dist}$	Distribution from which $\mathcal{X}$ is sampled
$\mathcal{D}_{dist}$	Distribution from which $\mathcal{D}$ is sampled
f	Black box model which maps a data point to a class label i.e., $f(x_i) \in \mathcal{C}$
g	Interpretable model that serves as an explanation of the black box model f generated by posthoc explanation techniques
ψ	Unbiased classification function
e	Adversarial classifier
$\mathcal{X}_p$	Set of perturbed data points generated from $\mathcal{X}$

Table C.1: *Description of Notation*

COMPAS LIME Adversarial Classifier

	Baseline classifier f	Attack 1 feature			Attack 2 features		
Importance Ranking	1	1	2	3	1	2	3
African-American	100	0	9	11	0	0	11
Unrelated Feature 1	0	100	0	0	49	51	0
Unrelated Feature 2	0	0	9	10	50	49	0
Other Features	0	0	82	79	0	0	89
Accuracy	56		56			56	

COMPAS SHAP Adversarial Classifier

	Baseline classifier f	Attack 1 feature			Attack 2 features		
Importance Ranking	1	1	2	3	1	2	3
African-American	100	16	82	1	34	31	33
Unrelated Feature 1	0	84	12	1	35	31	19
Unrelated Feature 2	0	0	0	13	32	31	18
Other Features	0	0	6	55	0	5	20
Accuracy	56		53			55	

Table C.2: *The percentage occurrence of the top features in e with the perfectly unfair f (if African-American predict high likely to recidivate) on COMPAS LIME and SHAP adversarial explanations. "Attack 1 feature" and "Attack 2 features" can be read as two different scenarios: the former where one feature is used in ψ and the latter where two features are used in ψ. The baseline is the explanation on f. The percentages are the mean rates of occurrences on a 10% test set over three different runs (including re splitting into a different train-test set). We see that whether the individual is African American is the most important feature in both LIME and SHAP on f. Using e, we are able to exclude all of the explanations of this feature in LIME and most of the explanations in SHAP in the top feature positions.*

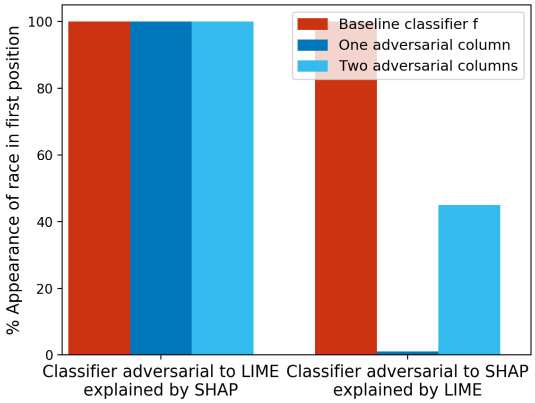


Figure C.2: *The effectiveness of explaining classifiers adversarial to SHAP with LIME and classifiers adversarial to LIME with SHAP. We see that classifiers adversarial to LIME are not effective at hiding bias from SHAP explanations, and classifiers adversarial to SHAP are moderately effective at hiding biases from LIME explanations.*

Communities and Crime LIME Adversarial Classifier							
	Baseline classifier f	Attack 1 feature			Attack 2 features		
Importance Ranking	1	1	2	3	1	2	3
Race % White	100	0	1	1	0	0	2
Unrelated Feature 1	0	100	0	0	48	52	0
Unrelated Feature 2	0	0	3	3	52	48	0
Other Features	0	0	96	96	0	0	98
Accuracy	73		73			73	

Communities and Crime SHAP Adversarial Classifier							
	Baseline classifier f	Attack 1 feature			Attack 2 features		
Importance Ranking	1	1	2	3	1	2	3
Race % White	100	0	78	3	26	26	40
Unrelated Feature 1	0	100	0	0	36	25	7
Unrelated Feature 2	0	0	0	3	35	30	6
Other Features	0	0	16	86	0	16	44
Accuracy	73		70			72	

Table C.3: *The percentage occurrence of the top features in e with the perfectly unfair f (if race % white $>$ median race % white predict nonviolent community) on Communities and Crime. Using the e, we are able to exclude all of the explanations of this feature in LIME and many of the SHAP explanations, consistent with our results on COMPAS.*

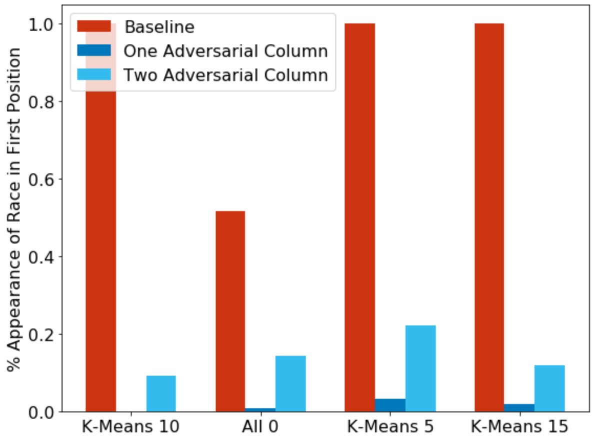


Figure C.1: *SHAP attack effectiveness across different background distributions: K-means 10 is the distribution assumed in training. We also test on K-means 5, K-means 15, and all 0. These represent different suggestions in the SHAP software package for representing a large dataset.*

German Credit LIME Adversarial Classifier				
	Baseline classifier f	With Attack		
Importance Ranking	1	1	2	3
Gender	100	0	4	4
Loan Rate % Income	0	91	0	0
Other Features	0	9	96	96
Accuracy	64	64		

German Credit SHAP Adversarial Classifier				
	Baseline classifier f	With Attack		
Importance Ranking	1	1	2	3
Gender	100	0	5	1
Loan Rate % Income	0	85	0	0
Other Features	0	0	72	65
Accuracy	64	64		

Table C.4: *The percentage occurrence of the top features in e with the perfectly unfair f (if Gender is male predict will repay loan) on COMPAS LIME and SHAP explanations. We use loan rate as a percentage of income as ψ and predict false if the value is above its mean. In both the LIME and SHAP case, we are able to exclude gender from the majority of the explanations. When the explanation is included, it appears at the same frequency as other features.*