Towards Trustworthy Machine Learning in High-Stakes Decision-Making Systems

by

David Madras

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy

Graduate Department of Computer Science
University of Toronto

Towards Trustworthy Machine Learning in High-Stakes Decision-Making Systems

David Madras
Doctor of Philosophy

Graduate Department of Computer Science
University of Toronto
2022

Abstract

Machine learning systems are increasingly applied to tasks of *high-stakes decision-making*, in areas like healthcare, personal finance, and criminal justice. In these areas, the *trustworthiness* of the system is essential; beyond just having a high-accuracy tool, a number of other desiderata are required. Stressing that creating trustworthy machine learning is a problem without a well-defined solution, we focus in particular on two of these desiderata: fairness and robustness. To these ends, we discuss three potential approaches to improving the trustworthiness of machine learning systems. In the first, we consider the task of learning representations which guarantee that downstream classifiers yield predictions that align with fairness metrics when trained on transfer tasks, demonstrating how designing an adversarial loss function can correspond to various commonly-used fairness metrics. In the second, we provide an intuitive heuristic for detecting underspecification, show how it can be computed in a post-hoc fashion using only second-order statistics of the trained model, and discuss how we navigate computational hurdles for calculating this score in deep neural networks using techniques for eigenspectrum estimation. In the third, we reframe rejection learning as a task which should be inherently adaptive, depending on the properties of external decision-makers, and show how to formulate this as a mixture-of-experts-type learning objective, studying its impact on both fairness and accuracy. Through theoretical analysis and empirical evidence, we examine the strengths and weaknesses of each approach, and critically discuss how each fits into the larger project of building trustworthy machine learning systems.

Acknowledgements

First and foremost, this thesis would not have been possible without my supervisor Rich Zemel. Throughout my PhD, I've always appreciated Rich's laid-back attitude, never pushing too hard but instead letting my research interests develop over time. Rich allowed me to explore the relatively new areas of machine learning early on in my PhD, whether it was our first dives into causal inference or some "fairness" thing that nobody could really make head or tails of at the time. This openness set the tone for the next few years, and it's been a joy to explore these areas alongside Rich as the whole community has tried to get a handle on these complex questions. Above and beyond that, Rich has been a fantastic role model — demonstrating how to be serious about work but also to have fun, how to have a great work-life balance and also be a world-class researcher. Rich opened up a number of really interesting opportunities for me — getting to travel to Rwanda with him (and Toni and Elliot) for three weeks in 2019 was a highlight, a truly enjoyable and unique experience I'll never forget. Through two degrees, six work locations, and a global pandemic, Rich enabled me to see many of the best sides of grad school and protected me from many of the worst. I didn't really know what I was getting into when I picked a supervisor and grad program (I think most students don't), but I don't think I could have done much better. [1]

Thanks also to Marzyeh Ghassemi and Roger Grosse for being part of my PhD committee, whose feedback on my work I've really appreciated throughout the past several years, despite the obstacles created by constantly shifting geography. I continue to deeply value their perspectives and to be inspired by the work they do. Additional thanks are due as well to my external thesis examiner Adrian Weller, for his thoughtful feedback and expertise.

I've benefited from a number of other senior mentors along the way. Thanks to Toni Pitassi, who I had the joy of working with and learning from in the first few years of my PhD, as we all tried to wrap our heads around questions of fairness, and whose intelligence, kindness, and insight have always astounded me. Also thanks to Toni for joining my final PhD committee at the last minute! Thanks also to my mentors at Google: Alex D'Amour, who taught me a whole lot about causality, statistics, and a number of other subjects every time he opened his mouth; and James Atwood, who was instrumental in settling me in and helping me wrestle with the internal infrastructure at Google. The two of them (and the Cambridge fairness group) were essential in making me feel welcomed and supported throughout my internship, and left a great impression on me of what working in industry could be like.

One of the most important aspects of research in a PhD is other students you collaborate closely with. Thanks to Elliot Creager for being my most frequent collaborator throughout my PhD, who I benefited a ton from learning and exploring alongside, from the best ways to think about fairness, to diving into causality, to picking up pieces of domain generalization and robustness. I'm always amazed by his generosity, commitment to detail and drive for justice. This PhD wouldn't have been the same without the opportunity to work with him, both when we were on papers together, and when I needed feedback or was struggling with some new concept. Thanks also to Aparna Balagopalan for her tireless work driving our project and whom it has been a pleasure to watch grow throughout the past couple of years. Coordinating a large project is very challenging, and I've been impressed with her great ideas as well as her ability to lead. Thanks as well to Sindy Löwe, who made

[1] To that end, thanks are in order to the 8 programs across CS and engineering departments in the USA where I was not accepted to grad school in 2016 — the situation I ended up in was likely preferable :)

the process of revising a paper after rejection four times as enjoyable as it could possibly be: I greatly appreciate both her technical ability as well as her easygoing nature. Thanks to my collaborators on leadership projects: Amy Zhang and Blanca Miller for co-organizing the Pan-Canadian SOCMLx conference in 2019 (which I hope gets picked up again post-pandemic at some point), and to Bogdan Kulynych, Smitha Milli, Deb Raji, and Angela Zhou for co-organizing the Participatory Approaches to Machine Learning workshop at ICML 2020. Both projects were (in my opinion) really successful and enjoyable learning experiences, and my graduate school experience has been so much richer for that.

This list isn't complete without a big thanks to the rest of my friends and labmates at grad school, who were a big part of so many of best moments of the past six years: Eleni, Will, Shems, Jesse, Jackson, Marc, James, Jake, Kamyar, Micha, Mengye, Renjie, Yasmin, Stavros, Ariel, Alex, Ben, Robert, and others I'm sure I've forgotten to include here. Grad school has the potential to be an extremely lonely, isolating time and because of these and many other colleagues and friends, it was never that way for me.

Thank you to other faculty members and mentors at the University of Toronto who have helped me immeasurably throughout, not only my nearly six years of grad school, but five years of undergraduate education as well. Thanks to Sanja Fidler and Raquel Urtasun for providing me with my first research experience in machine learning, and consequently my gateway into a great research community; thanks to Orion Buske, Mike Brudno, Justin Boutilier and Timothy Chan for supervising my initial non-ML undergrad research experiences and making them positive ones; thanks to Sheila McIlraith and Leanne Dawkins for your kindness and invaluable assistance in securing my first undergraduate NSERC grant in 2014; and thanks to Jen Campbell for encouraging me to pursue the computer science major in 2012, and opening some administrative doors to make that possible for me.

Finally, the biggest thank you of all to my parents, without which none of this would have been possible (literally and figuratively). Thanks always for loving me, supporting me, and giving me the most solid foundation I could ever ask for.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

> Professor Horst Rittel of the University of California Architecture Department has suggested in a recent seminar that the term "wicked problem" refer to that class of social system problems which are ill-formulated, where the information is confusing, where there are many clients and decision makers with conflicting values, and where the ramifications in the whole system are thoroughly confusing. The adjective "wicked" is supposed to describe the mischievous and even evil quality of these problems, where proposed "solutions" often turn out to be worse than the symptoms.
>
> C. West Churchman, 1967 (Churchman, 1967)

1.1 High-Stakes Machine Learning

How do you use an automated data-driven system to make high-stakes, potentially life-or-death, decisions? This is not the type of question that has traditionally fallen within the purview of machine learning. More standard questions might have been "how long does it take for this optimizer to converge?" or "how do we find the separating hyperplane with the largest margin?" or "what are good heuristics for minimizing the energy function over this class of graphical models?" There is a good reason for this: the latter questions are answerable with the traditional tools of machine learning (formal reasoning and empirical exploration), whereas the first question may not be satisfactorily answerable at all.

While the question is *historically* unusual in the context of the machine learning literature, it is entirely reasonable in a modern context, now that algorithmic and computational advancements have made machine learning (ML) more broadly used than ever. ML has become more performant, with deep learning improving rapidly in domains like vision (He et al., 2016, Redmon et al., 2016) and language comprehension (Brown et al., 2020, Devlin et al., 2019). Methods like random forests (Breiman, 2001) and gradient boosting (Chen et al., 2015) have achieved huge impact in industry due to their ease of deployment and predictive utility on tabular data. Additionally, ML has rapidly become more scalable, with the advent of GPU- and TPU-based deep learning (Jouppi et al., 2017, Krizhevsky et al., 2012), as well as continuously improving general computation techniques (Schaller, 1997) that make large amounts of data easier to manipulate. On top of this, applying ML has never been easier, with the dissemination of popular open-source libraries (Abadi et al., 2016a, Baydin

et al., 2018, Harris et al., 2020, Paszke et al., 2019, Pedregosa et al., 2011) and free online expertise (Bates, 2019), and a new focus on the importance of data collection (Goodman et al., 2013, Roh et al., 2019, Rutledge et al., 2019). This more effective, scalable, and accessible version of ML is projected to have enormous economic impact (Brynjolfsson and Mitchell, 2017), and has resulted in application to widely disparate domains, ranging from scientific research (Mjolsness and DeCoste, 2001) to business enterprise (Vafeiadis et al., 2015) to virtual assistants (Campagna et al., 2017) to search and recommendations (Haldar et al., 2019) to weather forecasting (Haupt et al., 2018).

With ML's growing popularity has come more frequent application in high-stakes decision-making domains, such as healthcare (Bibault et al., 2016, Nemati et al., 2018, Rajkomar et al., 2019, Tafti et al., 2017), criminal justice (Berk, 2012, Campbell, 2020, Hannah-Moffat, 2013), and personal finance (Baer and Kamalnath, 2017, Brotcke, 2022). These situations differ from other applications of ML in that each individual decision can have a high impact on an individual person's life or livelihood. For instance, even a single prediction has significant impact when it comes to denying bail or parole, or mortgage eligibility, or determining a course of medical treatment. It is therefore reasonable for us to have different desiderata for these systems than for ML applications in general — rather than just asking that we e.g. maximize accuracy or profit across a large pool of examples, we imbue individual predictions with particular meaning in these scenarios, and desire them to have certain properties. Outlining some of these properties, and discussing methods to move towards their inclusion in ML systems, will be the focus of this thesis.

1.2 Trust and Contracts in Machine Learning

If a system is intended to make important, life-altering decisions, we should have *trust* in it. Jacovi et al. (2021) explore what it means for an automated agent or system to be "trustworthy", saying that "the trustor must be vulnerable to the agent's actions, and the trustor's goal in developing trust is to anticipate the impact of the AI model's decisions". They formalize this as follows:

> If H (human) perceives that M (AI model) is trustworthy to contract C, and accepts vulnerability to M's actions, then H trusts M contractually to C. The objective of H in trusting M is to anticipate that M will maintain C in the presence of uncertainty, and consequently, trust does not exist if H does not perceive risk.

We note two aspects of this definition. First, the importance of risk — it only makes sense to discuss a model's trustworthiness to the degree which the model's output has the potential to harm the human. This is why trust is a useful concept for analyzing high-stakes applications in particular. Second, the notion of a contract — ML models tend to provide (implicitly or explicitly) some sort of guarantee upon deployment, regarding a certain level of performance along a metric. The trust that a human holds towards a model is then essentially trust in its ability to fulfill that contract[1].

This type of trust can be secured or improved in many ways, allowing us to effectively use the notion of trust as an umbrella term for a number of properties we might care about in high-stakes decision-making systems. For instance, we can build tools which allow for the model's decisions to

[1] We use the term "contract" here somewhat informally; however, Hadfield-Menell and Hadfield (2019) connect these ideas to the economics literature on incomplete contracting, noting that, as with humans, AI contracts will "unintentionally and unavoidably" be incomplete with respect to the full specification of behaviour we desire, advocating for the building of external structure to help fill these inevitable gaps.

be interpreted by a user (helping the user to anticipate a model's decision process) (Rudin, 2019); we could improve the privacy of a learning process (so a user can feel confident any that information which is provided to a model not be leaked) (Chaudhuri et al., 2011); or we could improve the security of a model so it cannot be easily hacked by external actors (Barreno et al., 2010, Papernot et al., 2016).

One way of advancing trust along these lines can be thought of as expanding the class of contracts which ML systems can uphold. Commonly in ML, contracts are defined in terms of an average example-wise accuracy metric (for instance, classification error or a proper scoring rule (Gneiting and Raftery, 2007)). Each example receives some type of score (or loss) indicating how "close" the prediction on that example is to ground truth, and these scores are averaged over an evaluation set to produce a final score (or provide some other form of performance guarantee e.g. worst-case error). Then the power of the contract stems from: 1) that the scoring function we use accurately represents the properties we desire from a performant model, and 2) that the evaluation set is drawn from the distribution of inputs that we want to perform well on at deployment time. Given these two assumptions, the contract provides a probabilistic guarantee that our deployed model's predictions will (likely) not be too much further from "ground truth" than its performance in evaluation[2]. However, in real world, high-stakes scenarios, these standard contracts may not be sufficient, as we may be looking for different types of properties from our models beyond average closeness to a given ground truth label. For instance, we may want to know that our model will not act discriminatorily towards various socially-defined groups in our data. We may want to ensure that our model can make appropriate decisions for data points which are not represented in our evaluation distribution — either expressing appropriate uncertainty, or passing off to another, more equipped decision-maker. We may have a discrepancy between the loss function we use to evaluate our model and the impact its output will actually have. In each of these situations, we will require different approaches in order to provide and satisfy a contract which a user can feel more closely matches their desiderata. While humans may exhibit some of these same flaws we hope to remove from ML systems as well (for instance, biased or imprecise decision-making), the scalability of ML exacerbates the need for research in this direction: any harm committed by a model may be committed hundreds or thousands of times over, frictionlessly, with a single deployment command.

1.3 Principles & Paradigms in Trustworthy ML Research

We now make this more concrete, enumerating some of the approaches taken in the trustworthiness literature. We first look at two principles from within the literature as key for building trustworthy ML, then discuss two different paradigms that researchers operate within.

1.3.1 Trustworthiness Principles

The notion of trust is broad, and can be understood as an umbrella term for a number of different principles that we might desire of of a high-stakes ML model. Here, we give a high level overview of

[2]For instance, a Hoeffding bound (Krafft and Schmitz, 1969) says that our model's accuracy on a held-out test set is no more than $t \in \mathbb{R}$ worse than the true accuracy on the distribution it was sampled from, with probability $1 - \exp(-t^2)$. A range of more complex guarantees can be given using information about the model structure (Vapnik et al., 1994) or by using a prior over parameters (McAllester, 1999) (possibly learned (Parrado-Hernández et al., 2012)).

proposed approaches for the two most relevant principles for this thesis. Note that each principle is fundamentally imprecise, with definitions that may be highly contested or context specific.

Fairness. Machine learning systems should not act in discriminatory ways. The principle of fairness is a classic example of an "essentially contested concept", a concept around which "disputes [are] not resolvable by argument of any kind ... [despite being] sustained by perfectly respectable arguments and evidence" (Gallie, 1955) — that is, two people may have different and contradictory definitions, despite both having valid reasoning, and even coming to core agreement about what the concept should represent. See Chapter 2 for further discussions of specific fairness definitions; here, we discuss several frameworks from the literature for thinking about fairness. These frameworks are predominantly applied to classification, but can and do extend to ideas about representation learning, regression, ranking, etc (Komiyama et al., 2018, Zehlike et al., 2021, Zemel et al., 2013).

Perhaps the dominant framework for thinking about fairness in the machine learning literature is *group fairness*. In group fairness, there is some grouping of our data that we care about (for instance, a socially-salient attribute like gender), and we want our model to treat each group similarly. The word "similarly" can take a number of meanings, but the basic structure is that some aggregate statistic is calculated over each group, and we want the statistics from each group to be as close as possible. For instance, demographic parity (Dwork et al., 2012) requires that the average predictions in each group be equal. Under this framework, each example receives a group association through a categorical attribute, and we evaluate our success using comparisons of group-wise aggregate statistics.

This differs significantly from an alternative framework, *individual fairness*, which states that "similar individuals should be treated similarly" (Dwork et al., 2012). There is no notion of group membership here; rather, there is some underlying similarity metric by which we can reliably compare individuals. Frequently, approaches to improving individual fairness assume the existence of some oracle to query this metric from, or auxiliary data regarding item similarity (Ilvento, 2020, Jung et al., 2021). It can be related to group fairness by the idea that in the "true" underlying similarity metric, axes along which unfair discrimination might occur "should" be ignored.

A third class of approaches, which can be thought of as existing between group and individual fairness, is *counterfactual fairness* (Kusner et al., 2017), which assumes the existence of a causal model describing the data generating process. Assuming the correctness of the causal model, we are allowed to ask *counterfactual queries*: what would an individual's data point be had something about them been altered? For instance, if gender is a variable in the model, we can ask what a person's data would be if they had been of another gender, given the data we observe about them already. Counterfactual fairness requires that, for every individual, the model output is the same for that individual's original data and for all the counterfactual inputs we can make for that individual by changing some group attribute. Since it considers group attributes, counterfactual fairness is like group fairness; but since it considers individual outcomes, it is like individual fairness. Critiques of this framework argue that these counterfactuals can never be well-defined, since interventions on these group attributes are unrealistic, and therefore impossible to specify correctly in a causal model (Hu and Kohler-Hausmann, 2020).

Robustness. The performance of machine learning should not be vulnerable to small or unimportant perturbations. We can think of various notions of robustness as corresponding to various classes of perturbations. This differs from the classical notion of robustness (Huber, 2011), which concerns the existence of outliers in training data, and how they can make statistical methods unreliable — for instance, using the median or trimmed mean rather than the mean improves this notion of robustness. For various reasons, this is no longer the predominant notion of robustness in the machine learning literature. [3] Here, we discuss some of the currently prominent ones.

It was first noted in Szegedy et al. (2014) that neural networks are vulnerable to adversarial perturbations: small, imperceptible alterations of input examples which nonetheless can drastically change a model's predictive output. In response, a large literature has arisen with the express purpose of improving *adversarial robustness*, that is, making deep learning robust to these perturbations, often defined in terms of those which have small norm. Some of these approaches involve theoretical guarantees of adversarial robustness (Cohen et al., 2019, Li et al., 2019, Raghunathan et al., 2018). Other approaches are more heuristic — the most common technique is to use adversarial examples at training time to improve robustness (Goodfellow et al., 2014b, Huang et al., 2015, Madry et al., 2018, Shaham et al., 2018), while another theme is leveraging the diversity of ensembles (Kariyappa and Qureshi, 2019, Pang et al., 2019, Tramèr et al., 2018, Yang et al., 2020). Evaluation of any proposed solution to this problem is very challenging, since there are often new methods by which adversarial examples can be obtained (Athalye et al., 2018, Carlini et al., 2019).

While adversarial robustness is a very challenging problem, it is a limited notion of robustness — in reality, we want to be able to ignore perturbations not just based on small norm, but rather other irrelevant factors: for instance, lighting conditions in object recognition, or writing style in sentiment classification. The field of *domain generalization* has this type of robustness in mind. Following the formulation of Peters et al. (2016), we define some *environments* over our data, which correspond to these different conditions — causally, we can think of them as representing different interventions on a causal graph. Then, the aim is to generalize to environments which are either unobserved or underrepresented in our training data. Approaches to this problem often emphasize learning appropriate invariances (Arjovsky et al., 2019, Muandet et al., 2013, Peters et al., 2016), improving worst-case or minority group performance (Krueger et al., 2021, Sagawa et al., 2019), or meta-learning (Li et al., 2018, Qiao et al., 2020). Again, it has been noted that evaluation of methods on this task is quite challenging, and that baseline supervised learning (i.e. expected risk minimization) is difficult to beat (Gulrajani and Lopez-Paz, 2021).

For an even more general framework, we can turn to the notion of *distributional robustness*, which asks that our model performs well on *all* input distributions similar to the training distribution. This is a flexible definition: it allows us to specify a whole family of nearby distributions, often according to some distance or divergence that we choose. It can encompass the domain generalization problem, where we set our family of distributions to be the set of environment-conditional distributions. It is difficult to isolate the origination of this framework, but it has a long history in optimization, going back at least as far as Žáčková (1966), and more recently revived within the operations research literature (Ben-Tal et al., 2013, Bertsimas et al., 2018, Goh and Sim, 2010). Within machine learning and statistics, distributionally robust approaches have been developed for any distributional family specified by an f-divergence ball around the training data (Duchi et al., 2016), and popular

[3] Although interesting extensions exist in research directions such as data poisoning (Steinhardt et al., 2017).

approaches leveraging gradient penalties (Gao et al., 2017), variance regularization (Namkoong and Duchi, 2017), and Lipschitz regularization (Cranko et al., 2021) have been incorporated into its framework. The flexibility of the approach is promising, with creative extensions including causal structure (Meinshausen, 2018) and human annotations (Srivastava et al., 2020).

1.3.2 Contract-Based vs. Design-Based Research

We identify two general paradigms of research within trustworthy ML, which we term *contract-based* research and *design-based* research[4]. We present this taxonomy in order to highlight the ways in which the work in this thesis, which falls under the design-based paradigm, contributes to trust without necessarily providing formally-specified "contracts" (i.e. guarantees) of our algorithmic contributions. Rather, we present useful techniques for designing ML-based systems, which move us towards more trustworthy outcomes.

In the contract-based paradigm, researchers present formal guarantees for a particular method of determining outcomes. Such guarantees are essential for the ultimate building of trustworthy systems, since formal guarantees are the bedrock of contracts. Therefore, it is no surprise that we find contract-based approaches across the trustworthiness space. There is work on providing new guarantees for the optimality and generalization capabilities of fair classifiers (Agarwal et al., 2018, Canetti et al., 2019, Celis et al., 2019, Hardt et al., 2016b), including extensions to multi-group fairness (Hébert-Johnson et al., 2018, Kearns et al., 2018, Kim et al., 2019) and fair representation learners (McNamara et al., 2017, Oneto et al., 2020). There is also a genre of works which formally guarantee the *limits* of fairness approaches, providing negative results around the incompatibility of different criteria (Chouldechova, 2017, Kleinberg et al., 2017, Lechner et al., 2021), as well as documenting the optimal tradeoffs between different desirable properties (Menon and Williamson, 2018, Zhao and Gordon, 2022). Among the many definitions of "robustness", contract-based work has been done on the statistical properties of robustness to outliers (Huber, 2011), guaranteeing adversarial robustness in deep learning models (Cohen et al., 2019, Li et al., 2019), and training distributionally robust models (Duchi and Namkoong, 2019, Namkoong and Duchi, 2016), all resulting in various forms of guarantees. Outside of these areas, differential privacy (Dwork, 2008) stands out as a theoretical approach to privacy which has gained massive popularity, resulting in contract-based approaches for logistic regressions and SVMs (Chaudhuri et al., 2011), ensembles (Papernot et al., 2018), as well as SGD-based approaches (Bassily et al., 2014), including deep learning specifically (Abadi et al., 2016b) which provide guarantees through their adherence to differentially private principles.

The other paradigm of research in this area is driven by a synthesis of empirical results and theoretical analysis, which jointly provide evidence for the advantages of new methods and techniques. We term this the *design-based* paradigm, since the main focus is on the design of new algorithms, architectures, systems, or principles which allow for new, interesting solutions of challenging problems which may not even have consensus formalized definitions. The name we give for this research paradigm is inspired by a blog post by David Chapman[5], which argues that AI research has elements of design practice:

[4]These paradigms exist within general ML research as well, but their application to trustworthy ML is the relevant aspect for our current purposes.

[5]A fascinating read on the nature of research in AI: `https://metarationality.com/artificial-intelligence-progress`

> AI typically applies ill-characterized methods to nebulous problems with nebulous solution criteria (Using neural networks to translate Mandarin Chinese to English, for example.) ... If you can nail down the problem, eliminate nebulosity, and demonstrate correctness, you are doing mainstream computer science, not AI[6]. Which is great! But not always possible. No one can say what the problem of translation is, and there is no such thing as an optimal translation. But, your aim as an AI researcher is to do it well enough to impress people. ... You build a series of quick-and-dirty prototypes ... the different patterns of good and bad translations the programs produce suggests each next implementation ... And as you proceed, your understanding of what translation even means changes. This is your "reflective conversation with the concrete materials" — which include both natural language texts and program structure.

This approach to research is particularly relevant to questions within trustworthy ML, since often these problems and their potential solutions are particularly challenging to formally define. In this paradigm, one can contribute to progress in trustworthy ML by designing new systems and approaches, and demonstrating evidence that they uphold some aspect of trust, either using experimental data or theoretical analysis. Crucially, this analysis need not rise to the level of "proofs of correctness" or "performance guarantees"; rather, combined with empirical evidence and conceptual synthesis, the work as a whole outlines the strengths and weaknesses of the approach is explores. We find examples of this paradigm all across trustworthy ML — in fact, it is arguably the dominant one, possibly due to the difficulty of obtaining theoretical guarantees in deep learning systems. Some examples of this type of work in fairness include demonstrations of how fairness principles can be incorporated into generative models (Edwards and Storkey, 2016, Louizos et al., 2016, Song et al., 2019), schemes for regularizing classifiers to decrease their disparate impact on different subgroups (Bechavod and Ligett, 2017, Kamishima et al., 2012, Zafar et al., 2017b), how the fairness of dynamic systems containing predictors evolves over time (Creager et al., 2020b, Hashimoto et al., 2018, Hu and Chen, 2018), how meta-learning can be adjusted according to some fairness objective (Slack et al., 2020), and how causal models can be incorporated into questions of discrimination (Kusner et al., 2017, Nabi and Shpitser, 2018). In robustness, we find popular, heuristically-based approaches for domain generalization which leverage intuition around invariance (Arjovsky et al., 2019) and distributional robustness (Sagawa et al., 2019) and defenses for adversarial robustness (Goodfellow et al., 2014b, Tramèr et al., 2018) which do not provide guarantees but nonetheless have been hugely influential. This indicates the value of work which advances our understanding of these principles and studies the empirical performance of promising approaches, without necessarily providing guarantees of correctness.

1.4 Trustworthy Machine Learning is a Wicked Problem

We argue that the flexible, multi-faceted research approach taken within the design-based paradigm is a particularly good fit for questions in trustworthy machine learning. Rittel and Webber (1973) introduce the term "wicked problem" to describe a set of public policy problems, which are poorly-

[6] At the time of writing this thesis (June 2022), AI research *is* mainstream computer science; but the blog post was written at least 5 years prior — mainstream CS here probably is meant to include areas like theoretical CS, networks, or programming languages.

defined and often involve directly conflicting values between various stakeholders. Examples they give include developing taxation schemes, urban planning, and designing school curricula[7]. They describe this class of problems as follows (Rittel and Webber, 1973):

> As distinguished from problems in the natural sciences, which are definable and separable and may have solutions that are findable, the problems of governmental planning — and especially those of social or policy planning — are ill-defined; and they rely upon elusive political judgment for resolution ...
>
> The problems that scientists and engineers have usually focused upon are mostly "tame" or "benign" ones. As an example, consider a problem of mathematics, such as solving an equation; or the task of an organic chemist in analyzing the structure of some unknown compound; or that of the chess player attempting to accomplish checkmate in five moves ... It is clear, in turn, whether or not the problems have been solved.
>
> Wicked problems, in contrast, have neither of these clarifying traits ... not because these properties are themselves ethically deplorable. We use the term "wicked" in a meaning akin to that of "malignant" (in contrast to "benign") ... or "tricky" (like a leprechaun)[.]

We argue here that automating high-stakes decision-making in a trustworthy manner can be understood as a wicked problem, and therefore, qualitatively different from other types of questions we ask in ML. Accordingly, we should expect qualitatively different types of solutions. To make this argument, we follow the simplified framework given by Conklin (2006)[8] and use evidence from across the trustworthy ML literature:

1. **The problem is not understood until after the formulation of a solution.** As Conklin (2006) says, "Every solution that is offered exposes new aspects of the problem, requiring further adjustments of the potential solutions". The canonical dialogue around COMPAS provides a clear example. First, Angwin and Larson (2016) suggested one fairness metric which COMPAS violated, then Dieterich et al. (2016) countered with another metric that COMPAS satisfied, and it was ultimately found that these two metrics were inherently incompatible (Chouldechova, 2017, Kleinberg et al., 2017). The tradeoffs that arise in potential solutions are a hallmark of wicked problems, as well as trustworthy machine learning (e.g. the tension between privacy and explainability (Shokri et al., 2021).) As Rittel and Webber (1973) says, "the formulation of a wicked problem *is* the problem!"

2. **Wicked problems have no stopping rule.** Trustworthiness principles or values rarely have agreed-upon definitions e.g. fairness (Narayanan, 2018) and explainability (Burkart and Huber, 2021). Without a definition, we cannot truly verify when we have successfully built a trustworthy model. Practically, these struggles can manifest as difficulties in evaluation, e.g. evaluating model robustness is a challenging problem (Carlini et al., 2019, Gulrajani and Lopez-Paz, 2021) without clear specifications (Madras and Zemel, 2021).

3. **Solutions to wicked problems are not right or wrong.** See #1 and #2: when we do not have a definition, we cannot say something is or is not trustworthy; rather, we can

[7] For a uniquely 2020s-flavoured example, consider setting guidelines around business closures in a pandemic.

[8] The original framework from Rittel and Webber (1973) has ten items but is more specifically targeted at public policy.

debate whether it is good enough or not. To this end, notable definitions of fairness (Hardt et al., 2016b) and privacy (Dwork, 2008) attempt to capture this property with a parameter modulating how far we can deviate from the ideal state.

4. **Every wicked problem is essentially novel and unique.** The trustworthy application of machine learning in medicine and criminal justice (for example) will look extremely different. While there are connections between the two, there is no one-size-fits-all solution — the relevant stakeholders are different and hold different values.

5. **Every solution to a wicked problem is a "one shot operation".** When working in high-stakes situations, we cannot implement and re-implement systems without penalty: every decision is high-impact on someone's life, and the potential for experimentation is limited or non-existent. This is the motivation behind approaches such as safe policy improvement (Ghavamzadeh et al., 2016), which has the goal of improving models without making any large errors.

6. **Wicked problems have no given alternative solutions.** This means that the shape of the solution space is not clear. When considering implementing and deploying trustworthy machine learning, every piece of the system matters: the training algorithm and model, which data and features we choose to input, the structure of human oversight, the provided explanations for decisions, the ability to obtain recourse and audits, etc. Indeed, even the choice to implement a machine learning solution is part of the problem space (Selbst et al., 2019).

In this thesis, we approach trustworthy machine learning as a wicked problem. Given this, we should not expect to solve it completely. Rather, we should hope to move towards a set of values that we care about. We can work with useful definitions, but should not be able to expect to perfectly define these values, nor should we expect to know when we have achieved them. We can propose solutions, but should anticipate that all solutions will be incomplete by nature. We can work within existing theoretical frameworks and run experiments on existing benchmarks, but the central thrust of our results should not rely fully on the validity of these frameworks or benchmarks; rather we should also try to expand the solution space and find creative new formulations or approaches. With this in mind, we turn to the technical content of this thesis.

1.5 Outline

In this thesis, we discuss three frames for how we can improve trust in a deployed ML model in cases where the standard ML contract does not quite hold. Along with the introduction of these issues, we continue by elaborating on different mitigation strategies of how we could improve an ML system from a trustworthiness perspective when these trust issues become a problem. Each of these chapters covers a potential pitfall to trustworthiness, and a potential mitigation approach to improving the ML system which a model is embedded within.

Each section of the thesis contains content corresponding to a paper I published during the course of my studies at the University of Toronto. For each paper, I was a central driver of the work, from conceptualization, to theoretical formulation, to writing code and empirical explorations, to paper writing. For the paper in Chapter 2 on LAFTR, I shared these central responsibilities with Elliot

Creager fairly evenly, where took on slightly more of the problem formulation and initial experiments, and he took on slightly more of the work of building out the experimental codebase. For the papers in Chapter 3 and 4, I was the central hands-on contributor. We will cover the following work:

1. **Learning Fair Representations:** What if we have a number of downstream users that we want to guarantee fairness for? We explore how representation learning can be used as a pre-processing step to remove sensitive information and encourage fair transfer learning. The content of this chapter is from the work "Learning Adversarially Fair and Transferable Representations", co-authored with Elliot Creager, Toniann Pitassi, and Richard Zemel (Madras et al., 2018a).

2. **Detecting Underspecification:** What happens if our training data is insufficient to specify a single "good" output on our training set? We explore how we can use the fingerprint of *underspecification* in an already-trained model to tell the user where a model might require more information in fully post-hoc way. The content of this chapter is from the work "Detecting Underspecification with Local Ensembles"[9], co-authored with James Atwood and Alexander D'Amour (Madras et al., 2019a).

3. **Learning to Defer to an External Decision-Maker:** How do human and machine decision-makers collaborate? How can we build a system when we know that they will interact - for instance, if the model output will actually be taken as an input to some human-driven decision-process? We explore how to model this scenario, and how we might take it into account in a learning process to improve both the accuracy and fairness of a joint human-machine decision-making system. The content of this chapter is from the work "Predict Responsibly: Improving Fairness and Accuracy by Learning to Defer", co-authored with Toniann Pitassi and Richard Zemel (Madras et al., 2018b).

[9]This work was originally published at ICLR 2020 under the title "Detecting Extrapolation with Local Ensembles"; the title was later changed to better reflect our evolved understanding of the paper's contribution.

Chapter 2

Learning Fair and Transferable Representations with an Adversary

2.1 Fairness

As machine learning tools are used more broadly in high-stakes scenarios, *fairness* — the ideal that these models should not behave in discriminatory ways towards various subpopulations — has arisen as a natural desideratum. General interest in this issue has increased dramatically in the past several years (Fig. 2.1), as a number of examples have gained popular traction by demonstrating the potentially discriminatory harms of machine learning in salient ways:

- Search results suggesting that a person may have a criminal record were more likely to arise when a typically black name was used than a white one, even for people who did not actually have criminal records (Sweeney, 2013)
- Translations systems reproduced gender stereotypes when making inferences about occupations (for instance, assuming that engineers are male) (Prates et al., 2020)
- Risk scores in the criminal justice system, intended to help judges make decisions around pre-trial detention, had vastly differing false positive and negative rates between different racial groups (Angwin et al., 2017)

In each of these examples, the potential (or already concrete) harms are evident, the stakes are clear, and the scalability of machine learning models means that any impact they may have will be multiplied many times over. Therefore, the question of what it means for machine learning models (or algorithms more generally) to discriminate, how that discrimination is learned or instantiated, and how best to mitigate it, has arisen as a serious one, and a key challenge in building trustworthy machine learning systems.

However, the formalization of this notion has proven elusive. A popular commentary on the topic from 2018 suggested that there are at least "21 definitions of fairness" (Narayanan, 2018) — that is, at least 21 different ways to mathematically formalize a condition under which we could declare that a machine learning model is behaving fairly[1]. Each of these definitions has their own implications for

[1] Today, this number is much higher than 21.

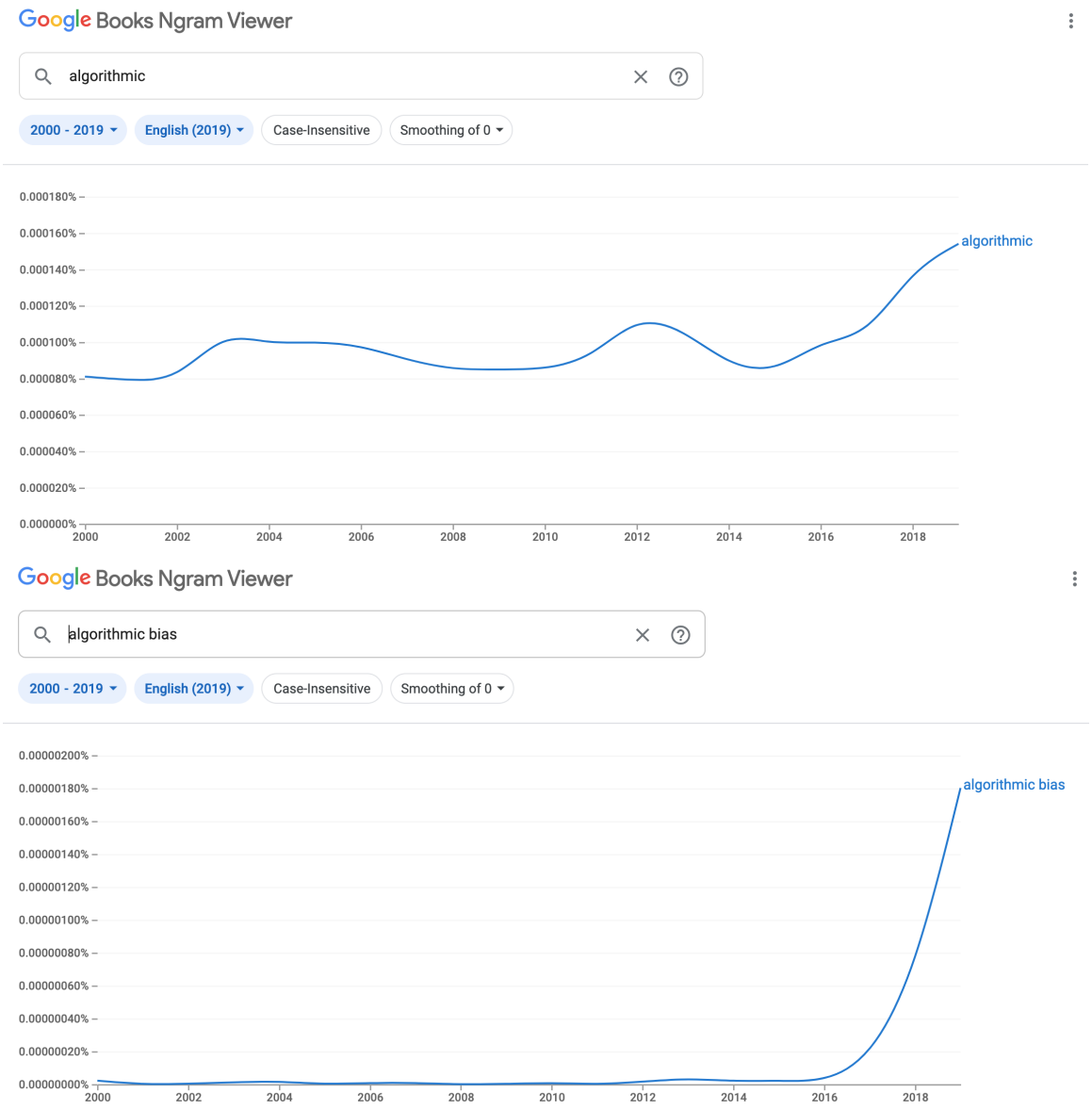


Figure 2.1: The recent, drastic rise in discussion of "algorithmic bias" (right), as contrasted with a more gentle increase in the discussion of anything "algorithmic" (left), suggests increased salience of fairness-related topics in the past several years, over and above increased discussion of algorithmic tools.

what properties are considered important: for instance, do we care about different groups in the data or how individuals are treated? Do we care about the causal generative process? Do we incorporate a notion of merit? Are we concerned with giving different groups similar outputs, or giving them similar rewards, or just with modelling them equally well within our learned system? These definitions each represent a different normative view of what fairness is, and each mean different things for our learned model — indeed, many of them directly contradict each other and cannot be satisfied within one model. To this end, much practical application of fairness methodologies is deeply concerned with identifying the needs of various stakeholders, and understanding the best tradeoffs between them (Bakalar et al., 2021).

2.1.1 Fairness Metrics and Fair Classification

Despite this inherent practical complexity, in this thesis we will discuss only a small number of fairness metrics in particular, and examine how our fairness-related mitigations perform along these axes. As a running example, we will consider the task of predicting if an applicant for a loan would default on a loan, if given one. This is a binary task with a target label $Y \in \{0, 1\}$ — 1 represents the "bad" outcome (a default on the loan), and and 0 represents the "good" outcome. This prediction will be used to guide policy, as we hope to inform our loan decisions using this information: we want to give loans to as many people as possible who will not default ($Y = 0$), and as few as possible to people who will default ($Y = 1$). We define our binary prediction to be $\hat{Y} \in \{0, 1\}$: this is the output of our learned model, which, in the standard ML setup, we want to train to be as predictive as possible of the ground truth Y.

We note quickly that this can be viewed as a causal inference problem, since we are interested in an *intervention*: what policy should we use to determine who receives a loan? However, Kleinberg et al. (2015) describe these as "prediction policy problems", where we can reduce the intervention question to a prediction problem. They state that this is the case for tasks which satisfy two criteria: first, the benefit of each action (loan or no loan), is modulated by the uncertain outcome we hope to predict (loanworthiness Y); and second, that the predictive outcome is not causally dependent on the chosen action. According to Kleinberg et al. (2015), many policy problems fit this framework and as such can be solved as prediction problems — our loan example is one such problem[2].

We also define a *sensitive attribute* $A \in \{0, 1\}$, which partitions our data into two groups (not necessarily of equal size). These groups represent some attribute along which we are worried about discrimination (e.g. race, gender, age, socio-economic status). While many or all of the attributes we could be concerned with are not binary, we use the binary formulation here as a simplification, and note that often attributes which are not binary do get binarized as a matter of practice in policy (for instance, over 18 years of age vs under 18 are frequently treated as distinct categories, even though there exists signifiant variance along the age spectrum within each category).

We define here three fairness metrics that we consider throughout this chapter. These metrics will be phrased as constraints: the metric form can be derived by simply seeing how far we deviate from perfect satisfaction from the constraint. The first is *demographic parity* (Dwork et al., 2012),

[2]While this task can be formulated and solved as a prediction problem, this does not mean that standard expected risk minimization usually yields an optimal solution if training a model on data collected under a historical policy. This is because the examples in this dataset will all be those for which we previously gave a loan to — hence it will be a biased sample of the population we care about. See Kallus and Zhou (2018) for an exploration of this "residual unfairness" of learning from data collected by a prior policy.

which stipulates that both groups have the same rate of positive prediction, formally

$$P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0) \tag{2.1}$$

or equivalently, $\hat{Y} \perp A$. This metric is helpful in cases where we want to treat our two groups identically, without regards for what the label distribution is — note that if we observe demographic parity, but our base rates are not equal (if $P(Y = 1|A = 1) \neq P(Y = 1|A = 0)$), then we can never achieve perfect accuracy. This suggests that if our fairness constraint is demographic parity, we might have a tradeoff between fairness and accuracy on this dataset. Tradeoffs are a theme in fairness work, as competing objectives are often incompatible.

At first glance it isn't clear why we would ever want to enforce a constraint which is almost guaranteed to decrease our accuracy. However, we might want to do this if we believe that the labels themselves are unfair or were collected unfairly; that is, if one group has a higher base rate of positive labels in the data, but we think this is only due to historical bias. For instance, this might occur if our previous loan policy was better at identifying repayment ability in white people than black people. That would result in an effective base rate different in the data which is not representative of the real ground truth distribution. Alternatively, this could also arise if our loan data was collected in a previous year, but now some social program was put into place which increased the rates of loan repayment among certain marginalized, predominantly black communities. In both cases, we may want to enforce a constraint such as demographic parity, even if this removes the possibility of perfect prediction.

The second metric we consider is *equalized odds* (Hardt et al., 2016b). In this, we stipulate that we match false positive and true positive rates across groups:

$$P(\hat{Y} = 1|A = 1, Y = 1) = P(\hat{Y} = 1|A = 0, Y = 1) \tag{2.2}$$

$$P(\hat{Y} = 1|A = 1, Y = 0) = P(\hat{Y} = 1|A = 0, Y = 0) \tag{2.3}$$

This is equivalently stated as requiring conditional independence, i.e. $\hat{Y} \perp A|Y$. It is a fundamentally different constraint than demographic parity because it includes the labels Y. Conceptually, this means that, when we require equalized odds, we want to have equal impact on the two groups *only once we have accounted for the difference in labels.* Another way to think about it is that we want our errors to be evenly distributed — this can be a good goal in cases where errors in the positive direction and the negative direction have vastly different impacts (such as in loan decisions: a positive error means wrongly denying a loan, a negative error means wrongly allocating a loan). We would only ask for this to hold if we believe that the labels in our dataset are a sufficiently good indicator of *merit* — that is, if we think that matching the labels exactly is a worthwhile objective.

It was noted that, while the equalized odds constraint does necessarily conflict with accuracy, it does present a practical tradeoff with *calibration*, another desirable property stipulating that our output can be interpreted the same way across both groups (Chouldechova, 2017, Kleinberg et al., 2017, Pleiss et al., 2017). To this end, Hardt et al. (2016b) propose a relaxation of equalized odds called *equal opportunity*, the third metric we consider, which focuses on only requiring equality in the "advantaged outcome group" i.e. those that will ultimately pay back their loan, which is just Eq. 2.3 in our notation.

How could we navigate trading off these classification objectives against standard accuracy

desiderata? A number of methods have been developed in the field of *fair classification* for this purpose. The key observation in these works was emphasized in the seminal work of Dwork et al. (2012): simply removing a sensitive attribute from the feature input space is not sufficient to achieve fairness, since a model can easily infer that attribute from other features. For instance, if postal code is a feature in our data, that can easily be used to infer a person's race with non-trivial accuracy, since different postal codes tend to have distinct racial makeups (Walks and Bourne, 2006). Therefore, the key is "fairness through awareness" (Dwork et al., 2012): we must use the sensitive attribute in classification in order to remove its impact.

Fair classification methods take a wide range of approaches. These include post-processing (Hardt et al., 2016b, Woodworth et al., 2017) and constrained optimization (Agarwal et al., 2018, Zafar et al., 2017b). These approaches are quite effective in solving the fair classification problem as defined — they are often able to navigate tradeoffs between the selected fairness metric and accuracy reasonably efficiently, according to some tuning hyperparameter.

2.2 Learning Fair Representations

We now turn to the core task of this chapter, which is learning fair representations. As stated previously, a number of methods have been proposed for fair classification: to understand the motivation for fair representation learning, we can break down the classification setup. There are two implicit steps involved in every prediction task: acquiring data in a suitable form, and specifying an algorithm that learns to predict well given the data. In practice these two responsibilities are often assumed by distinct parties. For example, in online advertising the so-called *prediction vendor* profits by selling its predictions (e.g., person A is likely to be interested in product B) to an advertiser, while the *data owner*[3] profits by selling a predictively useful dataset to the prediction vendor Dwork et al. (2012).

Because the prediction vendor seeks to maximize predictive accuracy, it may (intentionally or otherwise) bias the predictions to unfairly favor certain groups or individuals. As discussed in the prior section, the use of machine learning in this context is especially concerning because of its reliance on historical datasets that include patterns of previous discrimination and societal bias.

Meanwhile, the data owner also faces a decision that critically affects the predictions: what is the correct *representation* of the data? Representation learning is a challenging problem in its own right. With the increasing effectiveness of deep learning systems, pre-trained representations have gained an increased role in the ML practitioner's handbook: often, the first step of a prediction problem will be to take some existing pre-trained model, and use these representations as a useful set of inputs to a downstream classifier (Bommasani et al., 2021, Brown et al., 2020, He et al., 2016). The advantages of learning a good representation are two-fold: first, a good representation can surface broadly useful features, which can be used on a number of potential downstream tasks (Bengio et al., 2013). Second, representation learning allows for efficient reuse of computation: learning a good representation can be computationally intensive. However, it only needs to be done once, and then the learned representation can be used by a number of downstream agents, many of whom may not

[3]This terminology was originated in Dwork et al. (2012) — we can consider the "data owner" role as encompassing the "compute owner" role assumed by the creators of large, pre-trained ("foundation") models today as well, where one party owns a large amount of relevant pre-training data and the resources to train a useful, general representation function, which is then given/licenced out to prediction vendors to assist with their data processing needs.

have access to the same amount of computational power.

It is for these reasons we turn to representation learning. Given the multitude of prediction vendors and potential tasks, fair classification methods may not present a feasible solution: relevant vendors may not have the technical know-how to implement such a fair classification scheme, and it's not clear that we should trust them to, given vendor incentives to maximize predictive accuracy. The data owner's role as a bottleneck in this system is promising — it may be possible for the data owner to output a dataset of representations such that downstream vendors are guaranteed to make fair predictions when given that dataset. We know that if we want to maximize the prediction vendor's utility, then the right choice is to simply collect and provide the prediction vendor with as much data as possible. However, such representations have been shown to contain unwanted biases (Bolukbasi et al., 2016, Caliskan et al., 2017, Steed and Caliskan, 2021). Could we instead output representations which assure that prediction vendors learn only *fair* predictors? Such a representation learning method could provide fairness guarantees elegantly and efficiently.

This is the question we turn to in this chapter, where we frame the data owner's choice as a representation learning problem with an adversary criticizing potentially unfair solutions. We connect common group fairness metrics to adversarial learning by providing appropriate adversarial objective functions for each metric that upper bounds the unfairness of arbitrary downstream classifiers in the limit of adversarial training; we distinguish our algorithm from previous approaches to adversarial fairness and discuss its suitability to fair classification due to the novel choice of adversarial objective and emphasis on representation as the focus of adversarial criticism; we validate experimentally that classifiers trained naively (without fairness constraints) from representations learned by our algorithm achieve their respective fairness desiderata; furthermore, we show empirically that these representations achieve *fair transfer* — they admit fair predictors on unseen tasks, even when those predictors are not explicitly specified to be fair.

In Sections 2.3 we discuss relevant background materials and related work. In Section 2.4 we describe our model and motivate our learning algorithm. In Section 2.5 we discuss our novel adversarial objective functions, connecting them to common group fairness metrics and providing theoretical guarantees. In Section 2.6 we discuss experiments demonstrating our method's success in fair classification and fair transfer learning.

2.3 Background

2.3.1 Fairness

We use the notation and setup provided in Sec. 2.1.1. In fair classification we have some data $X \in \mathbb{R}^n$, labels $Y \in \{0, 1\}$, and sensitive attributes $A \in \{0, 1\}$. The predictor outputs a prediction $\hat{Y} \in \{0, 1\}$. We seek to learn to predict outcomes that are accurate with respect to Y but fair with respect to A; that is, the predictions are accurate but not biased in favor of one group or the other. There are many possible criteria for group fairness in this context: the relevant ones for this chapter (demographic parity, equalized odds, and equal opportunity) are covered in Sec. 2.1.1.

Satisfying these constraints is known to conflict with learning well-calibrated classifiers (Chouldechova, 2017, Kleinberg et al., 2017, Pleiss et al., 2017). It is common to instead optimize a relaxed objective (Kamishima et al., 2012), whose hyperparameter values negotiate a tradeoff between

maximizing utility (usually classification accuracy) and fairness.

2.3.2 Adversarial Learning

Adversarial learning is a popular method of training neural network-based models. Goodfellow et al. (2014a) framed learning a deep generative model as a two-player game between a generator G and a discriminator D. Given a dataset X, the generator aims to fool the discriminator by generating convincing synthetic data, i.e., starting from random noise $z \sim p(z)$, $G(z)$ resembles X. Meanwhile, the discriminator aims to distinguish between real and synthetic data by assigning $D(G(z)) = 0$ and $D(X) = 1$. Learning proceeds by the max-min optimization of the joint objective, originally proposed by Goodfellow et al. (2014a) to be

$$V(D, G) \triangleq \mathbb{E}_{p(X)}[\log(D(X))] + \mathbb{E}_{p(z)}[\log(1 - D(G(z)))],$$

where D and G seek to maximize and minimize this quantity, respectively. Since then, there have been a number of advances in adversarial learning, notably a non-saturating objective based on Wasserstein distance (Arjovsky et al., 2017); however, we do not explore these in this work.

2.3.3 Related Work

Definitional works in fair machine learning have been broadly concerned with *group fairness* or *individual fairness*. Dwork et al. (2012) discussed individual fairness within the owner-vendor framework we utilize. Zemel et al. (2013) encouraged elements of both group and individual fairness via a regularized objective, introducing the notion of "learning fair representations" via a clustering objective. An intriguing body of recent work unifies the individual-group dichotomy by exploring fairness at the intersection of multiple group identities, and among small subgroups of individuals Hébert-Johnson et al. (2018), Kearns et al. (2018).

Calmon et al. (2017) and Hajian et al. (2015) explored fair machine learning by pre- and post-processing training datasets. McNamara et al. (2017) provides a framework where the data producer, user, and regulator have separate concerns, and discuss fairness properties of representations, including calculating the "cost of mistrust" in such a pipeline. Louizos et al. (2016) give a method for learning fair representations with deep generative models by using maximum mean discrepancy (Gretton et al., 2007) to eliminate disparities between the two sensitive groups.

Adversarial training for deep generative modeling was popularized by Goodfellow et al. (2014a) and applied to deep semi-supervised learning (Odena, 2016, Salimans et al., 2016) and segmentation (Luc et al., 2016), although similar concepts had previously been proposed for unsupervised and supervised learning (Gutmann and Hyvärinen, 2010, Schmidhuber, 1992). Ganin et al. (2016) proposed adversarial representation learning for domain adaptation, which resembles fair representation learning in the sense that multiple distinct data distributions (e.g., demographic groups) must be expressively modeled by a single representation. Related approaches have been broadly popular. Some approaches use simpler discrepancy measures (which can be thought of as a more limited adversary) to align the statistics of different domains: Sun et al. (2016) in the input data, and Long et al. (2015) and Tzeng et al. (2014) in later layers. Other adversarial-based approaches include Hoffman et al. (2018), which highlight cycle-consistency; Tzeng et al. (2017), which unties the model weights for the two

domains, and Saito et al. (2018) which incorporates task-specific information around the classification boundary.

Edwards and Storkey (2016) made this connection explicit by proposing adversarially learning a classifier that achieves demographic parity. This work is the most closely related to ours, and we discuss some key differences in sections 2.5.4. Similarly, Feldman et al. (2015) also look at a related question around pre-processing data to make two groups look similar. The theorem that we prove in Sec. 2.5.1 has rough analogues (although not exact) in both of these works. Recent work has explored the use of adversarial training to other notions of group fairness. Beutel et al. (2017) explored the particular fairness levels achieved by the algorithm from Edwards and Storkey (2016), and demonstrated that they can vary as a function of the demographic unbalance of the training data. In work concurrent to the work in this chapter, Zhang et al. (2018) use an adversary which attempts to predict the sensitive variable solely based on the classifier output, to learn an equal opportunity fair classifier. Whereas they focus on fairness in classification outcomes, in our work we allow the adversary to work directly with the learned representation, which we show yields fair and transferable representations that in turn admit fair classification outcomes.

A recent related line of work explores a similar concept from the perspective of *invariance*, often with inspiration from the causality literature. Johansson et al. (2016) and Johansson et al. (2020) show how similar discrepancy-minimizing techniques can assist with the estimation of causal effects and counterfactual inference. Ruan et al. (2021) approach the problem from a covariate shift perspective, focusing on the role of shared support between the domains. Wang and Jordan (2021) leans on causal reasoning to provide theoretically motivated representation learning methods. Some works invoke the notion of "invariant causal mechanisms", the idea that some data generating mechanisms stay consistent between domains Mitrovic et al. (2020), Parascandolo et al. (2018). A line of work looks at how disentanglement relates to these questions (Locatello et al., 2020, Träuble et al., 2021). Finally, work inspired by nonlinear ICA (Hyvarinen et al., 2019) provides identifiability guarantees for representation learning in related settings, including with nonlinear models (Khemakhem et al., 2020, Lu et al., 2021). Understanding the practical utility of these approaches is a key question for building reliable systems for fair representation learning with guarantees.

2.4 Adversarially Fair Representations

2.4.1 A Generalized Model

We assume a generalized model (Figure 2.2), which seeks to learn a data representation Z capable of reconstructing the inputs X, classifying the target labels Y, and protecting the sensitive attribute A from an adversary. Either of the first two requirements can be omitted by setting hyperparameters to zero, so the model easily ports to strictly supervised or unsupervised settings as needed. This general formulation was originally proposed by Edwards and Storkey (2016); below we address our specific choices of adversarial objectives and explore their fairness implications, which distinguish our work as more closely aligned to the goals of fair representation learning.

The dataset consists of tuples (X, A, Y) in $\mathbb{R}^n$, $\{0, 1\}$ and $\{0, 1\}$, respectively. The encoder $f : \mathbb{R}^n \to \mathbb{R}^m$ yields the representations Z. The encoder can also optionally receive A as input.

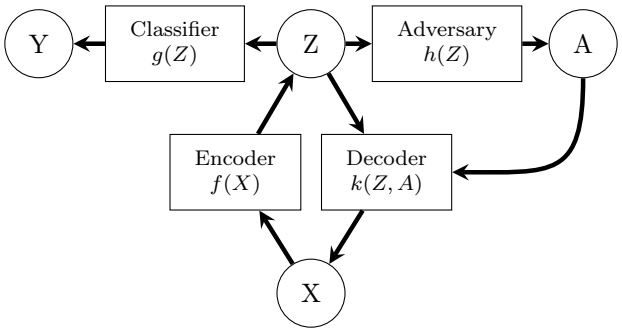


Figure 2.2: Model for learning adversarially fair representations. The variables are data X, latent representations Z, sensitive attributes A, and labels Y. The encoder f maps X (and possibly A - not shown) to Z, the decoder k reconstructs X from (Z, A), the classifier g predicts Y from Z, and the adversary h predicts A from Z (and possibly Y - not shown).

The classifier and adversary[4] $g, h : \mathbb{R}^m \to \{0, 1\}$ each act on Z and attempt to predict Y and A, respectively. Optionally, a decoder $k : \mathbb{R}^m \times \{0, 1\} \to \mathbb{R}^n$ attempts to reconstruct the original data from the representation and the sensitive variable.

The adversary h seeks to maximize its objective $L_{Adv}(h(f(X, A)), A)$. We discuss a novel and theoretically motivated adversarial objective in Sections 2.4.2 and 2.5, whose exact terms are modified according to the fairness desideratum.

Meanwhile, the encoder, decoder, and classifier jointly seek to minimize classification loss and reconstruction error, and also minimize the adversary's objective. Let L_C denote a suitable classification loss (e.g., cross entropy, ℓ_1), and L_{Dec} denote a suitable reconstruction loss (e.g., ℓ_2). Then we train the generalized model according to the following min-max procedure:

$$\underset{f,g,k}{\text{minimize}} \underset{h}{\text{maximize}} \, \mathbb{E}_{X,Y,A} \left[L(f, g, h, k) \right], \tag{2.4}$$

with the combined objective expressed as

$$\begin{aligned} L(f, g, h, k) = \alpha L_C(g(f(X, A)), Y) \\ + \beta L_{Dec}(k(f(X, A), A), X) \\ + \gamma L_{Adv}(h(f(X, A)), A) \end{aligned} \tag{2.5}$$

The hyperparameters α, β, γ respectively specify a desired balance between utility, reconstruction of the inputs, and fairness.

Due to the novel focus on fair transfer learning, we call our model Learned Adversarially Fair and Transferable Representations (LAFTR).

2.4.2 Learning

We realize f, g, h, and k as neural networks and alternate gradient decent and ascent steps to optimize their parameters according to (2.5). First the encoder-classifier-decoder group (f, g, k) takes a gradient step to minimize L while the adversary h is fixed, then h takes a step to maximize L with

[4]In learning equalized odds or equal opportunity representations, the adversary $h : \mathbb{R}^m \times \{0, 1\} \to \{0, 1\}$ also takes the label Y as input.

fixed (f, g, k). Computing gradients necessitates relaxing the binary functions g and h, the details of which are discussed in Section 2.5.3.

One of our key contributions is a suitable adversarial objective, which we express here and discuss further in Section 2.5. For shorthand we denote the adversarial objective $L_{Adv}(h(f(X, A)), A)$—whose functional form depends on the desired fairness criteria—as $L_{Adv}(h)$. For demographic parity, we take the average absolute difference on each sensitive group $\mathcal{D}_0, \mathcal{D}_1$:

$$L_{Adv}^{DP}(h) = 1 - \sum_{i \in \{0,1\}} \frac{1}{|\mathcal{D}_i|} \sum_{(x,a) \in \mathcal{D}_i} |h(f(x, a)) - a| \tag{2.6}$$

For equalized odds, we take the average absolute difference on each sensitive group-label combination $\mathcal{D}_0^0, \mathcal{D}_1^0, \mathcal{D}_0^1, \mathcal{D}_1^1$, where $\mathcal{D}_i^j = \{(x, y, a) \in \mathcal{D} | a = i, y = j\}$:

$$L_{Adv}^{EO}(h) = 2 - \sum_{(i,j) \in \{0,1\}^2} \frac{1}{|\mathcal{D}_i^j|} \sum_{(x,a) \in \mathcal{D}_i^j} |h(f(x, a)) - a| \tag{2.7}$$

To achieve equal opportunity, we need only sum terms corresponding to $Y = 0$.

2.4.3 Motivation

For intuition on this approach and the upcoming theoretical section, we return to the framework from Section 2.2, with a data *owner* who sells representations to a (prediction) *vendor*. Suppose the data owner is concerned about the unfairness in the predictions made by vendors who use their data. Given that vendors are strategic actors with goals, the owner may wish to guard against two types of vendors:

- The *indifferent* vendor: this vendor is concerned with utility maximization, and doesn't care about the fairness or unfairness of their predictions.
- The *adversarial* vendor: this vendor will attempt to actively discriminate by the sensitive attribute.

In the adversarial model defined in Section 2.4.1, the encoder is what the data owner really wants; this yields the representations which will be sold to vendors. When the encoder is learned, the other two parts of the model ensure that the representations respond appropriately to each type of vendor: the classifier ensures utility by simulating an indifferent vendor with a prediction task, and the adversary ensures fairness by simulating an adversarial vendor with discriminatory goals. It is important to the data owner that the model's adversary be as strong as possible—if it is too weak, the owner will underestimate the unfairness enacted by the adversarial vendor.

However, there is another important reason why the model should have a strong adversary, which is key to our theoretical results. Intuitively, the degree of unfairness achieved by the adversarial vendor (who is optimizing for unfairness) will not be exceeded by the indifferent vendor. Beating a strong adversary h during training implies that downstream classifiers naively trained on the learned representation Z must also act fairly. Crucially, this fairness bound depends on the discriminative power of h; this motivates our use of the representation Z as a direct input to h, because it yields a strictly more powerful h and thus tighter unfairness bound than adversarially training on the predictions and labels alone as in Zhang et al. (2018).

2.5 Theoretical Properties

We now draw a connection between our choice of adversarial objective and several common metrics from the fair classification literature. We derive adversarial upper bounds on unfairness that can be used in adversarial training to achieve either demographic parity, equalized odds, or equal opportunity.

We are interested in quantitatively comparing two distributions corresponding to the learned group representations, so consider two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ over the same sample space $\Omega_{\mathcal{D}}$, as well as a binary test function $\mu : \Omega_{\mathcal{D}} \to \{0, 1\}$. μ is called a test since it can distinguish between samples from the two distributions according to the absolute difference in its expected value. We call this quantity the *test discrepancy* and express it as

$$d_\mu(\mathcal{D}_0, \mathcal{D}_1) \triangleq |\mathop{\mathbb{E}}_{x\sim\mathcal{D}_0}[\mu(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{D}_1}[\mu(x)]\,|. \tag{2.8}$$

The *statistical distance* (a.k.a. total variation distance) between distributions is defined as the maximum attainable test discrepancy (Cover and Thomas, 2012):

$$\Delta^*(\mathcal{D}_0, \mathcal{D}_1) \triangleq \sup_\mu d_\mu(\mathcal{D}_0, \mathcal{D}_1). \tag{2.9}$$

When learning fair representations we are interested in the distribution of Z conditioned on a specific value of group membership $A \in \{0, 1\}$. As a shorthand we denote the distributions $p(Z|A = 0)$ and $p(Z|A = 1)$ as $\mathcal{Z}_0$ and $\mathcal{Z}_1$, respectively.

2.5.1 Bounding Demographic Parity

In supervised learning we seek a g that accurately predicts some label Y; in fair supervised learning we also want to quantify g according to the fairness metrics discussed in Section 2.3. For example, the demographic parity distance is expressed as the absolute expected difference in classifier outcomes between the two groups:

$$\Delta_{DP}(g) \triangleq d_g(\mathcal{Z}_0, \mathcal{Z}_1) = |\mathbb{E}_{\mathcal{Z}_0}[g] - \mathbb{E}_{\mathcal{Z}_1}[g]|. \tag{2.10}$$

Note that $\Delta_{DP}(g) \leq \Delta^*(\mathcal{Z}_0, \mathcal{Z}_1)$, and also that $\Delta_{DP}(g) = 0$ if and only if $g(Z) \perp\!\!\!\perp A$, i.e., demographic parity has been achieved.

Now consider an adversary $h : \Omega_{\mathcal{Z}} \to \{0, 1\}$ whose objective (negative loss) function[5] is expressed as

$$L^{DP}_{Adv}(h) \triangleq \mathbb{E}_{\mathcal{Z}_0}[1 - h] + \mathbb{E}_{\mathcal{Z}_1}[h] - 1. \tag{2.11}$$

Given samples from $\Omega_{\mathcal{Z}}$ the adversary seeks to correctly predict the value of A, and learns by maximizing $L^{DP}_{Adv}(h)$. Given an optimal adversary trained to maximize (2.11), the adversary's loss will bound $\Delta_{DP}(g)$ from above for any function g learnable from Z. Thus a sufficiently powerful adversary h will expose through the value of its objective the demographic disparity of any classifier g. We will later use this to motivate learning fair representations via an encoder $f : \mathcal{X} \to \mathcal{Z}$ that simultaneously minimizes the task loss and minimizes the adversary objective.

[5] This is equivalent to the objective expressed by Equation 2.6 when the expectations are evaluated with finite samples.

Theorem. *Consider a classifier $g : \Omega_{\mathcal{Z}} \to \Omega_{\mathcal{Y}}$ and adversary $h : \Omega_{\mathcal{Z}} \to \Omega_{\mathcal{A}}$ as binary functions, i.e,. $\Omega_{\mathcal{Y}} = \Omega_{\mathcal{A}} = \{0, 1\}$. Then $L^{DP}_{Adv}(h^*) \geq \Delta_{DP}(g)$: the demographic parity distance of g is bounded above by the optimal objective value of h.*

Proof. By definition $\Delta_{DP}(g) \geq 0$. Suppose without loss of generality (WLOG) that $\mathbb{E}_{\mathcal{Z}_0}[g] \geq \mathbb{E}_{\mathcal{Z}_1}[g]$, i.e., the classifier predicts the "positive" outcome more often for group A_0 in expectation. Then, an immediate corollary is $\mathbb{E}_{\mathcal{Z}_1}[1-g] \geq \mathbb{E}_{\mathcal{Z}_0}[1-g]$, and we can drop the absolute value in our expression of the disparate impact distance:

$$\Delta_{DP}(g) = \mathbb{E}_{\mathcal{Z}_0}[g] - \mathbb{E}_{\mathcal{Z}_1}[g] = \mathbb{E}_{\mathcal{Z}_0}[g] + \mathbb{E}_{\mathcal{Z}_1}[1-g] - 1 \tag{2.12}$$

where the second equality is due to $\mathbb{E}_{\mathcal{Z}_1}[g] = 1 - \mathbb{E}_{\mathcal{Z}_1}[1-g]$. Now consider an adversary that guesses the opposite of g , i.e., $h = 1 - g$. Then[6], we have

$$\begin{aligned} L^{DP}_{Adv}(h) = L^{DP}_{Adv}(1-g) &= \mathbb{E}_{\mathcal{Z}_0}[g] + \mathbb{E}_{\mathcal{Z}_1}[1-g] - 1 \\ &= \Delta_{DP}(g) \end{aligned} \tag{2.13}$$

The optimal adversary $h^\star$ does at least as well as any arbitrary choice of h, therefore $L^{DP}_{Adv}(h^\star) \geq L^{DP}_{Adv}(h) = \Delta_{DP}$. ■

2.5.2 Bounding Equalized Odds

We now turn our attention to equalized odds. First we extend our shorthand to denote $p(Z|A = a, Y = y)$ as $\mathcal{Z}^y_a$, the representation of group a conditioned on a specific label y. The equalized odds distance of classifier $g : \Omega_{\mathcal{Z}} \to \{0, 1\}$ is

$$\begin{aligned} \Delta_{EO}(g) \triangleq\ & |\mathbb{E}_{\mathcal{Z}^0_0}[g] - \mathbb{E}_{\mathcal{Z}^0_1}[g]| \\ & + |\mathbb{E}_{\mathcal{Z}^1_0}[1-g] - \mathbb{E}_{\mathcal{Z}^1_1}[1-g]|, \end{aligned} \tag{2.14}$$

which comprises the absolute difference in false positive rates plus the absolute difference in false negative rates. $\Delta_{EO}(g) = 0$ means g satisfies equalized odds. Note that $\Delta_{EO}(g) \leq \Delta(\mathcal{Z}^0_0, \mathcal{Z}^0_1) + \Delta(\mathcal{Z}^1_0, \mathcal{Z}^1_1)$.

We can make a similar claim as above for equalized odds: given an optimal adversary trained on Z with the appropriate objective, if the adversary also receives the label Y, the adversary's loss will upper bound $\Delta_{EO}(g)$ for any function g learnable from Z.

Theorem. *Let the classifier $g : \Omega_{\mathcal{Z}} \to \Omega_{\mathcal{Y}}$ and the adversary $h : \Omega_{\mathcal{Z}} \times \Omega_{\mathcal{Y}} \to \Omega_{\mathcal{Z}}$, as binary functions, i.e., $\Omega_{\mathcal{Y}} = \Omega_{\mathcal{A}} = \{0, 1\}$. Then $L^{EO}_{Adv}(h^*) \geq \Delta_{EO}(g)$: the equalized odds distance of g is bounded above by the optimal objective value of h.*

Proof. Let the adversary h's objective be

$$\begin{aligned} L^{EO}_{Adv}(h) = & \mathbb{E}_{\mathcal{Z}^0_0}[1-h] + \mathbb{E}_{\mathcal{Z}^0_1}[h] \\ & + \mathbb{E}_{\mathcal{Z}^1_0}[1-h] + \mathbb{E}_{\mathcal{Z}^1_1}[h] - 2 \end{aligned} \tag{2.15}$$

By definition $\Delta_{EO}(g) \geq 0$. Let $|\mathbb{E}_{\mathcal{Z}^0_0}[g] - \mathbb{E}_{\mathcal{Z}^0_1}[g]| = \alpha \in [0, \Delta_{EO}(g)]$ and $|\mathbb{E}_{\mathcal{Z}^1_0}(1-g) - \mathbb{E}_{\mathcal{Z}^1_1}[1-g]| =$

[6] Before we assumed WLOG $\mathbb{E}_{\mathcal{Z}_0}[g] \geq \mathbb{E}_{\mathcal{Z}_1}[g]$. If instead $\mathbb{E}_{\mathcal{Z}_0}[g] < \mathbb{E}_{\mathcal{Z}_1}[g]$ then we simply choose $h = g$ instead to achieve the same result.

$\Delta_{EO}(g) - \alpha$. WLOG, suppose $\mathbb{E}_{\mathcal{Z}_0^0}[g] \geq \mathbb{E}_{\mathcal{Z}_1^0}[g]$ and $\mathbb{E}_{\mathcal{Z}_0^1}[1-g] \geq \mathbb{E}_{\mathcal{Z}_1^1}[1-g]$. Thus we can partition (2.14) as two expressions, which we write as

$$\begin{aligned} &\mathbb{E}_{\mathcal{Z}_0^0}[g] + \mathbb{E}_{\mathcal{Z}_1^0}[1-g] = 1 + \alpha, \\ &\mathbb{E}_{\mathcal{Z}_0^1}[1-g] + \mathbb{E}_{\mathcal{Z}_1^1}[g] = 1 + (\Delta_{EO}(g) - \alpha), \end{aligned} \tag{2.16}$$

which can be derived using the familiar identity $\mathbb{E}_p[\eta] = 1 - \mathbb{E}_p[1-\eta]$ for binary functions.

Now, let us consider the following adversary h

$$h(z) = \left\{ \begin{array}{ll} g(z), & \text{if } y = 1 \\ 1 - g(z), & \text{if } y = 0 \end{array} \right\}. \tag{2.17}$$

Then the previous statements become

$$\begin{aligned} &\mathbb{E}_{\mathcal{Z}_0^0}[1-h] + \mathbb{E}_{\mathcal{Z}_1^0}[h] = 1 + \alpha \\ &\mathbb{E}_{\mathcal{Z}_0^1}[1-h] + \mathbb{E}_{\mathcal{Z}_1^1}[h] = 1 + (\Delta_{EO}(g) - \alpha) \end{aligned} \tag{2.18}$$

Recalling our definition of $L_{Adv}^{EO}(h)$, this means that

$$\begin{aligned} L_{Adv}^{EO}(h) &= \mathbb{E}_{\mathcal{Z}_0^0}[1-h] + \mathbb{E}_{\mathcal{Z}_1^0}[h] + \mathbb{E}_{\mathcal{Z}_0^1}[h] + \mathbb{E}_{\mathcal{Z}_1^1}[h] - 2 \\ &= 1 + \alpha + 1 + (\Delta_{EO}(g) - \alpha) - 2 = \Delta_{EO}(g) \end{aligned} \tag{2.19}$$

That means that for the optimal adversary $h^\star = \sup_h L_{Adv}^{EO}(h)$, we have $L_{Adv}^{EO}(h^\star) \geq L_{Adv}^{EO}(h) = \Delta_{EO}$. ■

An adversarial bound for equal opportunity distance, defined as $\Delta_{EOpp}(g) \triangleq |\mathbb{E}_{\mathcal{Z}_0^0}[g] - \mathbb{E}_{\mathcal{Z}_1^0}[g]|$, can be derived similarly.

2.5.3 Additional points

One interesting note is that in each proof, we provided an example of an adversary which was calculated only from the joint distribution of Y and $\hat{Y} = g(Z)$—we did not require direct access to Z—and this adversary achieved a loss exactly equal to the quantity in question (Δ_{DP} or Δ_{EO}). Therefore, if we only allow our adversary access to those outputs, our adversarial objective (assuming an optimal adversary), is equivalent to simply adding either Δ_{DP} or Δ_{EO} to our classification objective, similar to common regularization approaches (Bechavod and Ligett, 2017, Kamishima et al., 2012, Madras et al., 2018b, Zafar et al., 2017b). Below we consider a stronger adversary, with direct access to the key intermediate learned representation Z. This allows for a potentially greater upper bound for the degree of unfairness, which in turn forces any classifier trained on Z to act fairly.

In our proofs we have considered the classifier g and adversary h as binary functions. In practice we want to learn these functions by gradient-based optimization, so we instead substitute their continuous relaxations $\tilde{g}, \tilde{h} : \Omega_{\mathcal{Z}} \to [0, 1]$. By viewing the continuous output as parameterizing a Bernoulli distribution over outcomes we can follow the same steps in our earlier proofs to show that in both cases (demographic parity and equalized odds) $\mathbb{E}[L(\bar{h}^*)] \geq \mathbb{E}[\Delta(\bar{g})]$, where $\bar{h}^*$ and $\bar{g}$ are randomized binary classifiers parameterized by the outputs of $\tilde{h}^*$ and $\tilde{g}$.

2.5.4 Comparison to Edwards and Storkey (2016)

An alternative to optimizing the expectation of the randomized classifier $\tilde{h}$ is to minimize its negative log likelihood (NLL - also known as cross entropy loss), given by

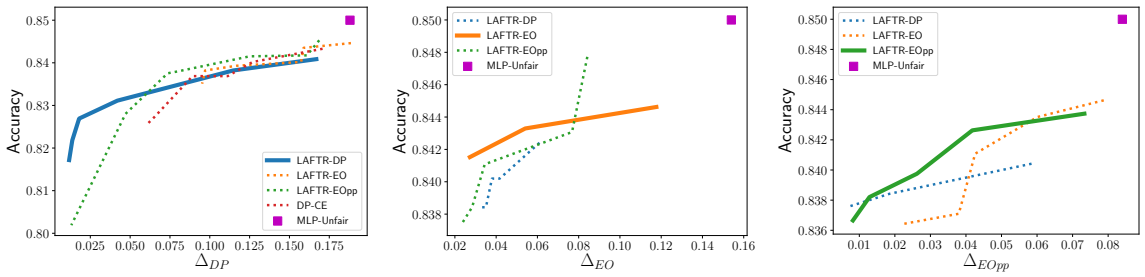
$$L(\tilde{h}) = -\mathbb{E}_{Z,A}\left[A \log \tilde{h}(Z) + (1-A)\log(1-\tilde{h}(Z))\right]. \tag{2.20}$$

This is the formulation adopted by Ganin et al. (2016) and Edwards and Storkey (2016), which propose maximizing (2.20) as a proxy for computing the statistical distance[7] $\Delta^*(\mathcal{Z}_0, \mathcal{Z}_1)$ during adversarial training.

The adversarial loss we adopt here instead of cross-entropy is group-normalized ℓ_1, defined in Equations 2.6 and 2.7. We choose group normalized ℓ_1 since it corresponds to a more natural relaxation of the fairness metrics in question. It is important that the adversarial objective incentivizes the test discrepancy, as group-normalized ℓ_1 does; this encourages the adversary to get an objective value as close to $\Delta^\star$ as possible, which is key for fairness (see Section 2.4.3). In practice, optimizing ℓ_1 loss with gradients can be difficult, so while we suggest it as a suitable theoretically-motivated continuous relaxation for our model (and present experimental results), there may be other suitable options beyond those considered in this work.

2.6 Experiments

2.6.1 Fair classification



(a) Tradeoff between accuracy and Δ_{DP} (b) Tradeoff between accuracy and Δ_{EO} (c) Tradeoff between accuracy and Δ_{EOpp}

Figure 2.3: Accuracy-fairness tradeoffs for various fairness metrics (Δ_{DP}, Δ_{EO}, Δ_{EOpp}), and LAFTR adversarial objectives ($L_{Adv}^{DP}, L_{Adv}^{EO}, L_{Adv}^{EOpp}$) on fair classification of the Adult dataset. Upper-left corner (high accuracy, low Δ) is preferable. Figure 2.3a also compares to a cross-entropy adversarial objective (Edwards and Storkey, 2016), denoted DP-CE. Curves are generated by sweeping a range of fairness coefficients γ, taking the median across 7 runs per γ, and computing the Pareto front. In each plot, the bolded line is the one we expect to perform the best. Magenta square is a baseline MLP with no fairness constraints. see Algorithm 1 and Appendix A.1.

LAFTR seeks to learn an encoder yielding fair representations, i.e., the encoder's outputs can be used by third parties with the assurance that their naively trained classifiers will be reasonably fair and accurate. Thus we evaluate the quality of the encoder according to the following training

[7]These papers discuss the bound on $\Delta_{DP}(g)$ in terms of the $\mathcal{H}$-divergence Blitzer et al. (2006), which is simply the statistical distance Δ^* up to a multiplicative constant.

Algorithm 1 Evaluation scheme for fair classification ($Y' = Y$) & transfer learning ($Y' \neq Y$).

Input: data $X \in \Omega_X$, sensitive attribute $A \in \Omega_A$, labels $Y, Y' \in \Omega_Y$, representation space Ω_Z
Step 1: Learn an encoder $f : \Omega_X \to \Omega_Z$ using data X, task label Y, and sensitive attribute A.
Step 2: Freeze f.
Step 3: Learn a classifier (without fairness constraints) $g : \Omega_Z \to \Omega_Y$ on top of f, using data $f(X')$, task label Y', and sensitive attribute A'.
Step 3: Evaluate the fairness and accuracy of the composed classifier $g \circ f : \Omega_X \to \Omega_Y$ on held out test data, for task Y'.

procedure, also described in pseudo-code by Algorithm 1. Using labels Y, sensitive attribute A, and data X, we train an encoder using the adversarial method outlined in Section 2.4, receiving both X and A as inputs. We then freeze the learned encoder; from now on we use it only to output representations Z. Then, using unseen data, we train a classifier on top of the frozen encoder. The classifier learns to predict Y from Z — note, this classifier is not trained to be fair. We can then evaluate the accuracy and fairness of this classifier on a test set to assess the quality of the learned representations.

During Step 1 of Algorithm 1, the learning algorithm is specified either as a baseline (e.g., unfair MLP) or as LAFTR, i.e., stochastic gradient-based optimization of (Equation 2.4) with one of the three adversarial objectives described in Section 2.4.2. When LAFTR is used in Step 1, all but the encoder f are discarded in Step 2. For all experiments we use cross entropy loss for the classifier (we observed training unstability with other classifier losses). The classifier g in Step 3 is a feed-forward MLP trained with SGD. See Appendix A.1 for details.

We evaluate the performance of our model[8]on fair classification on the UCI Adult dataset[9], which contains over 40,000 rows of information describing adults from the 1994 US Census. We aimed to predict each person's income category (either greater or less than 50K/year). We took the sensitive attribute to be gender, which was listed as Male or Female.

Figure 2.3 shows classification results on the Adult dataset. Each sub-figure shows the accuracy-fairness trade-off (for varying values of γ; we set $\alpha = 1, \beta = 0$ for all classification experiments) evaluated according to one of the group fairness metrics: Δ_{DP}, Δ_{EO}, and Δ_{EOpp}. For each fairness metric, we show the trade-off curves for LAFTR trained under three adversarial objectives: L_{Adv}^{DP}, L_{Adv}^{EO}, and L_{Adv}^{EOpp}. We observe, especially in the most important regime for fairness (small Δ), that the adversarial objective we propose for a particular fairness metric tends to achieve the best trade-off. Furthermore, in Figure 2.3a, we compare our proposed adversarial objective for demographic parity with the one proposed in (Edwards and Storkey, 2016), finding a similar result.

For low values of un-fairness, i.e., minimal violations of the respective fairness criteria, the LAFTR model trained to optimize the target criteria obtains the highest test accuracy. While the improvements are somewhat uneven for other regions of fairness-accuracy space (which we attribute to instability of adversarial training), this demonstrates the potential of our proposed objectives. However, the fairness of our model's learned representations are not limited to the task it is trained on. We now turn to experiments which demonstrate the utility of our model in learning fair representations for a variety of tasks.

[8]See https://github.com/VectorInstitute/laftr for code.
[9]https://archive.ics.uci.edu/ml/datasets/adult

2.6.2 Transfer Learning

In this section, we show the promise of our model for *fair transfer learning*. As far as we know, beyond a brief introduction in Zemel et al. (2013), we provide the first in-depth experimental results on this task, which is pertinent to the common situation where the data owner and vendor are separate entities.

We examine the Heritage Health dataset[10], which comprises insurance claims and physician records relating to the health and hospitalization of over 60,000 patients. We predict the Charlson Index, a comorbidity indicator that estimates the risk of patient death in the next several years. We binarize the (nonnegative) Charlson Index as zero/nonzero. We took the sensitive variable as binarized age (thresholded at 70 years old). This dataset contains information on sex, age, lab test, prescription, and claim details.

The task is as follows: using data X, sensitive attribute A, and labels Y, learn an encoding function f such that given unseen X', the representations produced by $f(X', A)$ can be used to learn a fair predictor for new task labels Y', even if the new predictor is being learned by a vendor who is indifferent or adversarial to fairness. This is an intuitively desirable condition: if the data owner can guarantee that predictors learned from their representations will be fair, then there is no need to impose fairness restrictions on vendors, or to rely on their goodwill.

The original task is to predict Charlson index Y fairly with respect to age A. The transfer tasks relate to the various primary condition group (PCG) labels, each of which indicates a patient's insurance claim corresponding to a specific medical condition. PCG labels $\{Y'\}$ were held out during LAFTR training but presumably correlate to varying degrees with the original label Y. The prediction task was binary: did a patient file an insurance claim for a given PCG label in this year? For various patients, this was true for zero, one, or many PCG labels. There were 46 different PCG labels in the dataset; we considered only used the 10 most common—whose positive base rates ranged from 9-60%—as transfer tasks.

Our experimental procedure was as follows. To learn representations that transfer fairly, we used the same model as described above, but set our reconstruction coefficient $\beta = 1$. Without this, the adversary will stamp out any information not relevant to the label from the representation, which will hurt transferability. We can optionally set our classification coefficient α to 0, which worked better in practice. Note that although the classifier g is no longer involved when $\alpha = 0$, the target task labels are still relevant for either equalized odds or equal opportunity transfer fairness.

We split our test set ($\sim 20,000$ examples) into transfer-train, -validation, and -test sets. We trained LAFTR ($\alpha = 0, \beta = 1$, ℓ_2 loss for the decoder) on the full training set, and then only kept the encoder. In the results reported here, we trained using the equalized odds adversarial objective described in Section 2.4.2; similar results were obtained with the other adversarial objectives. Then, we created a feed-forward model which consisted of our frozen, adversarially-learned encoder followed by an MLP with one hidden layer, with a loss function of cross entropy with no fairness modifications. Then, $\forall\ i \in 1 \dots 10$, we trained this feed-forward model on PCG label i (using the transfer-train and -validation) sets, and tested it on the transfer-test set. This procedure is described in Algorithm 1, with Y' taking 10 values in turn, and Y remaining constant ($Y \neq Y'$).

We trained four models to test our method against. The first was an MLP predicting the PCG label directly from the data (Target-Unfair), with no separate representation learning involved and

[10] https://www.kaggle.com/c/hhp

no fairness criteria in the objective—this provides an effective upper bound for classification accuracy. The others all involved learning separate representations on the original task, and freezing the encoder as previously described; the internal representations of MLPs have been shown to contain useful information Hinton and Salakhutdinov (2006). These (and LAFTR) can be seen as the values of ReprLearn in Alg. 1. In two models, we learned the original Y using an MLP (one regularized for fairness (Bechavod and Ligett, 2017), one not; Transfer-Fair and -Unfair, respectively) and trained for the transfer task on its internal representations. As a third baseline, we trained an adversarial model similar to the one proposed in Zhang et al. (2018), where the adversary has access only to the classifier output $\hat{Y} = g(Z)$ and the ground truth label (Transfer-Y-Adv), to investigate the utility of our adversary having access to the underlying representation, rather than just the joint classification statistics $(Y, A, \hat{Y})$.

We report our results in Figure 2.4 and Table 2.1. In Figure 2.4, we show the relative change from the high-accuracy baseline learned directly from the data for both classification error and Δ_{EO}. LAFTR shows a clear improvement in fairness; it improves Δ_{EO} on average from the non-transfer baseline, and the relative difference is an average of $\sim$20%, which is much larger than other baselines. We also see that LAFTR's loss in accuracy is only marginally worse than other models.

A fairly-regularized MLP ("Transfer-Fair") does not actually produce fair representations during trasnfer; on average it yields similar fairness results to transferring representations learned without fairness constraints. Another observation is that the output-only adversarial model ("Transfer Y-Adv") produces similar transfer results to the regularized MLP. This shows the practical gain of using an adversary that can observe the representations.

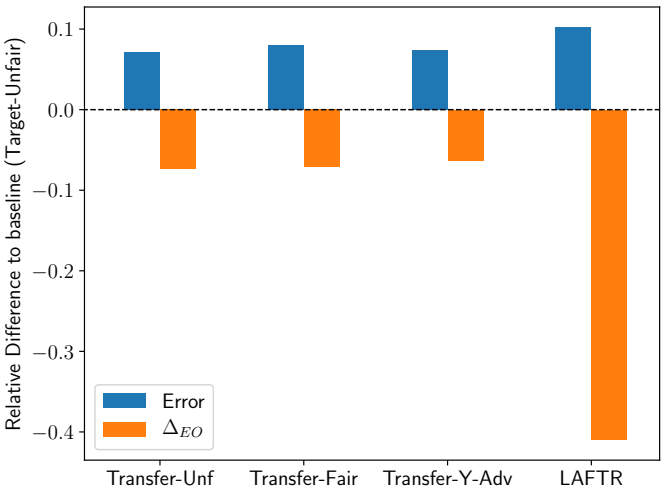


Figure 2.4: Fair transfer learning on Health dataset. Displaying average across 10 transfer tasks of relative difference in error and Δ_{EO} unfairness (the lower the better for both metrics), as compared to a baseline unfair model learned directly from the data. -0.10 means a 10% decrease. Transfer-Unf and -Fair are MLP's with and without fairness restrictions respectively, Transfer-Y-Adv is an adversarial model with access to the classifier output rather than the underlying representations, and LAFTR is our model trained with the adversarial equalized odds objective.

Since transfer fairness varied much more than accuracy, we break out the results of Fig. 2.4 in

Table 2.1: Results from Figure 2.4 broken out by task. Δ_{EO} for each of the 10 transfer tasks is shown, which entails identifying a primary condition code that refers to a particular medical condition. Most fair on each task is bolded. All model names are abbreviated from Figure 2.4; "TarUnf" is a baseline, unfair predictor learned directly from the target data without a fairness objective.

Tra. Task	TarUnf	TraUnf	TraFair	TraY-AF	LAFTR
MSC2a3	0.362	0.370	0.381	0.378	**0.281**
METAB3	0.510	0.579	**0.436**	0.478	0.439
ARTHSPIN	0.280	0.323	0.373	0.337	**0.188**
NEUMENT	0.419	0.419	0.332	0.450	**0.199**
RESPR4	0.181	0.160	0.223	0.091	**0.051**
MISCHRT	0.217	0.213	0.171	0.206	**0.095**
SKNAUT	0.324	**0.125**	0.205	0.315	0.155
GIBLEED	0.189	0.176	0.141	0.187	**0.110**
INFEC4	0.106	0.042	0.026	**0.012**	0.044
TRAUMA	0.020	0.028	0.032	0.032	**0.019**

Table 2.2: Transfer fairness, other metrics. Models are as defined in Figure 2.4. MMD is calculated with a Gaussian RBF kernel ($\sigma = 1$). AdvAcc is the accuracy of a separate MLP trained on the representations to predict the sensitive attribute; due to data imbalance an adversary predicting 0 on each case obtains accuracy of approximately 0.74.

Model	MMD	AdvAcc
Transfer-Unfair	1.1×10^{-2}	0.787
Transfer-Fair	1.4×10^{-3}	0.784
Transfer-Y-Adv ($\beta = 1$)	3.4×10^{-5}	0.787
Transfer-Y-Adv ($\beta = 0$)	1.1×10^{-3}	0.786
LAFTR	$\mathbf{2.7 \times 10^{-5}}$	**0.761**

Table 2.1, showing the fairness outcome of each of the 10 separate transfer tasks. We note that LAFTR provides the fairest predictions on 7 of the 10 tasks, often by a wide margin, and is never too far behind the fairest model for each task. The unfair model TraUnf achieved the best fairness on one task. We suspect this is due to some of these tasks being relatively easy to solve without relying on the sensitive attribute by proxy. Since the equalized odds metric is better aligned with accuracy than demographic parity (Hardt et al., 2016b), high accuracy classifiers can sometimes achieve good Δ_{EO} if they do not rely on the sensitive attribute by proxy. Because the data owner has no knowledge of the downstream task, however, our results suggest that using LAFTR is safer than using the raw inputs; LAFTR is relatively fair even when TraUnf is the most fair, whereas TraUnf is dramatically less fair than LAFTR on several tasks.

We provide coarser metrics of fairness for our representations in Table 2.2. We give two metrics: maximum mean discrepancy (MMD) (Gretton et al., 2007), which is a general measure of distributional distance; and adversarial accuracy (if an adversary is given these representations, how well can it learn to predict the sensitive attribute?). In both metrics, our representations are more fair than the baselines. We give two versions of the "Transfer-Y-Adv" adversarial model ($\beta = 0, 1$); note that it has much better MMD when the reconstruction term is added, but that this does not improve its adversarial accuracy, indicating that our model is doing something more sophisticated than simply matching moments of distributions.

2.7 Conclusion

In this chapter, we proposed and explore methods of learning adversarially fair representations. We provided theoretical grounding for the concept, and proposed novel adversarial objectives that guarantee performance on commonly used metrics of group fairness. Experimentally, we demonstrated that these methods can learn fair and useful predictors through using an adversary on the intermediate representation. We also demonstrate success on fair transfer learning, by showing that our methods can produce representations which transfer utility to new tasks as well as yielding fairness improvements.

Since the publication of this work, a number of other works have built on this line of this work, exploring and extending the concept of fair representation learning. One key exploration was to the question of multiple-sensitive attributes (Creager et al., 2020b), providing an analogue to multi-group fairness in the representation space. Creager et al. (2019) approach this question using techniques from disentanglement and variational learning. Relatedly, Agarwal et al. (2021) extend fair representation learning to structured data, looking at graph representation learning. Quadrianto et al. (2019) explore how to learn fair representations of data in the original data domain, understanding the interpretability implications of these methods, and Adel et al. (2019) examine how to incorporate adversarial fair learning within a single network. Another key question is explored by Song et al. (2019) and Moyer et al. (2018), who both use mutual information-inspired approaches to derive objectives where the amount of sensitive information left in the representations can be better controlled. Several works examine how to think about fairness in the context of disentangled representations, notably Träuble et al. (2021) and Locatello et al. (2019). Finally, works such as Samadi et al. (2018) and Zhao and Gordon (2022) examine the theoretical tradeoffs within learning fair representations between fairness and utility.

Several open problems remain around the question of learning representations fairly. In particular, the question of transfer learning in this setting remains paramount. In some setups, it seems as though transfer learning is too much to ask for — for instance, we show in Section 2.6.2 that we achieve some transfer fairness with respect to the equalized odds metric. This is not possible in the general case if the two tasks are selected adversarially, since the metric itself depends on the task labels (Lechner et al., 2021). However, for a metric like demographic parity, transfer seems more plausible. It is unclear where exactly other metrics fall along this spectrum: for instance, do causal metrics such as counterfactual fairness (Kusner et al., 2017) lend themselves to transferability? Another relevant factor is the range of tasks: which tasks can have good transfer accuracy on which fair representations? For instance, if a representation has been made full fair with respect to demographic parity, then it will not be able to transfer well to a task that is correlated with the sensitive attribute. However, a metric like equalized odds, which can co-exist with perfect accuracy, may be amenable to a higher range of downstream high-accuracy tasks when applied to a fair representation.

Additionally, the question of how to learn these representations in a stable way is a major challenge — we found in our experiments with L1 loss that that learning was fairly unstable, which makes deployment difficult. While our proposed L1 loss has positive theoretical properties, potential alternatives exist which could open the door to more reliable and stable methods. One such approach is non-adversarial methods, with the most common one being MMD regularization (Gretton et al., 2007) (see Louizos et al. (2016) or Veitch et al. (2021) for examples). MMD regularization can be thought of as an adversarial approach with a more limited class that the adversary can choose from (a unit ball in a reproducing kernel Hilbert space). The downside of this is that a more powerful

adversary could potentially uncover sensitive information, even when MMD = 0; however, the stability it offers in learning is a major practical upside. Another possible direction is to return to the cross-entropy loss, which has been used successfully in other adversarial approaches (Edwards and Storkey, 2016), potentially using adaptations that have been effective for unbalanced classification problems elsewhere (Cao et al., 2019) A careful in-depth comparison of these approaches would help elucidate their pros and cons.

Chapter 3

Detecting Underspecification

3.1 Underspecification

In the previous chapter, we discussed the situation where we have a single attribute whose correlation with the label or representation we hope to remove. Here, we turn to a more general notion of unreliability: *underspecification*. D'Amour et al. (2020) discuss and define the concept as follows (emphasis ours):

> ... predictors trained to the same level of i.i.d. generalization will often show widely divergent behavior when applied to real-world settings ... In general, the solution to a problem is underspecified if there are many distinct solutions that solve the problem equivalently ... In the context of ML, **we say an ML pipeline is underspecified if there are many distinct ways (...) for the model to achieve equivalent held-out performance on iid data, even if the model specification and training data are held constant**.

Here, we extend this definition to make reference to a particular input, and say that a trained model is underspecified *at a test input* if many different predictions at that input are all equally consistent with the constraints posed by the training data and the learning problem specification (i.e., the model architecture and the loss function).

3.1.1 Spurious Correlations as a Special Case

We can think of underspecification as a generalization of the notion of "spurious correlations". When a spurious correlation exists between some variable C and label Y on some task, this means that there are (at least) two possible predictors which achieve good performance on this task (assuming we are selecting from a large enough hypothesis class): the true prediction function, and one which predicts using information from C. This yields underspecification on test inputs where the spurious correlation is broken: the true prediction function will output the true label, and the spurious prediction function will output the one indicated by C (which will be different, since we assumed the spurious correlation was broken).

Spurious correlations have also been shown to present problems for trustworthiness. Winkler et al. (2019) provide a particularly salient example in the medical imaging setting. They audited the

performance of a CNN which was "approved for use as a medical device in the European market" on the task of skin lesion classification, with the model providing a melanoma diagnosis. Concerningly, they found that this model was particularly sensitive to violet skin markings, which are more commonly found on problematic lesions due to their use in medical practice. Winkler et al. (2019) tested this model on lesions for which they had both marked and unmarked versions, and found that the model's false positive rate increased drastically on marked lesions, indicating that the model had learned this incorrect association strongly.

Robustness to spurious correlations are closely tied to questions of fairness (see Creager et al. (2020a) for a more general overview of the connections between these fields). For instance, we discussed previously that some machine translation systems have been shown to make unwanted inferences around gender, e.g. assuming that if a person in a sentence is a nurse, then that person should be assumed female, even if no gender information is provided (Cho et al., 2019, Kuczmarski, 2018). We can interpret this inference as learning of an undesired statistical relationship in the training corpus: most nurses are female (91% in Canada (CNA, 2020)). However, this not due to intrinsic properties of the definition of a nurse, but rather, characteristics of the particular data generating process which produced the corpus i.e. gender roles in a particular social context and their relationship to career choice (McLaughlin et al., 2010). Depending on the values we want our system to support, we may not want it to amplify these existing stereotypes.

3.1.2 The Challenge of Uninterpretable Correlations

In some sense, these popular case studies of spurious correlations represent the *easy* case of underspecification, since they are relatively interpretable — we can easily grasp the idea that gender pronouns and occupations may be correlated, or that skin markings may be more frequently placed on lesion types. However, we may not always be able to visualize these correlations so easily. Gichoya et al. (2022) show that ML models are able to predict race from chest X-rays at a much higher accuracy than medical experts can, despite there being no known proxy for race which is visible from these images. That such prediction performance is possible but invisible to the human eye means that racial disparities in performance could arise for reasons that we are unable to understand or control for.

We emphasize that underspecification, as a general phenomenon, does not occur with any type of interpretable "signature" in the data (e.g. a visible marking, or even the concept of "race") with which to diagnose the issue. Underspecification frequently arises without being able to pinpoint a cause in terms of known spurious correlations among human-readable variables. In a large-scale empirical study, D'Amour et al. (2020) show that underspecification has the potential to be a large practical roadblock to building trustworthy ML systems, in ways that are completely uninterpretable to humans. They show through case studies in computer vision, medical imaging, natural language processing, and electronic health records prediction, that models trained to the same level of test set performance can perform vastly differently on various *stress tests*, i.e. evaluation sets designed to examine a model's performance on a task which is slightly outside the standard distribution. For example, Fig. 3.1 shows how an ensemble of Resnet-50's, using the same architecture and model training process, achieve vastly different scores on tasks such as classification of pixelated images, despite achieving nearly identical scores on the Imagenet test set. Identifying the mechanism by which this occurs, i.e. how you would re-sample the Imagenet dataset to lower the ensemble variance

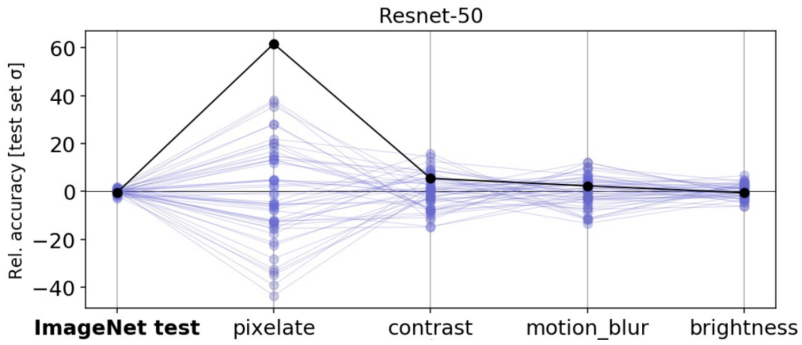


Figure 3.1: From D'Amour et al. (2020). An ensemble of Resnet-50s, trained to the same level of test set performance using identical architecture and training processes and varying only the random seed, can nonetheless vary widely in performance on a range of stress tests. The stress tests here all use Imagenet images with some modification: added pixelation, decreased or increased contrast, added blur, and modified brightness level (from L to R). Y-axis is in terms of standard deviation of ensemble accuracies on the standard Imagenet test set.

on pixelate images, seems like a hopeless task to approach only with human intuitions[1].

Given the difficulty (or impossibility) of visualizing many cases of underspecification, or discovering the underlying causes through research or independent analysis, developing automated methods for detecting when underspecification is an issue is paramount. This is the question we turn to in the rest of this chapter.

3.2 Detecting Underspecification

Underspecification is particularly relevant in the context of overparameterized model classes (e.g. deep neural networks (Neyshabur et al., 2018)). It has been shown that these problems arise frequently in these models, and that many deep neural networks can be trained which perform similarly well on some training metric, but vastly differently on that test metric D'Amour et al. (2020). This aligns with our intuition around parametrization: the more parameters, the more different "interpretations" of the data there are.

This can be seen in the simplest case, with linear regression: if we duplicate some dimension (hence adding one parameter, but no extra information), our resulting model will be underspecified. [2] In more complex model classes like deep neural networks, it is harder to construct interpretable examples, but it seems obvious that the extra degrees of freedom provided by overparameterization could not reduce these issues. [3]

Underspecification can be an obstacle to trust — if a model's specification could easily lead to

[1] Of course, having observed this issue, one could augment the original Imagenet training set with pixelated images, but the lack of a human-readable method for identifying issues beforehand by examining the training data will always leave models vulnerable to unknown failure modes at deployment time which are not covered by the given evaluation set(s).

[2] Since X will have linearly dependent columns, $X^\top X$ will be non-invertible and there will be many solutions β to $X^\top X\beta = X^\top Y$.

[3] Deep neural networks are known to be underspecified at least up to a permutation, since hidden units can always be re-arranged (Hou et al., 2019), and even the best identifiability guarantees for flexible latent variable models only hold up to a permutation and a simple transformation (Hyvarinen et al., 2019, Khemakhem et al., 2020). However, we note that simple permutation underspecification will not necessarily affect trustworthiness, as we can permute hidden units without affecting the actual function the model computes.

many predictions on some example, the user's ability to anticipate the model's impact is greatly impeded. There are a number of potential approaches one could take to mitigate this. We could consider using domain information to "break ties", i.e. add some external information to the learning problem so that the model's prediction is more well-specified — we can think of approaches that add group or environment side-information (Peters et al., 2016, Sagawa et al., 2019) as implicitly accomplishing a similar objective. We could simply output many different possible predictions if they exist, and allow the user to use that information for their own purposes however they choose (Lee et al., 2022). In this chapter, we take a different approach: we hope to flag examples where underspecification could be an issue, potentially allowing for some further investigation.

This approach calls for a method of measuring underspecification in a given model. Simple (but computationally expensive) ensembling methods (Lakshminarayanan et al., 2017), which train many models on the same data from different random seeds, provide an effective measure of underspecification and have proven highly effective at uncertainty quantification tasks (Ovadia et al., 2019). These approaches specifically measure the underspecification induced by some model class, since they hold the problem specification constant and only change the source of randomness. The effectiveness of these methods motivates flexible methods that can detect underspecified predictions *cheaply*.

With this motivation, in this chapter we ask the question: can we detect underspecification in a fully post-hoc manner? Post-hoc methods for this task would be particularly useful, since we do not always have the ability to re-train a model using a different (e.g. ensemble) procedure, whether due to privacy concerns around data, computational constraints, or simply because we want to audit some already-existing model. To this end, we present *local ensembles*, a post-hoc method for measuring the extent to which a pre-trained model's prediction is underspecified for a particular test input. Given a trained model, our method returns an underspecification score that estimates the variability of test predictions across a *local ensemble*, i.e. a set of local perturbations of the trained model parameters that fit the training data equally well. Local ensembles are a computationally cheap, post-hoc alternative to fully trained ensembles and approximate Bayesian ensembling methods that require special training procedures (Blundell et al., 2015, Gal and Ghahramani, 2016). Local ensembles also address a gap in approximate methods for estimating prediction uncertainty. Specifically, whereas exact Bayesian or Frequentist uncertainty includes underspecification as one component, approximate methods such as Laplace approximations (MacKay, 1992) or influence function-based methods (Schulam and Saria, 2019) break down when underspecification is present. In contrast, our method leverages the pathology (an ill-conditioned Hessian) that makes these methods struggle.

3.3 Underspecification Score and Local Ensembles

3.3.1 Setup

Let $z = (x, y)$ be an example input-output pair, where x is a vector of features and y is a label. We define a model in terms of a loss function $\mathcal{L}$ with parameters θ as a sum over training examples $(z_i)_{i=1}^n$, i.e., $\mathcal{L}(\theta) = \sum_i^n \ell(z_i, \theta)$, where ℓ is an example-wise loss (e.g., mean-squared error or cross entropy). Let $\theta^\star$ be the parameters of the trained model, obtained by, e.g., minimizing the loss over this dataset, i.e., $\theta^\star = \text{argmin}_\theta \mathcal{L}(\theta)$. We write the prediction function given by parameters θ at an

input x as $\hat{y}(x, \theta)$. We consider the problem of auditing a trained model, where unlabeled test points x' arrive one at a time in a stream, and we wish to assess underspecification on a point-by-point basis.

In this section, we introduce our local ensemble underspecification score $\mathcal{E}_m(x')$ for an unlabeled test point x' (significance of m explained below). The score is designed to measure the variability that would be induced by randomly choosing predictions from an ensemble of models with similar training loss. Our score has a simple form: it is the norm of the prediction gradient $g_{\theta^\star}(x') := \nabla_\theta \hat{y}(x', \theta^\star)$ multiplied by a matrix of Hessian eigenvectors spanning a subspace of low curvature U_m (defined in more detail below).

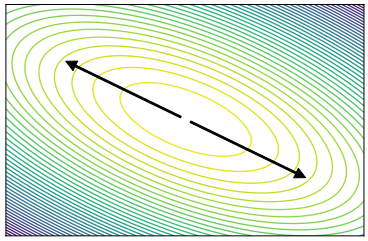


Figure 3.2: In this quadratic bowl, arrows denote the small eigendirection, where predictions are slow to change. We argue this direction is key to underspecification.

$$\mathcal{E}_m(x') = \|U_m^\top g_{\theta^\star}(x')\|_2 \tag{3.1}$$

Here, we show that this score is proportional to the standard deviation of predictions across a local ensemble of models with near-identical training loss, and demonstrate that this approximation holds in practice.

3.3.2 Derivation

Our derivation proceeds in two steps. First, we define a local ensemble of models with similar training loss, then we state the relationship between our underspecification score and the variability of predictions within this ensemble.

The spectral decomposition of the Hessian $H_{\theta^\star}$ plays a key role in our derivation. Let

$$H_{\theta^\star} = U \Lambda U^\top, \tag{3.2}$$

where U is a square matrix whose columns are the orthonormal eigenvectors of $H_{\theta^\star}$, written $(\xi_{(1)}, \cdots, \xi_{(p)})$, and Λ is a square, diagonal matrix with the eigenvalues of $H_{\theta^\star}$, written $(\lambda_{(1)}, \cdots, \lambda_{(p)})$, along its diagonal. As a convention, we index the eigenvectors and eigenvalues in decreasing order of the eigenvalue magnitude.

To construct a local ensemble of loss-preserving models, we exploit the fact that eigenvectors with large corresponding eigenvalues represent directions of high curvature, whereas eigenvectors with small corresponding eigenvalues represent directions of low curvature. Thus, under the assumption that the model has been trained to a local minimum or saddle point, parameter perturbations in flat directions (those corresponding to small eigenvalues $\lambda_{(j)}$) do not change the training loss substantially (see Fig. 3.2). We characterize this subspace by the span of eigenvectors with corresponding small eigenvalues. Formally, let m be the eigenvalue index such that the eigenvalues $\{\lambda_{(j)} : j > m\}$, are sufficiently small to be considered "flat"[4]. We call the subspace spanned by $\{\xi_{(j)} : j \in \sigma\}$ the *ensemble subspace*. Parameter perturbations in the ensemble subspace generate an ensemble of models with near-identical training loss[5].

[4] For now, we take m to be given, and discuss tradeoffs for choosing m in practice in Section 3.4.

[5] We note that not all loss-preserving ensembles will necessarily be local; however, the existence of some such local ensemble is sufficient to enable underspecification.

Our goal is to characterize the variance of test predictions with respect to random parameter perturbations Δ_θ that occur in the ensemble subspace. We now show that our underspecification score $\mathcal{E}_m(x')$ is, to the first order, proportional to the standard deviation of predictions at a point x'.

Proposition 1. *Let Δ_θ be the projection of a random perturbation with mean zero and covariance proportional to the identity $\epsilon \cdot I$ into the ensemble subspace spanned by $\{\xi_{(j)} : j > m\}$. Let P_Δ be the linearized change in prediction induced by the perturbation*

$$P_\Delta(x') := g_{\theta^\star}(x')^\top \Delta_\theta \approx \hat{y}(x', \theta^\star + \Delta_\theta) - \hat{y}(x', \theta^\star).$$

Then $\mathcal{E}_m(x') = \epsilon^{-1/2} \cdot SD(P_\Delta(x'))$.

Proof. Let U_m be the matrix whose columns are $\{\xi_{(j)} : j > m\}$. Then $U_m U_m^\top$ is a projection matrix that projects vectors into the ensemble subspace, and Δ_θ has covariance $\epsilon \cdot U_m U_m^\top$. It follows that

$$\begin{aligned}
\epsilon^{-1} Var(P_\Delta) &= \epsilon^{-1} Var(g_{\theta^\star}(x')^\top \Delta) \\
&= \epsilon^{-1} g_{\theta^\star}(x')^\top Var(\Delta) g_{\theta^\star}(x') \\
&= \epsilon^{-1} g_{\theta^\star}(x')^\top (\epsilon U_m U_m^\top) g_{\theta^\star}(x') \\
&= \|U_m^\top g_{\theta^\star}(x')\|_2^2 \\
&= \mathcal{E}_m(x')^2
\end{aligned}$$

□

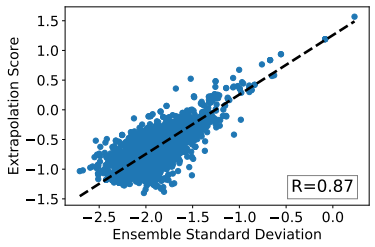


Figure 3.3: Mean underspecification score vs. ensemble prediction standard deviation on WineQuality dataset. Shows line of best fit and Pearson coefficient R. Both axes log-scaled.

We test this hypothesized relationship on several tabular datasets. We train an ensemble of twenty neural networks using the same architecture and training set, varying only the random seed. Then, for each model in the ensemble, we calculate our local ensemble underspecification score $\mathcal{E}_m(x)$ for each input x' in the test set (see Sec. 3.4 for details). For each x', we compare, across the ensemble, the mean value of $\mathcal{E}_m(x')$ to the standard deviation $\hat{y}(x')$. In Fig. 3.3, we plot these two quantities against each other for one of the datasets, finding a nearly linear relationship. On each dataset, we found a similar, significantly linear relationship (see Table 3.1 and Appendix B.1). We note the relationship is weaker with the Diabetes dataset; the standard deviations of these ensembles are an order of magnitude higher than the other datasets, indicating much noisier data.

Finally, we note that we can obtain similar results for the standard deviation of the loss at test points if we redefine g as the loss gradient rather than the prediction gradient.

3.4 Computing Underspecification Scores

We now discuss a practical method for computing our underspecification scores. The key operation is constructing the set of eigenvectors that span a suitably loss-preserving ensemble subspace (i.e. have sufficiently small corresponding eigenvalues). Because eigenvectors corresponding to large eigenvalues are easier to estimate, we construct this subspace by finding the top m eigenvectors and defining the ensemble subspace as their orthogonal complement. Our method can be implemented with any algorithm that returns the top m eigenvectors. See below for discussion on some tradeoffs in the choice of m, as well as the algorithm we choose (the Lanczos iteration (Lanczos, 1950)).

Dataset	**Pearson**
Boston	0.76
Diabetes	0.50
Abalone	0.76
Wine	0.87

Table 3.1: Correlation of underspecification scores and ensemble std. deviations on 4 datasets.

Our method proceeds as follows. For a given choice of m, we calculate the m eigenvectors of H with the *largest* eigenvalues. These eigenvectors define an m-dimensional subspace which is the orthogonal complement to the ensemble subspace. We use these eigenvectors to construct a matrix that projects gradients into the ensemble subspace. Specifically, let $U_{m^\perp}$ be the matrix whose columns are these large-eigenvalue eigenvectors $\{\xi_{(j)} : j \leq m\}$. Then $U_{m^\perp}U_{m^\perp}^\top$ is the projection matrix that projects vectors into the "large eigenvalue" subspace, and $I - U_{m^\perp}U_{m^\perp}^\top$ projects into its complement: the ensemble subspace. Now, for any test input x', we take the norm of this projected gradient to compute our score

$$\mathcal{E}_m(x') = \left\| \left(I - U_{m^\perp}U_{m^\perp}^\top\right) g_{\theta^\star}(x') \right\|.$$

The success of this approach depends on the eigenspectrum of the Hessian and the choice of m. Specifically, the underspecification score $\mathcal{E}_m(x')$ is the most sensitive to underspecification in the region of the trained parameters if we set m to be the smallest index for which the training loss is relatively flat in the implied ensemble subspace. If m is set too low, the ensemble subspace will include well-constrained directions, and $\mathcal{E}_m(x')$ will over-estimate the prediction's sensitivity to loss-preserving perturbations. If m is set too high, the ensemble subspace will omit some under-constrained directions, and $\mathcal{E}_m(x')$ will be less sensitive. For models where all parameters are well-constrained by the training data, a suitable m may not exist. This will usually not be the case for deep neural network models, which are known to have very ill-conditioned Hessians (see, e.g., Sagun et al., 2017). It may be possible to develop diagnostics for choosing m based on trace estimation techniques (Hutchinson, 1990) or eigenvalue distributions (Marčenko and Pastur, 1967, Martin and Mahoney, 2021, Wigner, 1967), but we leave this as future work.

We use the Lanczos iteration, a method for tridiagonalizing a Hermitian matrix, (Lanczos, 1950) to estimate the top m eigenvectors. Once we have achieved this tridiagonalization, we can find the eigendecomposition of the tridiagonalized matrix more easily, and then convert it back to an eigendecomposition of the original matrix. Specifically, suppose our Hessian matrix H decomposes as $H = VTV^\top$, where T is tridiagonal and V is orthonormal. If w is an eigenvector of T with eigenvalue q, then Vw is an eigenvalue of H with the same eigenvalue.

The Lanczos iteration presents a number of practical advantages for usage in our scenario. Firstly, it performs well under early stopping, returning good estimates of the top m eigenvectors after m iterations. Secondly, we can cache intermediate steps, meaning that computing the $m+1$-th eigenvector is fast once we have computed the first m. Thirdly, it requires only implicit access to the Hessian through a function which applies matrix multiplication, meaning we can take advantage of

efficient Hessian-vector product methods (Pearlmutter, 1994).

Finally, the Lanczos iteration is simple – it can be implemented in less than 20 lines of Python code (see Appendix B.2.1). It contains only one hyperparameter, the stopping value m. Fortunately, tuning this parameter is efficient — given a maximum value M, we can try many values $m < M$ at once, by estimating M eigenvectors and then calculating $\mathcal{E}_m$ by using the first m eigenvectors. The main constraint of our method is space rather than time — while estimating the first m eigenvectors enables easy caching for later use, it may be difficult to work with these eigenvectors in memory as m and model size p increase. This tradeoff informed our choice of m in this paper; we note in some cases that increasing m further could have improved performance (see Appendix B.5). This suggests that further work on techniques for mitigating this tradeoff, e.g. online learning of sparse representations (Wang and Lu, 2016, Wang et al., 2012), could improve the performance of our method. See Appendix B.2 for more details on the Lanczos iteration.

3.5 Related Work

3.5.1 Relation to Bayesian and Frequentist Second-Order Methods

It is instructive to compare our underspecification score to two other approximate reliability quantification methods that are aimed at Bayesian and Frequentist notions of underspecification, respectively. Like our underspecification score, both of these methods make use of local information in the Hessian to make an inference about the variance of a prediction. First, consider the Laplace approximation of the posterior predictive variance. This metric is derived by interpreting the loss function as being equivalent to a Bayesian log-posterior distribution over the model parameters θ, and approximating it with a Gaussian. Specifically, (see, e.g., MacKay, 1992)

$$Var(y \mid x') \approx g_{\theta^\star}(x')^\top H_{\theta^\star}^{-1} g_{\theta^\star} = \sum_{j=1}^{p} \lambda_{(j)}^{-1} \left(\xi_{(j)}^\top g_{\theta^\star}\right)^2. \tag{3.3}$$

Second, consider scores such as RUE (Resampling Under Uncertainty) designed to approximate the variability of predictions by resampling the training data (Schulam and Saria, 2019). These methods approximate the change in trained parameter values induced by perturbing the training data via influence functions (Cook and Weisberg, 1982, Koh and Liang, 2017). Specifically, the gradient of the parameters with respect to the weight of a given training example z_i is given by

$$I(z_i) = -H_{\theta^\star}^{-1} \nabla_\theta \ell(z_i, \theta^\star) = \sum_{j=1}^{p} \lambda_{(j)}^{-2} \left(\xi_{(j)}^\top \nabla_\theta \ell(z_i, \theta^\star)\right)^2. \tag{3.4}$$

Schulam and Saria (2019) combine this influence function with a specific random distribution of weights to approximate the variance of predictions under bootstrap resampling; other similar formulations are possible, sometimes with theoretical guarantees (Alaa and Van Der Schaar, 2020, Giordano et al., 2019). Lu et al. (2020) use these types of estimators to construct a pseudo-ensemble of leave-one-out predictors.

Importantly, both of these methods work well when model parameters are well-constrained by the training data, but they struggle when predictions are (close to) underspecificied. This is because, in the

presence of underspecification, the Hessian becomes ill-conditioned. Practical advice for dealing with this ill-conditioning is available (Koh and Liang, 2017), but we note that this not merely a numerical pathology; by our argument above, a poorly conditioned Hessian is a clear signal of underspecification. In contrast to these methods, our method focuses specifically on prediction variability induced by underconstrained parameters. Our underspecification score incorporates *only* those terms with small eigenvalues, and removes the inverse eigenvalue weights that make inverse-Hessian methods break down. This is clear from its summation representation: $\mathcal{E}_m(x') = \sum_{j>m} \left(\xi_{(j)}^\top g_{\theta^\star}\right)^2$.

Our method also has computational advantages over approaches that rely on inverting the Hessian. Firstly, implicitly inverting the Hessian is a complex and costly process — by finding only the important components of the projection explicitly, our method is simpler and more efficient. Furthermore, we only need to find these components once; we can reuse them for future test inputs. This type of caching is not possible with methods which require us to calculate the inverse Hessian-vector product for each new test input. Other recent work aims at improving the scalability of manipulating the Hessian, one through exploring the Arnoldi iteration (a generalization which, when applied to Hermitian matrices, reduces to the Lanczos iteration - our approach) (Schioppa et al., 2021), and another through using a Kronecker-factored approximation (Ritter et al., 2018).

3.5.2 Related Work in Detecting Underspecification

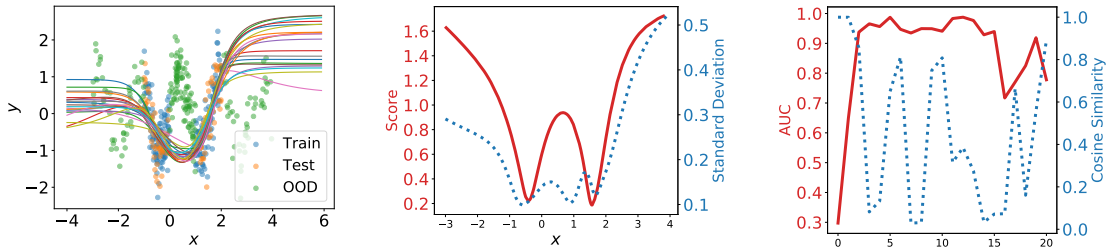
Most closely related to our work is a line of work on "Rashomon sets", which explores loss-preserving ensembles in simpler model classes more formally (Fisher et al., 2019, Semenova et al., 2019). The Rashomon set is a slightly different notion from underspecification: in underspecification, we fix a model class where as a Rashomon set can include models from many different model classes. A related concept of "predictive multiplicity", the propensity of a problem specification to yield a range of disagreeing models, is proposed and studied by Marx et al. (2020).

Some recent works explore the relationship between test points, the learned model, and the training set. Several papers examine reliability criteria that are based on distance in some space: within/between-group distances (Jiang et al., 2018), a pre-specified kernel in a learned embedding space (Card et al., 2019), or the activation space of a neural network (Papernot and McDaniel, 2018). We implement some nearest-neighbor baselines inspired by this work in Sec. 3.6. Additionally, a range of methods exist for related tasks, such as OOD detection (Choi et al., 2018, Gal and Ghahramani, 2016, Liang et al., 2018, Schölkopf et al., 2001) and calibration (Guo et al., 2017, Naeini et al., 2015). Some work using generative models for OOD detection makes use of related second-order analysis (Nalisnick et al., 2018). Sastry and Oore (2020) use Gram matrices to perform OOD detection in a related approach. A line of work explores the benefits of training ensemble methods explicitly, discussed in detail in Dietterich (2000). These methods have been discussed for usage in some of the applications we present in Sec. 3.6, including uncertainty detection (Lakshminarayanan et al., 2017), active learning (Melville and Mooney, 2004) and OOD detection (Choi et al., 2018), by leveraging the utility of an ensemble's variance to be an effective proxy for epistemic uncertainty. Ovadia et al. (2019) show that ensembles are among the most reliable predictors of uncertainty, particularly as distributions shift — this suggests that underspecification (as captured by ensembles) is a particularly robust notion of unreliability.

3.6 Experiments

In this section, we give evidence that local ensembles can detect underspecification in trained models. If we can detect that a model is underspecified on some input in practice, we can use that as helpful signal that the model's prediction may be less trustworthy, and that we may need to collect more data which is relevant to that input example. As discussed in the introduction to this chapter, underspecification is difficult to evaluate, since it is often challenging to observe in a given model. In order to explicitly evaluate underspecification, we present a range of experiments where a pre-trained model has a known "blind spot", i.e. a portion of the input distribution which is not well-represented in the training set, and evaluate its ability to detect when an input is in that blind spot. We probe our method's ability to detect a range of underspecification, exploring cases where the blind spot is: **1.** easily visualized, **2.** well-defined by the feature distribution, **3.** well-defined by a latent distribution, and **4.** unknown, but where we can evaluate our model's detection performance through an auxiliary task. See Appendix B.4 for experimental details. Code for running the local ensembles method can be found at `https://github.com/dmadras/local-ensembles`.

3.6.1 Visualizing Underspecification Detection



(a) Data and trained model (b) Underspecification scores (c) Eigenvector estimates & AUC

Figure 3.4: We train a neural network ensemble (Fig. 3.4a). We compute underspecification scores (solid line), which correlate with the standard deviation of the ensemble (dotted line) (Fig. 3.4b). Our OOD performance achieves high AUC (solid line) even though some of our eigenvector estimates have low cosine similarity to ground truth (dotted line) (Fig. 3.4c).

We begin with an easily visualized toy experiment. In Fig. 3.4a, we show our data ($y = \sin 4x + \mathcal{N}(0, \frac{1}{4})$). We generate in-distribution training data from $x \in [-1, 0] \cup [1, 2]$, but at test time, we consider all $x \in [-3, 4]$. We train 20 neural networks with the same architecture. As shown in Fig. 3.4a, the ensemble disagrees most on $x < -1, x > 2$. This means that we should most mistrust predictions from this model class on these extreme values, since there are many models within the class that perform equally well on the training data, but differ greatly on those inputs. We should also mistrust predictions from $x \in [0, 1]$, although detecting this underspecification may be harder since the ensemble agrees more strongly on these points.

For each model in the ensemble, we test our method's performance by AUC on an OOD task: can we flag test points which fall outside the training distribution? We show that the underspecification score is empirically related to the standard deviation of the ensemble's predictions at the input, which in turn is related to whether the input is OOD (Fig. 3.4b). Examining one model from this ensemble, we observe that by estimating only $m = 2$ eigenvectors, we achieve $> 90\%$ AUC (Fig. 3.4c). It turns

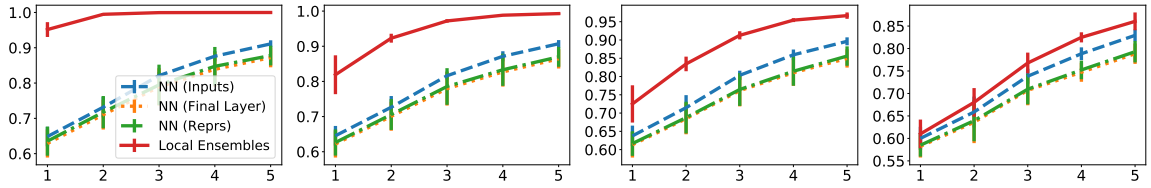


Figure 3.6: AUC achieved on WineQuality simulated features task (y-axis) compared to the number of extra features simulated (x-axis). Noise parameter σ increased from left to right $\sigma \in \{0, 0.1, 0.2, 0.5\}$. Solid line is our method. Results averaged across 5 random seeds, standard deviations shown. Results for the other three datasets are qualitatively similar and shown in Appendix B.3.

out that $m = 10$ performs best on this task/model. As we complete more iterations ($m > 10$) we start finding smaller eigenvalues, which are more important to the ensemble subspace and whose eigenvector we do not wish to project out. We note our AUC improves even with some eigenvector estimates having low cosine similarity to the ground truth eigenvectors (the Lanczos iteration has some stochasticity due to minibatch estimation — see Appendix B.2 for details). We hypothesize this robustness is because the ensemble subspace of this model class is relatively low-dimensional. Even if an estimated vector is noisy, the non-overlapping parts of the projection will likely be mostly perpendicular to the ensemble subspace, due to the properties of high-dimensional space.

3.6.2 Simulated Features

In this experiment, we create a blind spot in a given dataset by extending and manipulating the data's feature distribution. We induce a collinearity in feature space by generating new features which are a linear combination of two other randomly selected features in the training data. This means there is potential for underspecification: multiple, equally good learnable relationships exist between those features and the target. However, at test time, we will sometimes sample these simulated features from their *marginal* distribution instead. This breaks the linear dependence, requiring underspecification (the model is by definition underconstrained), without making the new data trivially out-of-distribution. We can make this underspecification detection task easier by generating several features this way, or make it harder by adding some noise $\sim \mathcal{N}(0, \sigma^2)$ to these features.

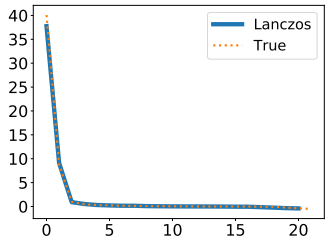


Figure 3.5: True and estimated eigenspectrums for toy model Hessian. We note that the first few eigenvalues account for most of the variation, and that our estimates are accurate.

We run this experiment on four tabular datasets. We compare to three nearest-neighbour baselines, where the metric is the distance (in some space) of the test point to its nearest neighbour by Euclidean distance in an in-distribution validation set. *NN (Inputs)* uses input space; *NN (Reprs)* uses hidden representation space, which is formed by concatenating all the activations of the network together (inspired by Papernot and McDaniel (2018), who propose a similar method for adversarial robustness); and *NN (Final Layer)*, uses just the final hidden layer of representations. We note that since our method is *post-hoc* and can be applied to any twice-differentiable pre-trained model, we do not compare to training-based methods e.g. those producing Bayesian predictive distributions (Blundell et al., 2015, Gal and Ghahramani, 2016). Our

metric is AUC: we aim to assign higher scores to inputs which break the collinearity (where the feature is drawn from the marginal), than those which do not. In Figure 3.6, we show that local ensembles (LE) outperform the baselines for each number of extra simulated features, and that this performance is fairly robust to added noise.

3.6.3 Correlated Latent Factors

Here, we extend the experiment from Section 3.6.2 by inducing a blind spot in *latent* space. The rationale is similar: if two latent factors are strongly correlated at training time, the model may grow reliant on that correlation, and at test-time may be underdetermined if that correlation is broken.

We use the CelebA dataset (Liu et al., 2015) of celebrity faces, which annotates every image with 40 latent binary attributes describing the image (e.g. "brown hair"). To induce the blind spot, we choose two attributes: a *label* L and a *spurious correlate* C. We then create a training set where L and C are perfectly correlated: a point is only included in the training set if $L = C$. We train a convolutional neural network (CNN) as a binary classifier to predict L. Then, we create a test set of held-out data where $P(L = C) = P(L \neq C)$. The test data where $L \neq C$ is in our model's blind spot; these are the inputs for which we want to output high underspecification scores. We show in Appendix B.5 that the models dramatically fail to classify these inputs ($L \neq C$). We compare to four baseline underspecification scores: the three nearest-neighbour methods described in Sec. 3.6.2, as well as *MaxProb*, where we use 1− the maximum outputted probability of the softmax. We test two values of L (*Male* and *Attractive*) and two values of C (*Eyeglasses* and *WearingHat*). We chose these specific values of L because they are difficult to predict and holistic i.e.and not localized to particular areas of image space. Note that these more holistic attributes make it a little more difficult to conceptualize what underspecification will "look like", and as such we should expect it to be more difficult to detect than in experiments above.

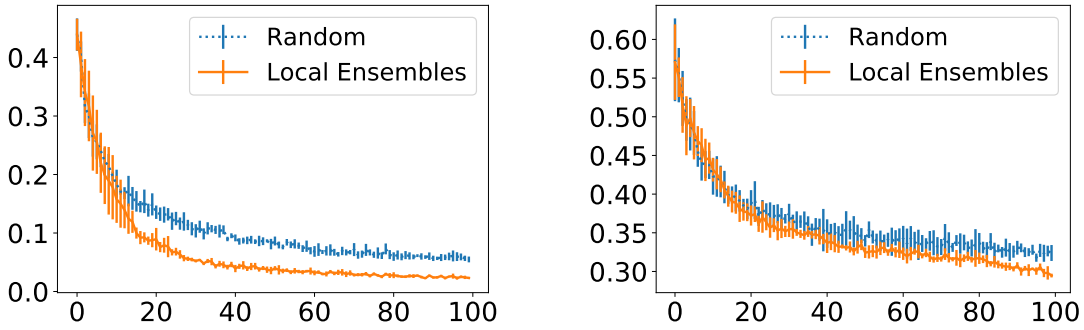
Method	**M/E**	**M/H**	**A/E**	**A/H**
MaxProb	0.738	**0.677**	0.461	0.433
NN (Pixels)	0.561	0.550	**0.521**	0.547
NN (Reprs)	0.584	0.578	**0.503**	0.533
NN (Final Layer)	0.589	0.517	0.480	0.497
LE (Loss)	**0.770**	**0.684**	0.454	0.456
LE (Predictions)	0.364	0.544	**0.519**	**0.582**

Table 3.2: AUC for Latent Factors OOD detection task. Column heading denotes in-distribution definitions: labels are M (Male) and A (Attractive); spurious correlates are E (Eyeglasses) and H (Wearing Hat). Image is in-distribution iff label = spurious correlate. LE stands for local ensembles. Each Lanczos iteration uses 3000 eigenvectors. 500 examples from each test set are used. 95% CI is bolded.

In Table 3.2, we present results for each of the four L, C settings, showing both the loss gradient and the prediction gradient variant of local ensembles. Note that the loss gradient cannot be calculated at test time since we do not have labels available — instead, we calculate a separate underspecification score using the gradient for the loss with respect to each possible label, and take the minimum. Our method achieves the best performance on most settings, and is competitive with the best baseline on each. However, the variation between the tasks is quite noteworthy. We note two patterns in particular. Firstly, we note that the performance of *MaxProb* and the loss gradient variant of our method are quite correlated, and we hypothesize this correlation is related to $\nabla_{\hat{Y}} \ell$. Additionally, we observe the effect of increasing m is inconsistent between experiments: we

discuss possible relationships to the eigenspectrum of the trained models. See Appendix B.5 for a discussion on these patterns — in particular, we note that one potential explanation for the variation in performance on these different tasks is how closely to a minimum we succeeded in training our model, suggesting that more care taken at training time could improve the results of the post-hoc method. This could affect performance if we are finding some large negative eigenvalues in the Lanczos iteration — while these would not be directions in which our local ensemble should lie, it could lead to a different interpretation in terms of how underspecification in the model corresponds to spurious correlations.

3.6.4 Active Learning



(a) MNIST (b) FashionMNIST

Figure 3.7: Active learning results. X-axis shows the number of rounds of active learning, Y-axis shows the error rate. Average of 5 random seeds shown with standard deviation error bars.

Finally, we consider the case where we know our model has blind spots, but do not know where they are. We use active learning to probe this situation, with the hypothesis that the most useful points to add to our training set may be among the most underspecified. We use MNIST (LeCun et al., 2010) and FashionMNIST (Xiao et al., 2017) for our active learning experiments. We begin the first round with a training set of twenty: two labelled data points from each of the ten classes. In each round, we train to a minimum validation loss using the current training set. After each round, we select ten new points from a randomly selected pool of 500 unlabelled points, and add those ten points and their labels to our training set. We compare local ensembles (selecting the points with the highest underspecification scores using the loss-gradient variant) to a random baseline selection mechanism. In Fig. 3.7, we show that our method outperforms the baseline on both datasets, and this improvement increases in later rounds of active learning. We only used 10 eigenvectors in our Lanczos approximation, which we found to be a surprisingly effective approximation; we did not observe improvement with more eigenvectors. This experiment serves to emphasize the flexibility of our method: by detecting an underlying property of the model, we are able to use the method for a range of tasks (active learning as well as OOD detection). We do not claim here that local ensembles are necessarily a state-of-the-art approach to active learning — although they do represent some promising conceptual insights which could result in practically helpful active learning approaches in future work.

3.7 Conclusion

We present *local ensembles*, a post-hoc method for detecting underspecification in a trained model. Our method uses local second-order information to approximate the variance of an ensemble. We describe how to tractably implement this method using the Lanczos iteration to estimate the largest eigenvectors of the Hessian, and demonstrate its practical flexibility and utility. Although this method is not a full replacement for ensemble methods, which can characterize more complexity in the loss landscape such as multiple modes, we believe it fills an important role in characterizing one component of poor prediction reliability. In future work, we hope to scale up these methods to larger models and datasets and to further explore the properties of different stopping points m. We also hope to explore applications in fairness and interpretability, where understanding model and training bias is of paramount importance.

One interesting note is the contrast between our paper's approach and work that suggests "flat minima" improve generalization. Works like Hochreiter and Schmidhuber (1997) suggest that finding flatter minima (i.e. smaller-eigenvalued Hessians) induces better in-distribution generalization to held-out test sets. This is seemingly in contrast to our point, that the flat directions in the loss landscape are those where underspecification can occur. However, on further examination, these two points are aligned: the flat minima literature suggests that flat minima are good for generalization on a held-out in-distribution test set since you can perturb your solution without increasing test loss; our work suggests that some of these perturbation directions may nonetheless increase loss on examples which are potentially atypical of this test set, and that this may be particularly misleading due to the construction of the training set.

Since the publication of this paper, underspecification has been pinpointed as an increasingly relevant and salient concept for the trustworth ML community, with empirical studies (D'Amour et al., 2020) and new, related concepts like Rashomon sets (Fisher et al., 2019) and predictive multiplicity (Marx et al., 2020). This hammers home the importance of methods such as the ones introduced in this paper, and principles such as ours for analyzing underspecification. Additionally, new work emerging analyzing the scalability of Lanczos iteration-like approaches (Schioppa et al., 2021) suggests some agreement in the community about their usefulness. However, the practical utility of these methods has not yet really been shown — modified approaches which improve interpretability (Barshan et al., 2020) highlight a path forward but also provide a reality check on the applicability of these approaches "out-of-the-box".

Chapter 4

Learning to Defer

4.1 Introduction

How can human and machine decision-makers work together? This is a question of crucial and broad importance, as many high-stakes applications of machine learning involve collaboration of predictive machines with human operators, rather than complete automation or replacement. Goldstein et al. (2017) describe the "Centaur Care" model as the ideal future of machine learning learning tools in healthcare: both humans and machines working jointly to make improved predictions and decisions. In a similar argument (but completely unrelated domain), McIlroy-Young et al. (2020) argue that there is "substantial promise" in collaborative systems, using chess playing as an example.

In this chapter, we expand on one specific component of human-machine collaboration, which Raghu et al. (2019a) call the "algorithmic automation" problem: on which examples does the machine predict and on which does the human predict? Of course, in practice this division is never quite so clean, as the human decision-maker very frequently has some type of control over the final outcome of the system. In fact, "meaningful human control" is often cited as a valuable property for a fully or partially automated high-stakes system to have[1]. In this chapter, we abstract away this complexity and consider a simple model for this collaboration: one where, at deployment time, the model is allowed to abstain from prediction on some examples, instead flagging them to be sent on to a downstream (probably human) external decision-maker. In particular, the examples we flag are ones where we expect the model to somehow be unreliable — Chapter 3 provides an example of such a method, using the fingerprint of underspecification in a trained model to derive a principled heuristic for which examples should be flagged.

Here, we zoom out on this approach and ask: what happens when an example is flagged? It is often sent downstream to some *external decision-maker*: be it a human user, committee of humans, a research lab for further inquiry, or some other, more complex algorithmic tool. How should we think about this piece of the pipeline in machine learning? What is its eventual effect on the impact of the system, both in terms of system accuracy and fairness?

These questions have broad conceptual importance. In general, model outputs are rarely system outputs; rather, they are often used as inputs to other decision-making processes. For example,

[1] What exactly it is that we desire out of "meaningful human control", and its pros and cons in implementation, can be complex - see McCoy et al. (2020) for a further discussion.

consider a black-box model which outputs risk scores to assist a decision-maker (e.g. presiding over a bail case (Hannah-Moffat, 2013)). How does a risk score factor into the decision-making process of an external agent such as a judge? How should this influence how the score is learned? The model producing the score may be state-of-the-art in isolation, but its true impact comes as an element of the judge's decision-making process. This can be seen as a special case of a mis-specified reward function (Amodei et al., 2016), where the objective that we are learning under (the ML model's loss) is not representative of the model's true impact (as a piece of the system).

In this chapter, we argue that when these models are used as part of larger systems e.g. in tandem with another decision maker, this adds a higher-level desideratum to our concerns. In these cases, a model's properties (e.g. fairness), should be considered in terms of its impact as part of the larger system, and the model should predict only if its predictions are reliably aligned with the system's objectives, which often include accuracy (predictions should mostly indicate ground truth) and fairness (predictions should be unbiased with respect to different subgroups).

Rejection learning (Chow, 1957, Cortes et al., 2016) examines this flagging problem, and when it is optimal to allow models to *reject* (not make a prediction) when they are not confidently accurate. However, this approach is inherently *nonadaptive*: both the model and the decision-maker act independently of one another. When a model is working in tandem with some external decision-maker, the decision to reject should depend not only on the model's confidence, but also on the decision-maker's expertise and weaknesses. For example, if the decision-makers's black-box is uncertain about some subgroup, but the decision-maker is very inaccurate or biased towards that subgroup, we may prefer the model make a prediction despite its uncertainty.

In this chapter, we formulate adaptive rejection learning, which we call *learning to defer*, where the model works *adaptively* with the decision-maker. We provide theoretical and experimental evidence that learning to defer improves upon the standard rejection learning paradigm, if models are intended to work as part of a larger system. We show that embedding a deferring model in a pipeline can improve the accuracy and fairness of the system as a whole. Experimentally, we simulate three scenarios where our model can defer judgment to external decision makers, echoing realistic situations where downstream decision makers are inconsistent, biased, or have access to side information. Our experimental results show that in each scenario, learning to defer allows models to work with users to make fairer, more responsible decisions.

4.2 Learning to Defer

4.2.1 A Joint Decision-Making Framework

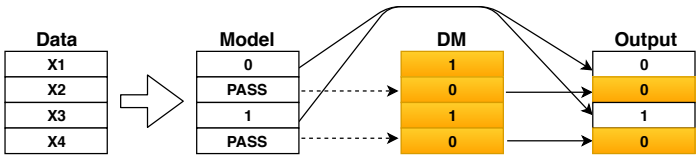


Figure 4.1: A larger decision system containing an automated model. When the model predicts, the system outputs the model's prediction; when the model says PASS, the system outputs the decision-maker's (DM's) prediction. Standard rejection learning considers the model stage, in isolation, as the system output, while learning-to-defer optimizes the model over the system output.

A complex real-world decision system can be modeled as an interactive process between various agents including decision makers, enforcement groups, and learning systems. Our focus in this paper is on a two-agent model, between one decision-maker and one learning model, where the decision flow is in two stages. This simple but still interactive setup describes many practical systems containing multiple decision-makers (Fig 4.1). The first stage is an automated model whose parameters we want to learn. The second stage is some external decision maker (DM) which we do not have control over e.g. a human user, a proprietary black-box model. The decision-making flow is modeled as a cascade, where the first-step model can either predict (positive/negative) or say PASS. If it predicts, the DM will output the model's prediction. However, if it says PASS, the DM makes its own decision. This scenario is one possible characterization of a realistic decision task, which can be an interactive (potentially game-theoretic) process[2].

We can consider the first stage to be flagging difficult cases for review, culling a large pool of inputs, auditing the DM for problematic output, or simply as an assistive tool. In our setup, we assume that the DM has access to information that the model does not — reflecting a number of practical scenarios where DMs later in the chain may have more resources for efficiency, security, or contextual reasons. However, the DM may be flawed, e.g. biased or inconsistent. A tradeoff suggests itself: can a machine learning model be combined with the DM to leverage the DM's extra insight, but overcome its potential flaws?

We can describe the problem of learning an automated model in this framework as follows. There exist data $X \in \mathbb{R}^n$, ground truth labels $Y \in \{0, 1\}$, and some auxiliary data $Z \in \mathbb{R}^m$ which is only available to the DM[3]. If we let $s \in \{0, 1\}$ be a PASS indicator variable ($s = 1$ means PASS), then the joint probability of the system in Fig. 4.1 can be expressed as follows:

$$\begin{aligned} P_{defer}(Y|X,Z) = \prod_i [P_M(Y_i = 1|X_i)^{Y_i}(1 - P_M(Y_i = 1|X_i))^{1-Y_i}]^{(1-s_i|X_i)} \\ [P_D(Y_i = 1|X_i, Z_i)^{Y_i}(1 - P_D(Y_i = 1|X_i, Z_i))^{1-Y_i}]^{(s_i|X_i)} \end{aligned} \tag{4.1}$$

where P_M is the probability assigned by the automated model, P_D is the probability assigned by the DM, and i indexes examples. This can be seen as a mixture of Bernoullis, where the labels are generated by either the model or the DM as determined by s. For convenience, we compress the probabilistic notation:

$$\begin{aligned} \hat{Y}_M = f(X) = P_M(Y = 1|X) \in [0, 1]; \qquad & \hat{Y}_D = h(X, Z) = P_D(Y = 1|X, Z) \in [0, 1] \\ \hat{Y} = (1 - s)\hat{Y}_M + s\hat{Y}_D \in [0, 1]; \qquad & s = g(X) \in \{0, 1\} \end{aligned} \tag{4.2}$$

$\hat{Y}_M, \hat{Y}_D, \hat{Y}$ are model predictions, DM predictions, and system predictions, respectively (left to right in Fig. 4.1). The DM function h is a fixed, unknown black-box. Therefore, learning good $\{\hat{Y}_M, s\}$ involves learning functions f and g which can adapt to h – the goal is to make $\hat{Y}$ a good predictor of Y. To do so, we want to find the maximum likelihood solution of Eq. 4.1. We can minimize its

[2]We will not discuss the game-theoretic aspect in this thesis; see the literature on *strategic classification* (Dong et al., 2018, Hardt et al., 2016a, Milli et al., 2019)

[3]We include this auxiliary data in our setup to formalize the notion that the external decision-maker has some sort of basis on which to make decisions which differ from the model. Note we do not guarantee that the auxiliary data is useful for prediction, so any difference in prediction from added information is not necessarily for the better.

negative log-likelihood $\mathcal{L}_{defer}$, which can be written as:

$$\mathcal{L}_{defer}(Y, \hat{Y}_M, \hat{Y}_D, s) = -\log P_{defer}(Y|X, Z) = -\sum_i [(1 - s_i)\ell(Y_i, \hat{Y}_{M,i}) + s_i \ell(Y_i, \hat{Y}_{D,i})] \tag{4.3}$$

where $\ell(Y, p) = Y \log p + (1 - Y) \log (1 - p)$ i.e. the log probability of the label with respect to some prediction p. We can optionally also include a regularization penalty on the value of s_i using a coefficient γ_{defer} to encourage higher rates of automation. Minimizing $\mathcal{L}_{defer}$ is what we call ***learning to defer***. In learning to defer, we aim to learn a model which outputs predictive probabilities $\hat{Y}_M$ and binary deferral decisions s, in order to optimize the output of the *system as a whole*. The role of s is key here: rather than just an expression of uncertainty, we can think of it as a gating variable, which tries to predict whether $\hat{Y}_M$ or $\hat{Y}_D$ will have lower loss on any given example. This leads naturally to a mixture-of-experts learning setup (Jacobs et al., 1991); however, we are only able to optimize the parameters for one expert ($\hat{Y}_M$), whereas the other expert ($\hat{Y}_D$) is out of our control. We discuss further in Sec. 4.3.

We now examine the relationship between learning to defer and rejection learning. Specifically, we will show that learning to defer is a generalization of rejection learning and argue why it is an important improvement over rejection learning for many machine learning applications.

4.2.2 Learning to Reject

Rejection learning is the predominant paradigm for learning models with a PASS option (see Sec. 4.4). In this area, the standard method is to optimize the accuracy-rejection tradeoff: how much can a model improve its accuracy on the cases it *does* classify by PASS-ing some cases? This is usually learned by minimizing a classification objective $\mathcal{L}_{reject}$ with a penalty γ_{reject} for each rejection (Cortes et al., 2016), where Y is ground truth, $\hat{Y}_M$ is the model output, and s is the reject variable ($s = 1$ means PASS); all binary:

$$\mathcal{L}_{reject}(Y, \hat{Y}_M, s) = -\sum_i [(1 - s_i)\ell(Y_i, \hat{Y}_{M,i}) + s_i \gamma_{reject}] \tag{4.4}$$

where ℓ is usually classification accuracy, i.e. $\ell(Y_i, \hat{Y}_i) = \mathbb{1}[Y_i = \hat{Y}_i]$. If we instead consider ℓ to be the log-probability of the label (i.e. the cross-entropy loss of the label given the model's prediction), then we can interpret $\mathcal{L}_{reject}$ probabilistically as the negative log-likelihood of the joint distribution P_{reject}:

$$P_{reject}(Y|X) = \prod_i [\hat{Y}_{M,i}^{Y_i}(1 - \hat{Y}_{M,i})^{(1-Y_i)}]^{1-s_i} \exp(\gamma_{reject})^{s_i} \tag{4.5}$$

4.2.3 Learning to Defer is Adaptive Rejection Learning

In learning to defer, the model leverages information about the DM to make PASS decisions adaptively. We can consider how learning to defer relates to rejection learning. Examining their loss functions Eq. 4.3 and Eq. 4.4 respectively, the only difference is that the rejection loss has a constant γ_{reject} where the deferring loss has variable $\ell(Y, \hat{Y}_D)$.

Let $\ell(Y, \hat{Y})$ be our desired example-wise objective, where $Y = \arg\min_{\hat{Y}} -\ell(Y, \hat{Y})$. Then, we can show that if the DM has constant loss (e.g. is an oracle), there exist values of $\gamma_{reject}, \gamma_{defer}$ for which the learning-to-defer and rejection learning objectives are equivalent.

As in Eq. 4.4, the standard rejection learning objective is

$$\mathcal{L}_{reject}(Y, \hat{Y}_M, \hat{Y}_D, s) = -\sum_i [(1 - s_i)\ell(Y_i, \hat{Y}_{M,i}) + s_i \gamma_{reject}] \tag{4.6}$$

where the first term encourages a low negative loss ℓ for non-PASS examples and the second term penalizes PASS at a constant rate, γ_{reject}. If we include a similar γ_{defer} penalty, the deferring loss function is

$$\mathcal{L}_{defer}(Y, \hat{Y}_M, \hat{Y}_D, s) = -\sum_i [(1 - s_i)\ell(Y_i, \hat{Y}_{M,i}) + s_i \ell(Y_i, \hat{Y}_{D,i}) + s_i \gamma_{defer}] \tag{4.7}$$

Now, if the DM has constant loss, meaning $\ell(Y, \hat{Y}_D) = \alpha$, we have (with $\gamma_{defer} = \gamma_{reject} - \alpha$):

$$\begin{aligned} \mathcal{L}_{defer}(Y, \hat{Y}_M, \hat{Y}_D, s) &= -\sum_i [(1 - s_i)\ell(Y_i, \hat{Y}_{M,i}) + s_i \cdot \alpha + s_i \gamma_{defer}] \\ &= -\sum_i [(1 - s_i)\ell(Y_i, \hat{Y}_{M,i}) + s_i(\gamma_{defer} + \alpha)] = \mathcal{L}_{reject}(Y, \hat{Y}_{M,i}, s) \end{aligned} \tag{4.8}$$

4.2.4 Why Learn to Defer?

The proof in Sec. 4.2.3 shows the central point of learning to defer: rejection learning is exactly a special case of learning to defer: a DM with constant loss α on each example. We argue that the adaptive version (learning to defer), more accurately describes real-world decision-making processes. Often, a PASS is not the end of a decision-making sequence. Rather, a decision must be made eventually on every example by a DM, whether the automated model predicts or not, and the DM will not, in general, have constant loss on each example.

Say our model is trained to detect melanoma, and when it says PASS, a human doctor can run an extra suite of medical tests. The model learns that it is very inaccurate at detecting amelanocytic (non-pigmented) melanoma, and says PASS if this might be the case. However, suppose that the doctor is even *less* accurate at detecting amelanocytic melanoma than the model is. Then, we may prefer the model to make a prediction despite its uncertainty. Conversely, if there are some illnesses that the doctor knows well, then the doctor may have a more informed, nuanced opinion than the model. Then, we may prefer the model say PASS more frequently relative to its internal uncertainty.

Saying PASS on the wrong examples can also have fairness consequences. If the doctor's decisions bias against a certain group, then it is probably preferable for a less-biased model to defer less frequently on the cases of that group. If some side information helps a DM achieve high accuracy on some subgroup, but confuses the DM on another, then the model should defer most frequently on the DM's high accuracy subgroup, to ensure fair and equal treatment is provided to all groups. In short, if the model we train is part of a larger pipeline, then we should train and evaluate the performance of *the pipeline with this model included*, rather than solely focusing on the model itself. We note that it is unnecessary to acquire decision data from a specific DM; rather, data could be sampled from many DMs (potentially using crowd-sourcing). Research suggests that common trends exist in DM behavior (Busenitz and Barney, 1997, Danziger et al., 2011), suggesting that a model trained on some DM could generalize to unseen DMs.

4.3 Formulating Adaptive Models within Decision Systems

In our decision-making pipeline, we aim to formulate a fair model which can be used for learning to defer (Eq. 4.3) (and by extension non-adaptive rejection learning as well (Eq. 4.4)). Such a model must have two outputs for each example: a predictive probability $\hat{Y}_M$ and a PASS indicator s. We draw inspiration from the mixture-of-experts model (Jacobs et al., 1991). One important difference between learning-to-defer and a mixture-of-experts is that one of the "experts" in this case is the DM, which is out of our control; we can only learn the parameters of $\hat{Y}_M$.

If we interpret the full system as a mixture between the model's prediction $\hat{Y}_M$ and the DM's predictions $\hat{Y}_D$, we can introduce a mixing coefficient π, where $s \sim Ber(\pi)$. π is the probability of deferral, i.e. that the DM makes the final decision on an example X, rather than the model; $1-\pi$ is the probability that the model's decision becomes the final output of the system. Recall that $\hat{Y}_M, \pi$ are functions of the input X; they are parametrized below by θ. Then, if there is some loss $\ell(Y, \hat{Y})$ we want our system to minimize, we can learn to defer by minimizing an expectation over s:

$$\begin{aligned}\mathcal{L}_{defer}(Y, \hat{Y}_M, \hat{Y}_D, \pi; \theta) &= \mathbb{E}_{s\sim Ber(\pi)}\mathcal{L}(Y, \hat{Y}_M, \hat{Y}_D, s; \theta) \\ &= \sum_i \mathbb{E}_{s_i \sim Ber(\pi_i)}[(1-s_i)\ell(Y_i, \hat{Y}_{M,i}; \theta) + s_i \ell(Y_i, \hat{Y}_{D,i})]\end{aligned} \tag{4.9}$$

or, in the case of rejection learning:

$$\mathcal{L}_{reject}(Y, \hat{Y}_M, \hat{Y}_D, \pi; \theta) = \sum_i \mathbb{E}_{s_i \sim Ber(\pi_i)}[(1-s_i)\ell(Y_i, \hat{Y}_{M,i}; \theta) + s_i \gamma_{reject}] \tag{4.10}$$

Next, we give two methods of specifying and training such a model and present our method of learning these models fairly, using a regularization scheme.

4.3.1 Post-hoc Thresholding

One way to formulate an adaptive model with a PASS option is to let π be a function of $\hat{Y}_M$ alone; i.e. $\hat{Y}_M = f(X)$ and $\pi = g(\hat{Y}_M)$. One such function g is a thresholding function — we can learn two thresholds t_0, t_1 (see Figure 4.2) which yield a ternary classifier. The third category is PASS, which can be outputted when the model prefers not to commit to a positive or negative prediction. A convenience of this method is that the thresholds can be trained post-hoc on an existing model with an output in $[0, 1]$ e.g. many binary classifiers. We use a neural network as our binary classifier, and describe our post-hoc thresholding scheme in Appendix C.4. At test time, we use the thresholds to partition the examples. On each example, the model outputs a score $\beta \in [0, 1]$. If $t_0 < \beta < t_1$, then we output $\pi = 1$ and defer (the value of $\hat{Y}_M$ becomes irrelevant). Otherwise, if $t_0 \geq \beta$ we output $\pi = 0, \hat{Y}_M = 0$; if $t_1 \leq \beta$ we output $\pi = 0, \hat{Y}_M = 1$. Since $\pi \in \{0, 1\}$ here, the expectation over $s \sim Ber(\pi)$ in Eq. 4.9 is trivial.

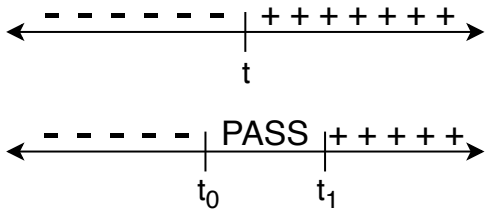


Figure 4.2: Binary classification (one threshold) vs. ternary classification with a PASS option (two thresholds)

.

4.3.2 Learning a Differentiable Model

We may wish to use continuous outputs $\hat{Y}_M, \pi \in [0, 1]$ and train our models with gradient-based optimization. To this end, we consider a method for training a differentiable adaptive model. One could imagine extending the method in Sec. 4.3.1 to learn smooth thresholds end-to-end on top of a predictor. However, to add flexibility, we can allow π to be a function of X as well as $\hat{Y}_M$, i.e. $\hat{Y}_M = f(X)$ and $\pi = g(\hat{Y}_M, X)$. This is advantageous because a DM's actions may depend heterogenously on the data: the DM's expected loss may change as a function of X, and it may do so differently than the model's. We can parametrize $\hat{Y}_M$ and π with neural networks, and optimize Eq. 4.9 or 4.10 directly using gradient descent. At test time, we defer when $\pi > 0.5$.

We estimate the expected value in Eq. 4.9 by sampling $s \sim Ber(\pi)$ during training (once per example). To estimate the gradient through this sampling procedure, we use the Concrete relaxation (Jang et al., 2016, Maddison et al., 2017). Additionally, it can be helpful to stop the gradient from π from backpropogating through $\hat{Y}_M$. This allows for $\hat{Y}_M$ to still be a good predictor independently of π.

4.3.3 Fair Classification through Regularization

We can build a regularized fair loss function to combine error rate with a fairness metric. We can extend the loss in Eq. 4.9 to include a regularization term $\mathcal{R}$, with a coefficient α_{fair} to balance accuracy and fairness:

$$\mathcal{L}_{defer}(Y, \hat{Y}_M, \hat{Y}_D, \pi; \theta) = \mathbb{E}_{s\sim Ber(\pi)}[\mathcal{L}(Y, \hat{Y}_M, \hat{Y}_D, s; \theta) + \alpha_{fair}\mathcal{R}(Y, \hat{Y}_M, \hat{Y}_D, s) \tag{4.11}$$

We now recall some background on fair classification, introduced in Section 2.1.1. In fair binary classification, we have input labels Y, predictions $\hat{Y}$, and sensitive attribute A (e.g., gender, race, age, etc.), assuming for simplicity that $Y, \hat{Y}, A \in \{0, 1\}$. In this work we assume that A is known. The aim is twofold: firstly, *accurate* classification i.e $Y_i = \hat{Y}_i$; and *fairness with respect to* A i.e. $\hat{Y}$ does not discriminate unfairly against particular values of A. Adding fairness constraints can provably hurt classification error (Menon and Williamson, 2018). We thus define a loss function which trades off between these two objectives, yielding a regularizer. We choose equalized odds as our fairness metric (Hardt et al., 2016b), which requires that false positive and false negative rates are equal between the two groups. We will refer to the difference between these rates as disparate impact (DI). Here we define a continuous relaxation of DI, having the model output a probability p and considering $\hat{Y} \sim Ber(p)$. The resulting term $\mathscr{D}$ acts as our regularizer $\mathcal{R}$ in Eq. 4.11:

$$\begin{aligned} DI_{Y=i}(Y, A, \hat{Y}) &= |\mathbb{E}_{\hat{Y}\sim Ber(p)}(\hat{Y} = 1 - Y | A = 0, Y = i) - \mathbb{E}_{\hat{Y}\sim Ber(p)}(\hat{Y} = 1 - Y | A = 1, Y = i)| \\ \mathscr{D}(Y, A, \hat{Y}) &= \frac{1}{2}(DI_{Y=0}(Y, A, \hat{Y}) + DI_{Y=1}(Y, A, \hat{Y})) \end{aligned} \tag{4.12}$$

Our regularization scheme is similar to Bechavod and Ligett (2017), Kamishima et al. (2012); see Appendix C.2 for results confirming the efficacy of this scheme in binary classification. We show experimentally that the equivalence between learning to defer with an oracle-DM and rejection learning holds in the fairness case (see Appendix C.5).

4.4 Related Work

Notions of Fairness. A challenging aspect of machine learning approaches to fairness is formulating an operational definition. Several works have focused on the goal of treating similar people similarly (individual fairness) and the necessity of fair-awareness (Dwork et al., 2012, Zemel et al., 2013).

Some definitions of fairness center around statistical parity (Kamiran and Calders, 2009, Kamishima et al., 2012), calibration (Guo et al., 2017, Pleiss et al., 2017) or equalized odds (Chouldechova, 2017, Hardt et al., 2016b, Kleinberg et al., 2017, Zafar et al., 2017a). It has been shown that equalized odds and calibration cannot be simultaneously (non-trivially) satisfied (Chouldechova, 2017, Kleinberg et al., 2017). Hardt et al. (2016b) present the related notion of "equal opportunity". Zafar et al. (2017a) and Bechavod and Ligett (2017) develop and implement learning algorithms that integrate equalized odds into learning via regularization. Other approaches to fair classification provide various theoretical guarantees (Agarwal et al., 2018, Donini et al., 2018, Woodworth et al., 2017).

Incorporating PASS. Several works have examined the PASS option (cf. *rejection learning*), beginning with Chow (1957, 1970) who studies the tradeoff between error-rate and rejection rate. Cortes et al. (2016) develop a framework for integrating PASS directly into learning, which Ramaswamy et al. (2018) build on for the multi-class setting, and Geifman and El-Yaniv (2019) explore learning differentiably end-to-end. Attenberg et al. (2011) discuss the difficulty of a model learning what it doesn't know (particularly rare cases), and analyzes how human users can audit such models. Wang et al. (2017) propose a cascading model, which can be learned end-to-end and study the efficiency of this model. However, none of these works look at the fairness impact of this procedure. From the AI safety literature, Hadfield-Menell et al. (2016) give a reinforcement-learning algorithm for machines to work with humans to achieve common goals. We also note that the phrase "adaptive rejection" exists independently of this work, but with a different meaning (Fischer and Villmann, 2016).

A few papers have addressed topics related to both above sections. Bower et al. (2017) describe fair sequential decision making but do not have a PASS concept or provide a learning procedure. In Joseph et al. (2016), the authors show theoretical connections between KWIK-learning and a proposed method for fair bandit learning. Grgić-Hlaca et al. (2017) discuss fairness that can arise out of a mixture of classifiers. Varshney and Alemzadeh (2017) propose "safety reserves" and "safe fail" options which combine learning with rejection and fairness/safety, but do not analyze learning procedures or larger decision-making frameworks.

4.5 Experiments

We experiment with three scenarios, each of which represent an important class of real-world decision-makers:

1. **High-accuracy DM**, ignores fairness: This may occur if the extra information available to the DM is important, yet withheld from the model for privacy or computational reasons.

2. **Highly-biased DM**, strongly unfair: Particularly in high-stakes/sensitive scenarios, DMs can exhibit many biases.

3. **Inconsistent DM**, ignores fairness (DM's accuracy varies by subgroup, with total accuracy lower than model): Human DMs can be less accurate, despite having extra information (Dawes et al., 1989). We add noise to the DM's output on some subgroups to simulate human inconsistency.

Due to difficulty obtaining and evaluating real-life decision-making data, we use "semi-synthetic data": real datasets, and simulated DM data by training a separate classifier under slightly different conditions (see Experiment Details). In each scenario, the simulated DM receives access to extra information which the model does not see.

Datasets and Experiment Details. We use two datasets: COMPAS[4] (Kirchner and Larson, 2016), where we predict a defendant's recidivism without discriminating by race, and Heritage Health (https://www.kaggle.com/c/hhp), where we predict a patient's Charlson Index (a comorbidity indicator) without discriminating by age. We train all models and DMs with a fully-connected two-layer neural network. See Appendix C.3 for details on datasets and experiments.

We found post-hoc and end-to-end models performed qualitatively similarly for high-accuracy and highly-biased DMs, so we show results from the simpler model (post-hoc) for those. However, the post-hoc model cannot adjust to the case of the inconsistent DM (scenario 3), since it does not take X as an input to π (as discussed in Sec. 4.3.2), so we show results from the end-to-end model for the inconsistent DM.

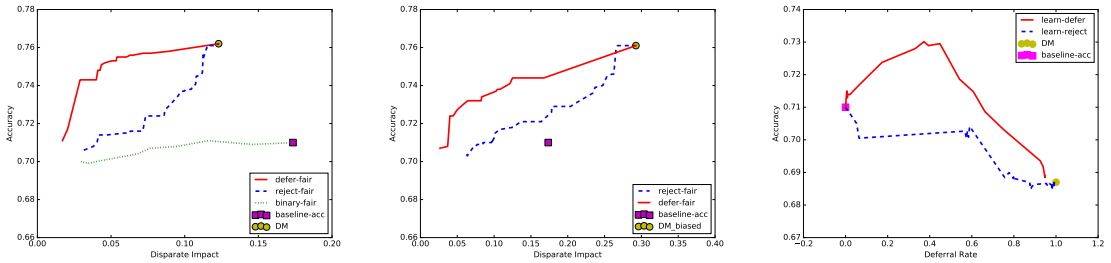
Each DM receives extra information in training. For COMPAS, this is the defendant's violent recidivism; for Health, this is the patient's primary condition group. To simulate high-bias DMs (scenario 2) we train a regularized model with $\alpha_{fair} = -0.1$ to encourage learning a "DM" with *high* disparate impact. To create inconsistent DMs (scenario 3), we flip a subset of the DM's predictions post-hoc with 30% probability: on COMPAS, this subset is people below the mean age; on Health this is males.

Displaying Results. We show results across various hyperparameter settings (α_{fair}, $\gamma_{defer}/\gamma_{reject}$), to illustrate accuracy and/or fairness tradeoffs. Each plotted line connects several points, which are each a median of 5 runs at one setting. In Fig. 4.3, we only show the Pareto front, i.e., points for which no other point had both better accuracy and fairness. All results are on held-out test sets.

4.5.1 Learning to Defer to Three Types of DM

High-Accuracy DM. In this experiment, we consider the scenario where a DM has higher accuracy than the model we train, due to the DM having access to extra information/resources for security, efficiency, or contextual reasons. However, the DM is not trained to be fair. In Fig. 4.3a, we show that learning-to-defer achieves a better accuracy-fairness tradeoff than rejection learning. Hence, learning-to-defer can be a valuable fairness tool for anyone who designs or oversees a many-part system - an adaptive first stage can improve the fairness of a more accurate DM. The fair rejection

[4]In later work, (Bao et al., 2021) highlight the flaws in the COMPAS dataset, stressing that it provides a non-representative and potentially misleading picture of the pre-trial risk assessment problem, and recommend against its usage as a benchmark in this setting. For these reasons, if the experiments here had been run at the time of publication of this thesis, we would have elected not to use COMPAS as a benchmark dataset. However, as the experiments in this section were run in 2017-18, we stress that these results do not provide direct evidence for the suitability of this or related approaches in a criminal justice context.



(a) COMPAS, High-Accuracy DM (b) COMPAS, Highly-Biased DM (c) COMPAS, Inconsistent DM

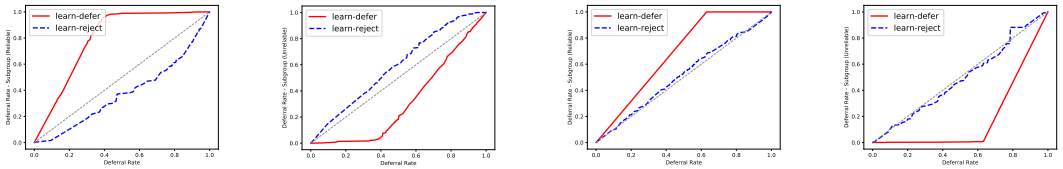
Figure 4.3: Comparing learning-to-defer, rejection learning and binary models. COMPAS dataset only; Health dataset results in Appendix C.1. Each figure is a different DM scenario. In Figs. 4.3a and 4.3b, X-axis is fairness (lower is better); in Fig. 4.3c, X-axis is deferral rate. Y-axis is accuracy for all figures. Square is a baseline binary classifier, trained only to optimize accuracy; dashed line is fair rejection model; solid line is fair deferring model. Yellow circle is DM alone. In Fig. 4.3a, green dotted line is a binary model also optimizing fairness. Figs. 4.3a and 4.3b are hyperparameter sweep over $\gamma_{reject/defer}/\alpha_{fair}$; Fig. 4.3c sweeps $\gamma_{reject/defer}$ only, with $\alpha_{fair} = 0$ (for $\alpha_{fair} \geq 0$, see Appendix C.6).

learning model also outperforms binary baselines, by integrating the extra DM accuracy on some examples.

Highly-Biased DM. In this scenario, we consider the case of a DM which is extremely biased (Fig. 4.3b). We find that the advantage of a deferring model holds in this case, as it adapts to the DM's extreme bias. For further analysis, we examine the deferral rate of each model in this plot (see Appendix C.7). We find that the deferring model's adaptivity brings two advantages: it can adaptively defer at different rates for the two sensitive groups to counteract the DM's bias; and it is able to modulate the overall amount that it defers when the DM is biased.

Inconsistent DM. In this experiment, we consider a DM with access to extra information, but which due to inconsistent accuracy across subgroups, has a lower overall accuracy than the model. In Fig. 4.3c, we compare deferring and rejecting models, examining their classification accuracy at different deferral rates. We observe that for each deferral rate, the model that learned to defer achieves a higher classification accuracy. Furthermore, we find that the best learning-to-defer models outperform both the DM and a baseline binary classifier. Note that although the DM is less accurate than the model, the most accurate result is not to replace the DM, but to use a DM-model mixture. Critically, only when the model is adaptive (i.e. learns to defer) is the potential of this mixture unlocked. In the accuracy/deferral rate plot, we note the relative monotonicity of the rejecting model's curve, as compared to the deferring model's. This visually captures the advantage of learning to defer; in the rejection model, PASSing on examples tends to hurt the system accuracy, whether it is done 10% of the time, or 90% of the time. However, in the deferring model the first ∼40% of PASSed examples actually improve the system's accuracy, demonstrating how learning to defer allows us to find the subset of examples on which deferral can be useful.

To analyze further how the deferring model in Fig. 4.3c achieves its accuracy, we examine two subgroups from the data: where the DM is reliable and unreliable (the unreliable subgroup is where

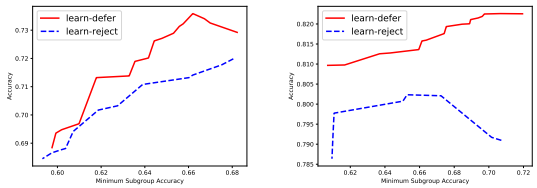


(a) COMPAS, Reliable (b) COMPAS, Unreliable (c) Health, Reliable (d) Health, Unreliable

Figure 4.4: Each point is some setting of $\gamma_{reject}/\gamma_{defer}$. X-axis is total deferral rate, Y-axis is deferral rate on DM reliable/unreliable subgroup (COMPAS: Old/Young; Health: Female/Male). Gray line = 45°: above is more deferral; below is less. Solid line: learning to defer; dashed line: rejection learning.

post-hoc noise was added to the DM's output; see Experiment Details). Fig. 4.4 plots the deferral rate on these subgroups against the overall deferral rates. We find that the deferring models deferred more on the DM's reliable subgroup, and less on the unreliable subgroup, particularly as compared to rejection models. This shows the advantage of learning to defer; the model was able to adapt to the strengths and weaknesses of the DM.

We also explore how learning-to-defer's errors distribute across subgroups. We look at accuracy on four subgroups, defined by the cross-product of the sensitive attribute and the attribute defining the DM's unreliability, both binary. In Fig. 4.5, we plot the minimum subgroup accuracy (MSA) and the overall accuracy. We find that the deferring models (which were higher accuracy in general), continue to achieve higher accuracy even when requiring that models attain a certain MSA. This means that the improvement we see in the deferring models are not coming at the expense of the least accurate subgroups. Instead, we find that the most accurate deferring models also have the highest MSA, rather than exhibiting a tradeoff. This is a compelling natural fairness property of learning to defer which we leave to future work for further investigation.



(a) COMPAS dataset (b) Health dataset

Figure 4.5: Each point is a single run in sweep over $\gamma_{reject}/\gamma_{defer}$. X-axis is the model's lowest accuracy over 4 subgroups, defined by the cross product of binarized (sensitive attribute, unreliable attribute), which are (race, age) and (age, gender) for COMPAS and Health respectively. Y-axis is model accuracy. Only best Y-value for each X-value shown. Solid line is learning to defer; dashed line is rejection learning.

4.6 Conclusion

In this work, we define a framework for multi-agent decision-making which describes many practical systems. We propose a method, learning to defer (or adaptive rejection learning), which generalizes rejection learning under this framework. We give an algorithm for learning to defer in the context of larger systems and explain how to do so fairly. Experimentally, we demonstrate that deferring models can optimize the performance of decision-making pipelines as a whole, beyond the improvement provided by rejection learning. This is a powerful, general framework, with ramifications for many

complex domains where automated models interact with other decision-making agents. Through deferring, we show how models can learn to predict responsibly within their surrounding systems, an essential step towards fairer, more responsible machine learning.

Since the publication of this research, there have been a number of advancements on its framework. Notably, Mozannar and Sontag (2020) provide a consistent estimator for classification error in the learning to defer setting. Wilder et al. (2021) provide a different framework for jointly optimizing two separate models — a prediction model and a when-to-query-the-DM model. Bansal et al. (2021) explore maximizing a human-machine team's joint expected utility without necessarily using the rejection learning frame. Keswani et al. (2021) consider the case of learning to defer when there are multiple experts in the picture. Verma and Nalisnick (2022) explore the question of calibration, showing how to ensure that the final model is calibrated when the external decision-maker's accuracy is also taken into consideration. Another line of work uses a very similar framework to ours under the name "human assistance", with a focus on theoretical guarantees (De et al., 2020, 2021, Okati et al., 2021). Focusing on the medical setting, Raghu et al. (2019b) consider the problem of directly predicting which examples should receive a second opinion. Finally, Meresht et al. (2020) examine a control model related to ours for "learning to switch" between various agents over time in an MDP.

Upon revisiting this paper's content, a conceptual tension between learning to defer and interpretability highlights itself. While learning to defer can achieve improved metrics across the board by learning a model which is adaptive to the human's specific needs, this could have troubling ramifications. It could mean that the human does not really know how to interpret the model, since the output may mean different things in different contexts. Additionally, the learning to defer objective can create an incentive for a model output to be based not on predictive utility, but on what action it may induce from the human downstream — it could be seen as encouraging the model to "lie" to the human decision-maker, rather than make its "honest best guess". The number of phrases in the previous sentence which necessitate scare-quotes highlights a gap in the literature around this topic — when is it advisable to learn to defer, and when is it not? In what situations would we consider it safe and ethical for a model to adapt its output to the characteristics of a downstream decision-maker? The ways in which we include external factors in a model's predictive scope is up for debate. Selbst et al. (2019) highlight "breaking the framing trap", as a potential framework for limiting the harms of ML models, but this paper's exploration and ensuing discussion highlight both the promises and pitfalls of this approach.

Chapter 5

Conclusion

5.1 Overview: System Thinking for Trustworthiness

We tie the various threads of this thesis together with a new concept: the importance of system-wide thinking in attacking problems for trustworthiness. The mitigations proposed throughout this thesis carefully consider what the role of a trained ML model is within a larger system, and how the inputs and outputs of that model are interpreted in the ML pipeline. Recall that one of the characteristics of wicked problems (Rittel and Webber, 1973) is that the solution space is never fully-defined — system-wide thinking allows us to explore new aspects of that solution space, and propose new ways of improving trustworthiness outside by leveraging a number of different machine learning frameworks. This type of thinking is helpful since it helps us to consider the whole range of vulnerabilities which exist when trying to build a trustworthy model — because users interact with a full system, every piece of the system is relevant to how trust is built. It has been widely explored in the fairness literature how different types of bias can enter a pipeline at different stages (Schelter and Stoyanovich, 2020, Suresh and Guttag, 2019). On the flip side, it can help us identify a number of different bottlenecks where we can propose mitigations to intervene to improve trustworthiness. For instance, we discuss a method for data pre-processing (representation learning) before training happens, and how this can guarantee certain properties of the predictive behaviour of the trained model; and an approach for flagging troublesome examples post-hoc (after training occurs), enforcing a system to abstain on examples for which is cannot make a reliable prediction.

This theme arises in other aspects of the author's research, which can be seen as laid out across the ML pipeline:

- **Understanding historical bias in data collection using a causal lens (Madras et al., 2019b)**: in this work, we re-frame "fair classification" as a causal intervention problem. Our end goal is to create a policy — a rule which tells us which *action* to take where (e.g. give a loan or not). Kleinberg et al. (2015) show how this causal question can be answered as a classification question; however, this does not mean that the training procedure should just be vanilla supervised learning. In fact, supervised learning is often wrong, since we are learning from *biased data*: data which was produced from some past policy, which may have had its own biases. In this case, our paper walks through a causal approach to learning from this biased data, and highlights some useful causal effects that we can talk about in this setting.

- **Identifying challenges in data labelling for normative contexts (Balagopalan et al., 2021)**: In certain high-stakes scenarios, particularly in questions of law, we are concerned with *normative* questions i.e. questions of what *should* be done. When considering applying ML to these cases, it can be tempting to think of them in terms of the facts underlying the relevant norms. For instance, a judge's decision about whether to grant an defendant bail could be thought of as, in its idealized state, simply a prediction about an defendant's probability of re-offending before trial. In fact, current risk scoring systems, which inform judge's decisions, are trained to do exactly this prediction problem (Brennan et al., 2009). However, we show that substituting a judgment of "how facts should be described" for a judgment of "if norms were violated" induces a significant measurement error: normative and descriptive judgement are not the same. We demonstrate this empirically using human labellers and show large gaps between these judgments. This has important ramifications for the way we collect data in normative scenarios (e.g. content moderation).

- **Developing approaches for pre-training representations which guarantee alignment with fairness metrics by downstream classifiers (Madras et al., 2018a)**: See Chapter 2 for a discussion.

- **Creating a framework for designing model evaluation protocols with a focus on out-of-context data (Madras and Zemel, 2021)**: ML models often fail at making predictions out-of-context (OOC), i.e. where examples are unusual or from uncommon context or subgroups from the training distribution. We consider the thorny problem of evaluating model performance on OOC examples, noting that often this depends on some notion of context defined by auxiliary data, and introducing a framework which unifies a growing literature on this topic. We present the NOOCh benchmark, a suite of "challenge sets" that consist of examples that arise naturally within a larger benchmark. Experimentally, we stress how different choices made in benchmark design can have varying effects on the conclusions we draw about the OOC capabilities of different models.

- **Building post-hoc schemes for improving model and system performance on unusual, unreliable, or unrepresented data (Madras et al., 2019a)**: See Chapter 3 for a discussion.

- **Integrating machine learning models into larger systems, containing multiple decision-making agents (Madras et al., 2018b)**: See Chapter 4 for a discussion.

Examining this body of work, we can see a consideration of how machine learning models fit into various places in the development pipeline, and how harms can arise or be mitigated at each of these steps.

5.2 Reflections and Limitations

In conclusion, we reflect on the breadth of the challenge ahead for the wicked problem of building trustworthy machine learning systems. First, we note that components of trust are much broader than those focused on in this thesis. For instance, techniques from explainability and interpretability are essential for building models whose behaviour can be anticipated by a user. For systems where user data is at play, privacy is of utmost concern so that a user is willing to engage at all with the

model, without fearing data leaks. The security of a model is paramount to trust, ensuring that no malicious actor can affect a system's output or modify its effects otherwise. Finally, notions of recourse are essential: a user's knowledge that they can affect a harmful prediction or action to be overturned if it is outside accepted behaviour is a key component to building trust.

Further, even restricting ourselves to just fairness or robustness reveals fraught, contested questions. There is no perfect metric or objective to make a system truly trustworthy along these lines, with many competing and sometimes conflicting notions available. Decision-making systems in particular can suffer from challenges around defining and evaluating important metrics, due to poor data availability (either because of sensitivity around the issue, or sampling issues from a distribution affected by past policies), or more deeply, conflicting notions around what these concepts should really mean, and how a system can appropriately represent these different values. In practice, meeting these expectations involves consistent evaluation and re-evaluation, as norms and expectations change, not to mention shifts in the incoming data distribution caused by changes in the external world.

Finally, we consider the ways in which the work in this thesis falls short of these goals, to highlight the challenges and gaps in these research directions. As a helpful framing device, we use Selbst et al. (2019)'s framework for assessing the "abstraction traps" of applying machine learning tools to socio-technical systems, focusing on just three:

1. **The Framing Trap** ("Failure to model the entire system over which a social criterion ... will be enforced"): Chapters 2 and 3 do not consider the entirety of a system in which they might be implemented. For Chapter 2, a useful critique is presented in Dwork and Ilvento (2018) who show how a number of different, separately fair classifiers, can compose to create an unfair classifier. An appropriate metaphor is online advertising, where we can imagine sending LAFTR representations off to a number of different predictors, who then compete with fair predictions and compose in such a way to yield more unfairness than before. A useful critique of the "example-flagging" approach from Chapter 3 is given in Chapter 4, showing again how external decision-makers can change the picture drastically. Additional discussion in the conclusion of Chapter 4 considers the ethical tensions around using learning to defer to avoid the Framing Trap.

2. **The Formalism Trap** ("Failure to account for the full meaning of social concepts ... [which] cannot be resolved through mathematical formalisms"): Throughout the thesis, we focus on several pre-defined, mathematically formal metrics as a measure of performance, despite the fact that they do not fully capture the meaning of their related constructs. In Chapters 2 and 4 we use fairness definitions such as demographic parity and equalized odds as stand-ins for the broader social notion of fairness, despite the fact that these metrics are by their nature narrow, and unable to represent the construct of fairness in its full richness. In fact, no single metric can represent the construct "fairness" since it is highly contested: many may disagree on its correct meaning. In Chapter 3 we use underspecification as a stand-in for a notion of robustness, implying that models which can appropriately express uncertainty with respect to a given training set are more robust. While this is a desirable property, it does not fully capture the concept of what a truly robust system would contain, which should include opportunities to update or replace model given a faulty training set, as well as methods for recourse given an incorrect (not just an underspecified) prediction. Additionally, we use metrics like AUC

to represent improved accuracy of detecting underspecification, which has been criticized for not corresponding to practical notions of predictivity, since it ignores probability values and considers unimportant pieces of the ROC curve in its average (Halligan et al., 2015, Lobo et al., 2008).

3. **The Solutionism Trap** ("Failure to recognize the possibility that the best solution to a problem may not involve technology"): Throughout, we identify issues in existing machine learning approaches to high-stakes decision-making problems, and take as given that the most appropriate fix for these problems also lies in the the domain of machine learning. However, this may not always be the case — rather, sometimes the solution could be accomplished by removing the flawed technological system and increasing human oversight. Particularly in fields such as criminal justice, where predictions are politically and ethically fraught, challenging to evaluate, and where ground truth may not always be well-defined, this is an important consideration. For instance, prediction of criminality from a facial recognition system is not an appropriate application, as there is no plausible underlying signal our system could pick up on (Aguera y Arcas et al., 2017). Solutionist criticisms also have been levied of emotion-recognition AI in education, claiming that "mining the emotional lives of children is normatively wrong" (McStay, 2020) [1].

As previously stated, the work here must fall short of proposing complete or broadly-scoped solutions. However, we hope that it contributes to the discussion and technical literature on how we can harness the power of machine learning in ways that move us towards better, fairer high-stakes decision-making systems which deserve and engender trust from users and the public at large.

5.3 Looking Forward

To close off this thesis, we turn to the future, and list some compelling research questions for the future of the field of trustworthy ML, as conceptualized within the scope of this thesis. Below, we list six questions which we believe to be both intellectually intriguing and practically important. The first three are short-term questions i.e. more concrete, and the last three are long-term i.e. broader research directions.

1. **How do we learn representations which are amenable to test-time adaptation to downstream constraints?** In Chapter 2, we introduced the problem of learning transferable representations. This is a relevant one for practice, since representation learning is a hugely important practical tool for transfer learning. However, a more useful approach would be to learn a single representation and then, depending on the application, create a "fair" representation out of it i.e. based on a task, sensitive attribute, and fairness metrics, adapt the representation at test time so that any downstream classifier is guaranteed to perform well on that fairness metric. Creager et al. (2019) take a step towards this: however, they focus on demographic parity, and require that all sensitive attributes are available at training time. It would be useful to be able to adapt this to (for instance) equality of odds, equal classification rates, or causal

[1] To similar ends, Tene and Polonetsky (2013) develop the amusingly-named but intuitively useful "theory of creepy" for some analytical guidance about the normative status of technological applications.

fairness notions as well, while also being able to add new sensitive attributes or confounding factors on the fly.

2. **Can we define notions of robustness for domain generalization which are more flexible than "environments"?** Standard notions as introduced in Peters et al. (2016) are centred around "environments", which are categorical variables specifying a partitioning of the data. Robustness is then considered with respect to these categories (possibly some unobserved). It would be interesting to see if we can extend this to more general notions of auxiliary data, which could allow us to take into account similarity of different environments across a larger space. This could be useful if context variables are naturally continuous or multi-dimensional. For instance, if we are trying to build a model which works equally well for image recognition on photos from all countries (Atwood et al., 2020), we may be able to leverage a large amount of auxiliary data we have on similarity between different countries (geographically, culturally, environmentally, etc.). Srivastava et al. (2020) provides an interesting example of this, using human annotations to describe similarity between challenging input examples.

3. **How do we think about overlap, and can we reliably identify when it isn't present?** The notion of *overlap* has recently arisen as a useful indicator for identifying when we can expect models to generalize to new domains (Johansson et al., 2019, Ruan et al., 2021, Wang and Jordan, 2021), i.e. whether or not a new distribution has shared support with the training distribution is helpful for proving a model's ability to generalize to that new distribution. However, as D'Amour et al. (2021) notes, overlap in high dimensions is an extremely strong requirement. This raises two questions: first, can we develop tools for identifying when overlap assumptions are violated? Detecting underspecification (see Chapter 3) could provide one avenue forward, since underspecification can be a symptom of lack of overlap. Second, is overlap the right way to think about generalization in high-dimensional models? If not, can we find analogues for overlap which apply in high-dimensional, continuous spaces?

4. **What is the best way to combine human and machine decision-makers?** As discussed in Chapter 4, many applications of ML in decision-making involve joint prediction systems, containing both an ML tools and a human user, expert or supervisor. Dawes et al. (1989) famously noted that these two decision-making agents operate under different paradigms: the ML model exercises *actuarial* judgment — "automatic . . . and based on empirically established relations", whereas the human exercises their personal, *clinical* judgment, "combin[ing] or process[ing] information in his or her head". It is known that these decision-making paradigms can be combined in many ways: for instance, actuarial outputs can feed into a clinical process (consider an ML tool which outputs a risk score that a human expert considers); or clinical outputs can feed into an actuarial process (consider a checklist where each item is checked off according to personal judgment). These two approaches both have their strengths and weaknesses, and the best way for them to complement each other it currently an open question with connections to HCI, and one which has enormous consequences for the future of ML system in high-stakes settings.

5. **Can we analyze the role of ML models in feedback loops?** When ML models are deployed and re-deployed in the same system over time, they can sometimes create a "feedback

loop" (Ensign et al., 2018) i.e. past predictions from the model can affect future training data. While these are messy systems and very difficult to analyze, they can have hugely important impacts. One example of an important feedback loop around ML models is the recent literature around strategic classification (Hardt et al., 2016a) and performative prediction (Perdomo et al., 2020), which aim to specify how the interaction between a model and a downstream agent who is affected by the model's predictions can differ wildly from standard ML assumptions. For another example, consider recommender systems, a canonical example of feedback loops: the content that was recommended in the past by the system creates an environment which affects the future iterations of the recommender. While current recommender systems by no means take naive approaches, with work in the area fully aware of the challenges brought up by systematic missingness in the data (Marlin and Zemel, 2009, Melville and Mooney, 2004, Steck, 2010), the technical tools do not yet exist to grapple with the systemic effects of recommender systems on their environment. Given the propensity of polarization (Dandekar et al., 2013), misinformation (Hassan, 2019) and radicalization (Ribeiro et al., 2020) on recommender-influenced social media, it is paramount to get a better understanding of these dynamics.

6. **How far can formal language take us?** Using ML to navigate any sociotechnical system involves negotiating the boundaries between complex social dynamics and precise mathematical formalizations. For instance, choosing a fairness metric to constrain a classifier is one (limited) technique for incorporating desiderata from the social world into a technical system. A tension arises between the natural drive of machine learning research towards increasing abstraction and generality, and the need to be context-specific and outlier-sensitive when dealing with high-stakes human systems. Pushing these boundaries and developing a wider range of abstractions provides a route to more broad application of trustworthy ML. For instance, Martin Jr. et al. (2020) discuss procedures for using the graph-based method of *system dynamics* to help elicit models of the world from affected populations to develop better machine learning solutions. More work at these types of interfaces could help to better clarify where and how machine learning solutions can be deployed effectively in high-stakes scenarios.

Machine learning has high ambitions, and has the potential for great positive impact through more precise, controllable, and efficient prediction. However, these predictions also present incredible potential to harm. Due to the scale at which ML solutions are applied, any negative impact they have is multiplied many times over. For this reason, it is of the utmost importance that the technologies we learn and deploy are trustworthy. These questions are very fraught and very new, sitting at the centre of a number of fields of inquiry: we are only beginning to understand how to wrap our heads around the issues that can arise applying machine learning in high-stakes settings. Because of the nature of these problems, it is important to understand that no perfect solutions exist; rather, that we should continue to strive to build systems that move closer to alignment with the values we hold.

Appendix A

Notes for Chapter 2

A.1 Training Details

We used single-hidden-layer neural networks for each of our encoder, classifier and adversary, with 20 hidden units for the Health dataset and 8 hidden units for the Adult dataset. We also used a latent space of dimension 20 for Health and 8 for Adult. We train with L_C and L_{Adv} as absolute error, as discussed in Section 2.5, as a more natural relaxation of the binary case for our theoretical results. Our networks used leaky rectified linear units and were trained with Adam (Kingma and Ba, 2015) with a learning rate of 0.001 and a minibatch size of 64, taking one step per minibatch for both the encoder-classifier and the discriminator. When training ClassLearn in Algorithm 1 from a learned representation we use a single hidden layer network with half the width of the representation layer, i.e., g. ReprLearn (i.e., LAFTR) was trained for a total of 1000 epochs, and ClassLearn was trained for at most 1000 epochs with early stopping if the training loss failed to reduce after 20 consecutive epochs.

To get the fairness-accuracy tradeoff curves in Figure 2.3, we sweep across a range of fairness coefficients $\gamma \in [0.1, 4]$. To evaluate, we use a validation procedure. For each encoder training run, model checkpoints were made every 50 epochs; r classifiers are trained on each checkpoint (using r different random seeds), and epoch with lowest median error $+\Delta$ on validation set was chosen. We used $r = 7$. Then r more classifiers are trained on an unseen test set. The median statistics (taken across those r random seeds) are displayed.

For the transfer learning expriment, we used $\gamma = 1$ for models requiring a fair regularization coefficient.

Appendix B

Notes for Chapter 3

B.1 Ensembles: Other Datasets

We show in Figures B.1 and B.2 the same plots as Fig. 3.3 for other tabular datasets, demonstrating the strongly linear relationship between underspecification score and ensemble standard deviation.

B.2 The Lanczos Iteration: Futher Details

The Lanczos iteration (Lanczos, 1950) is a method for tridiagonalizing a Hermitian matrix. It can be thought of as a variant of power iteration, iteratively building a larger basis through repeatedly multiplying an initial vector by M, ensuring orthogonality at each step. Once M has been tridiagonalized, computing the final eigendecomposition is relatively simple — a number of specialized algorithms exist which are relatively fast ($O(p^2)$) (Cuppen, 1980, Dhillon, 1997).

The Lanczos iteration is simple to implement, but presents some challenges. The first challenge is numerical instability — when computed in floating point arithmetic, the algorithm is no longer guaranteed to return a good approximation to the true eigenbasis, or even an orthogonal one. As such, the standard implementation of the Lanczos iteration is unstable, and can be inaccurate even on simple problems. Fortunately, solutions exist: a procedure known as two-step classical Gram-Schmidt orthogonalization — which involves ensuring *twice* that each new vector is linearly independent of the previous ones — is guaranteed to produce an orthogonal basis, with errors on the order of machine roundoff (Giraud et al., 2005). A second potential challenge is presented by the stochasticity of minibatch computation. Since we access our Hessian H only through Hessian-vector products,

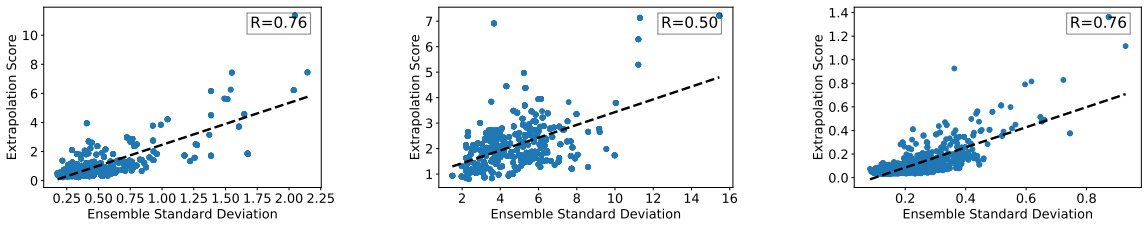


Figure B.1: Datasets are (left to right) Boston, Diabetes, Abalone. Dotted line represents linear fit of data. R is Pearson correlation coefficient.

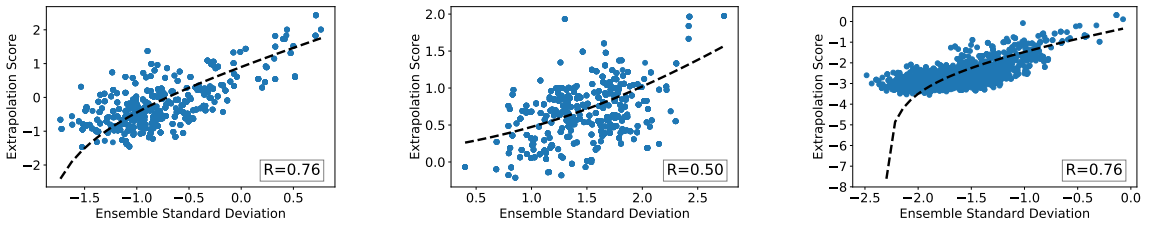


Figure B.2: Datasets are (left to right) Boston, Diabetes, Abalone. Each axis is log-scaled. Dotted line represents linear fit of data. R is Pearson correlation coefficient.

we must use only minibatch computation at each stage. This means that each iteration will be stochastic, which will decrease the accuracy (but not the orthogonality) of the eigenvector estimates provided. However, in practice, we found that even fairly noisy estimates were nonetheless useful — see Sec. 3.6.1 for more discussion.

B.2.1 Lanczos Algorithm Code Snippet

The Lanczos algorithm is quite simple to implement. Figure B.3 shows a a short implementation using Python/Numpy (Oliphant, 2006).

B.3 Simulated Features - Other Datasets

In Section 3.6.2, we present results for an experiment where we aim to detect broken collinearities in the feature space. In Figures B.4, B.5, and B.6, we show results on three more tabular datasets. See the main text for more experimental details.

B.4 Experimental Details

Here we give more experimental details on the datasets and models used.

B.4.1 Datasets

Tabular Datasets

We subtracted the mean and divided by the standard deviation for each feature (as calculated from the training set).

Boston (Harrison Jr and Rubinfeld, 1978) and Diabetes (Efron et al., 2004). These datasets were loaded from Scikit-Learn (Pedregosa et al., 2011).

Abalone (Nash et al., 1994). This dataset was downloaded from the UCI repository (Dua and Graff, 2017) at `http://archive.ics.uci.edu/ml/datasets/Abalone`. We converted sex to a three-dimensional one-hot vector (*M, F, I*).

```
import numpy as np

def lanczos(matmul_fn, dim, num_iters, eps=1e-8):
    '''Given implicit access to a matrix M of size (dim x dim)
    through mamtul_fn, tridiagonalize M into diagonal elements
    Alpha and off-diagonal elements Beta. Also return the
    associated orthonormal basis Q.'''

    # Initialize orthonormal basis.
    Q = [np.zeros((dim, 1))]
    # Initialize off-diagonal elements.
    Beta = []
    # Initialize diagonal elements.
    Alpha = []

    # Begin with random initial vector.
    q = np.random.uniform(size=(dim, 1))
    Q.append(q / np.linalg.norm(q))
    Q_k_range = Q[0]

    for k in range(1, num_iters + 1):
        # Compute next step of power iteration.
        z = matmul_fn(Q[k])
        Alpha.append(np.matmul(Q[k].T, z))
        Q_k_range = np.concatenate([Q_k_range, Q[k]], axis=1)

        # Reorthogonalize using two-step classical Gram-Schmidt.
        for _ in range(2):
            z_orth = np.sum(np.matmul(z.T, Q_k_range) *
                    Q_k_range, axis=1, keepdims=True)
            z = z - z_orth

        Beta.append(np.linalg.norm(z))
        Q.append(z / Beta[k])
        # Check for convergence.
        if np.linalg.norm(z) < eps:
            break

    return Q, Beta, Alpha
```

Figure B.3: Example Python implementation of Lanczos algorithm for tridiagonalizing an implicit matrix M.

WineQuality (Cortez et al., 2009). This dataset was downloaded from the UCI repository (Dua and Graff, 2017) at `http://archive.ics.uci.edu/ml/datasets/Wine+Quality`. We used both red and white wines.

B.4.2 MNIST, FashionMNIST and CelebA.

These datasets were loaded using Tensorflow Datasets `https://github.com/tensorflow/datasets`. We divided the pixel values by 255 and, for MNIST and FashionMNIST, binarized by thresholding

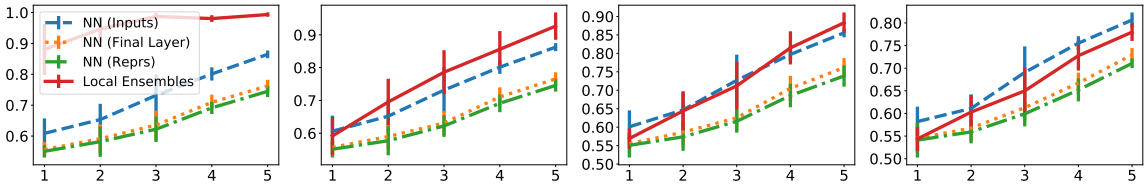


Figure B.4: Boston dataset

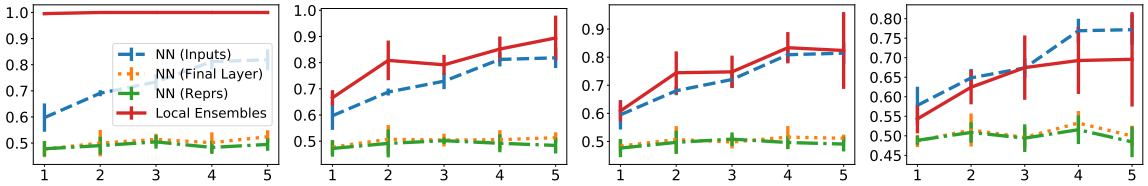


Figure B.5: Diabetes dataset

at 0.7.

B.4.3 Experimental Details

Toy Data Experiments

For the first experiment, we train a two-layer neural network with 3 hidden units in each layer and tanh units. We train for 400 optimization steps using minibatch size 32. Our data is generated from $y = \sin(4x) + \mathcal{N}(0, \frac{1}{4})$. We generate 200 training points, 100 test points, and 200 OOD points. We aggregate Y over a grid of 10 points from -1 to 1, with aggregation function min. We run the Lanczos algorithm until convergence.

For the second experiment, we train a two-layer neural network with 5 hidden units in each layer and ReLU units. Our data is generated from $y = \beta x^2 + \mathcal{N}(0, 1)$. Our training set consists of x drawn uniformly from $[-0.5, 0.5]$ and $[2.5, 3.5]$. However, at test time, we will consider $x \in [-3, 6]$. We generate 200 training points, 100 test points, and 200 OOD points. We aggregate Y over a grid of 5 points from -6 to 9, with aggregation function min.

Simulated Features

For each dataset we use the same setup. We use a two-layer MLP with ReLU activations and hidden layer sizes of 20 and 100. We trained all models with mean squared error loss. We use batch size 64, patience 100 and a 100-step running average window for estimating current performance. For the Lanczos iteration, we run up to 2000 iterations. We always report numbers from the final iteration

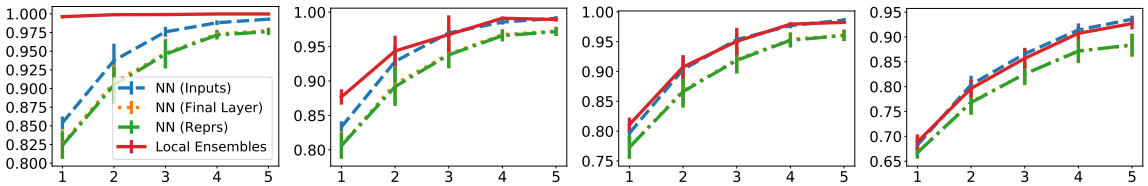


Figure B.6: Abalone dataset

Test Set	M/E	M/H	A/E	A/H
In-Distribution	0.03	0.06	0.01	0.02
Out-of-Distribution	0.98	0.96	0.90	0.93

Table B.1: Error rate for in and out of distribution test set with correlated latent factors setup. Column heading denotes in-distribution definitions: labels are M (Male) and A (Attractive); spurious correlate are E (Eyeglasses) and H (Wearing Hat). Image is in-distribution iff label == spurious correlate.

run. For estimating the Hessian in the HVPs in the Lanczos iteration, we use batch size 32 and sample 5 minibatches.

To pre-process the data, we first split randomly out 30% of the dataset as OOD. We choose 2 random features i, j and a number $\beta \sim \mathcal{U}(0, 1)$, and generate the new feature $\tilde{x} = \beta x[i] + (1 - \beta)x[j]$. We also normalize this feature by its mean and standard deviation as calculated on the training set. We add random noise to the features after splitting into in-distribution and OOD — meaning we are not redrawing from the same marginal noise distribution. We use 1000 examples from in-distribution and OOD for testing.

Correlated Latent Factors

We use a CNN with ReLU hidden activations. We use two convolutional layers with 50 and 20 filters each and stride size 2 for a total of 1.37 million parameters. We trained all models with cross entropy loss. We use an extra dense layer on top with 30 units. We use batch size 32, patient 100 steps, and a 100-step running average window for estimating current performance. We sample the validation set randomly as 20% of the training set. For the Lanczos iteration, we run 3000 iterations. We always report numbers from the final iteration run. We use 500 examples from in-distribution and OOD for testing. For estimating the Hessian in the HVPs in the Lanczos iteration, we use batch size 16 and sample 5 minibatches.

Active Learning

We use a CNN with ReLU hidden activations. We use two convolutional layers with 16 and 32 layers, stride size 5, and a dense layer on top with 64 units. We trained all models with mean squared error loss. We use batch size 32, patient 100 steps, and a 100-step running average window for estimating current performance.

B.5 Correlated Latent Factors

B.5.1 Performance of Binary Classifiers

In Section 3.6.3, we discuss an experiment where we correlated a latent label L and confounder C attribute. Table B.1 shows the in-distribution and out-of-distribution test error. These are drastically different, meaning that learning to detect this type of underspecification is critical to maintain model performance.

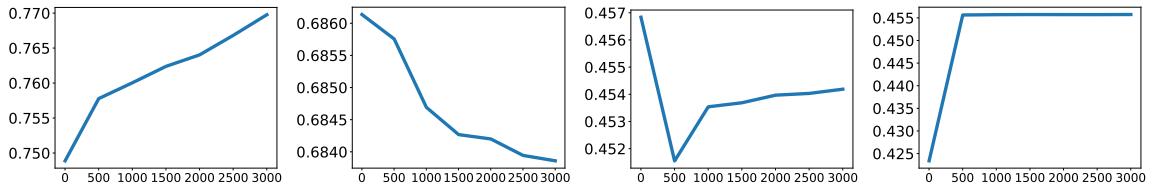


Figure B.7: Latent Factors OOD detection task, using loss gradient. Y-axis shows AUC, X-axis shows the number of eigenvectors estimated by the Lanczos algorithm, data sampled every 500 eigenvectors. Tasks from left to right are *M/E*, *M/H*, *A/E*, *A/H*.

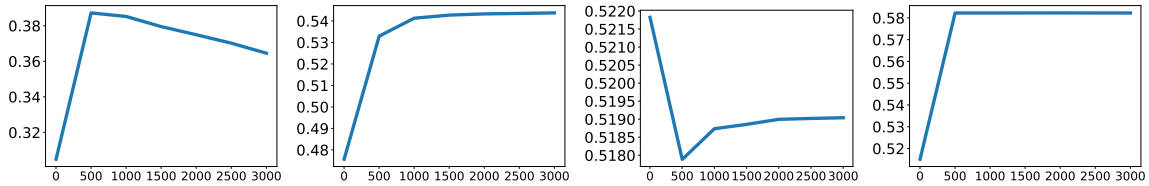


Figure B.8: Latent Factors OOD detection task, using loss gradient. Y-axis shows AUC, X-axis shows the number of eigenvectors estimated by the Lanczos algorithm, data sampled every 500 eigenvectors. Tasks from left to right are *M/E*, *M/H*, *A/E*, *A/H*.

B.5.2 Behaviour of AUC with More Estimated Eigenvectors

In Fig. B.7 and B.8, we show that the tasks present differing behaviours as more eigenvectors are estimated. We observe that for the *Male/Eyeglasses* and *Attractive/WearingHat* tasks, we get improved performance with more eigenvectors, but for the others we do not necessarily see the same improvements. Interestingly, this upward trend occurs both times that our method achieves a statistically significant improvement over baselines. It is unclear why this occurs for some settings of the task and not others, but we hypothesize that this is a sign that the method is working more correctly in these settings.

B.5.3 Relationship between Loss Gradient and *MaxProb* Method

As discussed in Sec. 3.6.3, we have the relationship between the loss ℓ, prediction $\hat{Y}$, and parameters θ: $\nabla_\theta \ell = \nabla_{\hat{Y}} \ell \cdot \nabla_\theta \hat{Y}$. Using min as an aggregation function, we find that $\min_{Y \in \{0,1\}} |\nabla_{\hat{Y}} \ell(Y, \hat{Y})|$ has an inverted V-shape (Fig. B.9). This is a similar shape to $1 - \max(\hat{Y}, 1 - \hat{Y})$, which is the metric implicitly measured by *MaxProb*.

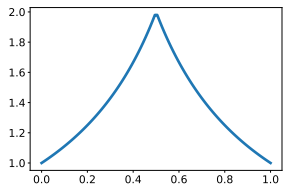
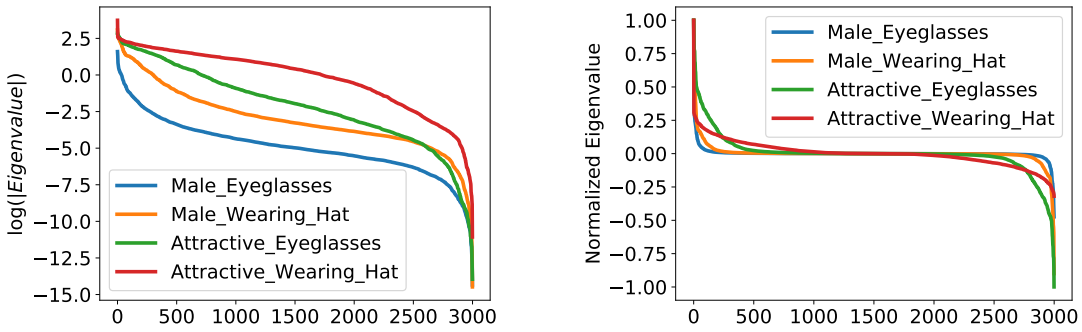


Figure B.9: X-axis is $\hat{Y}$, Y-axis is $\min_{Y \in \{0,1\}} \nabla_{\hat{Y}} \ell(Y, \hat{Y})$.

B.5.4 Estimated Eigenspectrum of Different Correlated Latent Factor Tasks

In Fig. B.10, we examine the estimated eigenspectrums of the four tasks we present in the correlated latent factors experiment, to see if we can detect a reason why performance might differ on these tasks.

In Fig. B.10a, we show that the two tasks with the label *Attribute*, the eigenvalues are larger. These are also the tasks where the loss-gradient-based variant of local ensembles failed, indicating



(a) Log absolute eigenvalues

(b) Normalized eigenvalues

Figure B.10: We show the estimated eigenspectrum of the four CNNs we train on the correlated latent factors task. In Fig. B.10a, we show the absolute estimated eigenvalues sorted by absolute value, on a log scale. In Fig. B.10b, we show the estimated eigenvalues divided by the maximum estimated eigenvalues. We only show 3000 estimated eigenvalues because we ran the Lanczos iteration for only 3000 iterations, meaning we did not estimate the rest of the eigenspectrum.

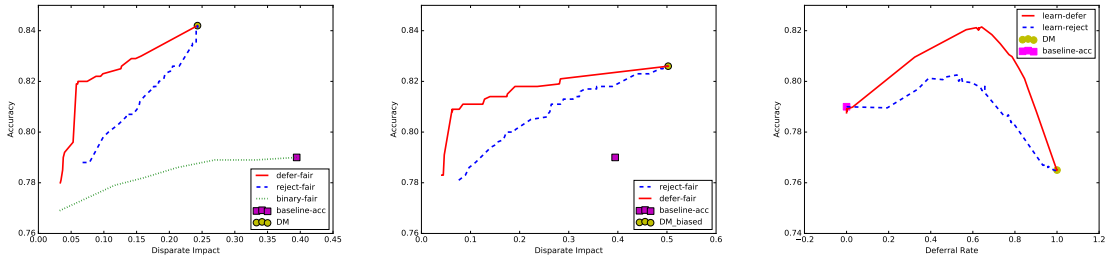
that that variant of the method may be worse at handling larger eigenvalues.

In Fig. B.10b, we show that the two tasks where the local ensembles methods performed best ($M/E, A/H$, achieving statistically significant improvements over the baselines at a 95% confidence level, and also showing improvement as more eigenvectors were estimated), the most prominent negative eigenvalue is relatively smaller magnitude compared to the most prominent positive eigenvalue. This could mean that the local ensembles method was less successful in the other tasks ($M/H, A/E$) simply because those models were not trained close enough to a convex minimum and still had fairly significant eigenvalues.

Appendix C

Notes for Chapter 4

C.1 Learning to Defer to Three Types of DM: Health Results



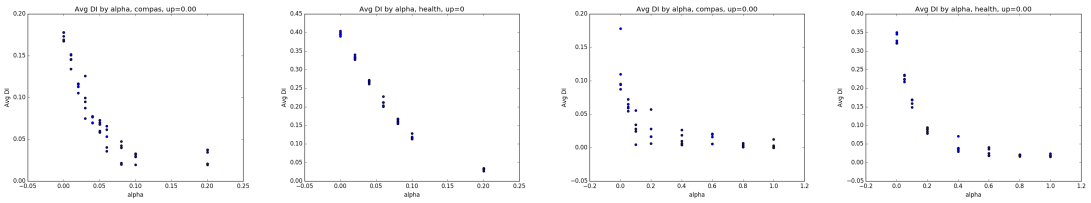
(a) Health, High-Accuracy DM (b) Health, Highly-Biased DM (c) Health, Inconsistent DM

Figure C.1: Comparing learning-to-defer, rejection learning and binary models. Health dataset. Each column is with a different DM scenario, according to captions. In left and centre columns, X-axis is fairness (lower is better); in right column, X-axis is deferral rate. Y-axis is accuracy for all. The red square is a baseline binary classifier, trained only to optimize accuracy; dashed line is the fair rejection model; solid line is a fair deferring model. Yellow circle shows DM alone. In left and centre columns, dotted line is a binary model also optimizing fairness. Each experiment is a hyperparameter sweep over $\gamma_{reject}/\gamma_{defer}$ (in left and centre columns, also α_{fair}; in right column, $\alpha_{fair} = 0$; for results with $\alpha_{fair} \geq 0$, see Appendix C.6).

Results for Health dataset corresponding to Fig. 4.3 in Sec. 4.5.1.

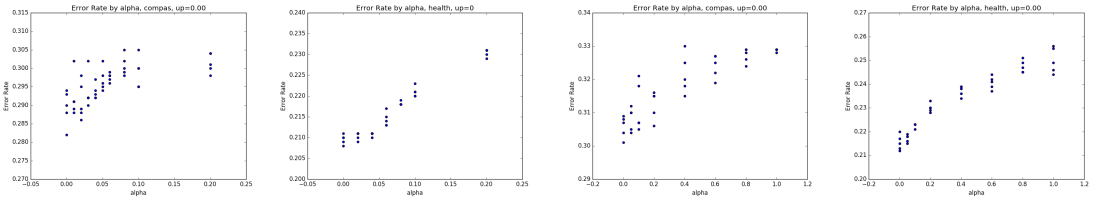
C.2 Results: Binary Classification with Fair Regularization

The results in Figures C.2 and C.3 roughly replicate the results from (Bechavod and Ligett, 2017), who also test on the COMPAS dataset. Their results are slightly different for two reasons: 1) we use a 1-layer NN and they use logistic regression; and 2) our training/test splits are different from theirs - we have more examples in our training set. However, the main takeaway is similar: regularization is an effective way to reduce DI without making too many more errors.



(a) COMPAS, MLP (b) Health, MLP (c) COMPAS, BNN (d) Health, BNN

Figure C.2: Relationship of DI to α, the coefficient on the DI regularizer, 5 runs for each value of α. Two datasets, COMPAS and Health. Two learning algorithms, MLP and Bayesian weight uncertainty.



(a) COMPAS, MLP (b) Health, MLP (c) COMPAS, BNN (d) Health, BNN

Figure C.3: Relationship of error rate to α, the coefficient on the DI regularizer, 5 runs for each value of α. Two datasets, COMPAS and Health. Two learning algorithms, MLP and Bayesian weight uncertainty.

C.3 Dataset and Experiment Details

We show results on two datasets. The first is the COMPAS recidivism dataset, made available by ProPublica (Kirchner and Larson, 2016) [1]. This dataset concerns recidivism: whether or not a criminal defendant will commit a crime while on bail. The goal is to predict whether or not the person will recidivate, and the sensitive variable is race (split into black and non-black). We used information about counts of prior charges, charge degree, sex, age, and charge type (e.g., robbery, drug possession). We provide one extra bit of information to our DM - whether or not the defendant *violently* recidivated. This clearly delineates between two groups in the data - one where the DM knows the correct answer (those who violently recidivated) and one where the DM has no extra information (those who did not recidivate, and those who recidivated non-violently). This simulates a real-world scenario where a DM, unbeknownst to the model, may have extra information on a subset of the data. The simulated DM had a 24% error rate, better than the baseline model's 29% error rate. We split the dataset into 7718 training examples and 3309 test examples.

The second dataset is the Heritage Health dataset[2]. This dataset concerns health and hospitalization, particularly with respect to insurance. For this dataset, we chose the goal of predicting the Charlson Index, a comorbidity indicator, related to someone's chances of death in the next several years. We binarize the Charlson Index of a patient as 0/greater than 0. We take the sensitive variable to be age and binarize by over/under 70 years old. This dataset contains information on sex, age, lab test, prescription, and claim details. The extra information available to the DM is the primary

[1] downloaded from https://github.com/propublica/compas-analysis
[2] Downloaded from https://www.kaggle.com/c/hhp

condition group of the patient (given in the form of a code e.g., 'SEIZURE', 'STROKE', 'PNEUM'). Again, this simulates the situation where a DM may have extra information on the patient's health that the algorithm does not have access to. The simulated DM had a 16% error rate, better than the baseline model's 21% error rate. We split the dataset into 46769 training examples and 20044 test examples.

We trained all models using a fully-connected two-layer neural network with a logistic non-linearity on the output, where appropriate. We used 5 sigmoid hidden units for COMPAS and 20 sigmoid hidden units for Health. We used ADAM (Kingma and Ba, 2015) for gradient descent. We split the training data into 80% training, 20% validation, and stopped training after 50 consecutive epochs without achieving a new minimum loss on the validation set.

C.4 Details on Optimization: Hard Thresholds

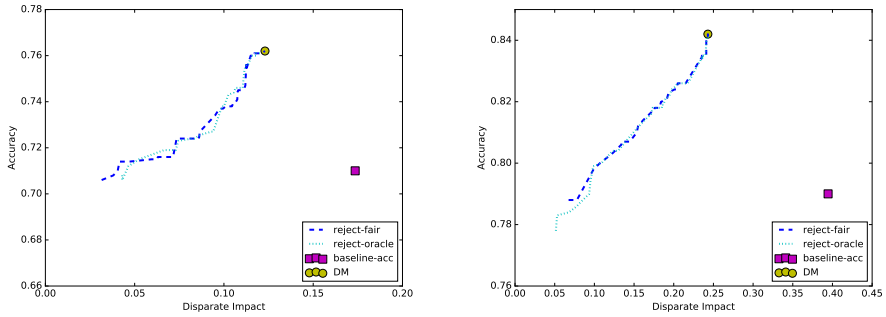
We now explain the post-hoc threshold optimization search procedure we used. To encourage fairness, we can learn a separate set of thresholds for each group, then apply the appropriate set of thresholds to each example. Since it is a very small space (one dimension per threshold = 4 dimensions;), we used a random search. We sampled 1000 combinations of thresholds, picked the thresholds which minimized the loss on one half of the test set, and evaluated these thresholds on the other half of the test set. We do this for several values of α, γ in thresholding, as well as several values of α for the original binary model.

We did not sample thresholds from the [0, 1] interval uniformly. Rather we used the following procedure. We sampled our lower thresholds from the scores in the training set which were below 0.5, and our upper thresholds from the scores in the training set which were above 0.5. Our sampling scheme was guided by two principles: this forced 0.5 to always be in the PASS region; and this allowed us to sample more thresholds where the scores were more dense. If only choosing one threshold per class, we sampled from the entire training set distribution, without dividing into above 0.5 and below 0.5.

This random search was significantly faster than grid search, and no less effective. It was also faster and more effective than gradient-based optimization methods for thresholds - the loss landscape seemed to have many local minima.

C.5 Comparison of Learning to Defer with an Oracle in Training to Rejection Learning

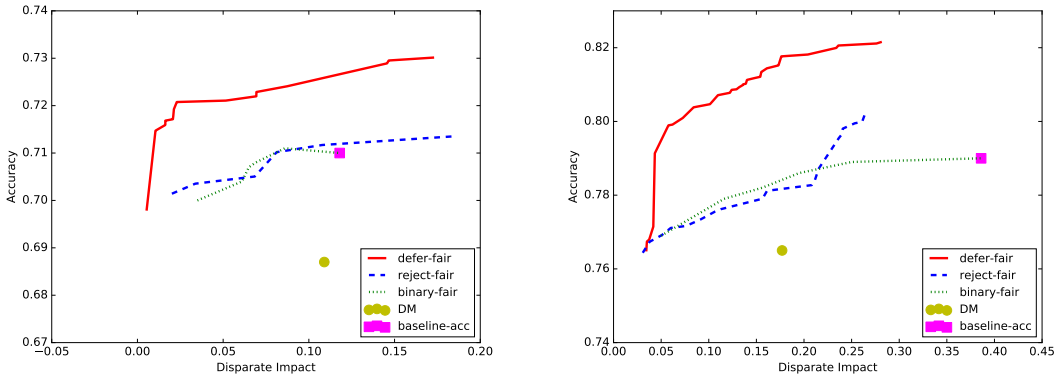
In Section 4.2.3, we discuss that rejection learning is similar to learning to defer training, except with a training DM who treats all examples similarly, in some sense. In Section 4.2.3, we show that theoretically these are equivalent. However, our fairness regularizer is not of the correct form for the proof in Sec.4.2.3 to hold. Here we show experimental evidence that the objectives are equivalent for the fair-regularized loss function. The plots in Figure C.4 compare these two models: rejection learning, and learning to defer with an oracle at training time, and the standard DM at test time. We can see that these models trade off between accuracy and fairness in almost an identical manner.



(a) COMPAS dataset (b) Health dataset

Figure C.4: Comparing model performance between learning to defer training with oracle as DM to rejection learning. At test time, same DM is used.

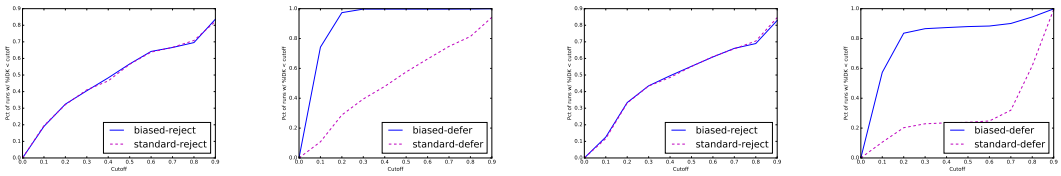
C.6 Results: Differentiable Learning-to-Defer Fairness with $\alpha_{fair} \geq 0$



(a) COMPAS dataset (b) Health dataset

Figure C.5: Comparing learning-to-defer, rejection learning and binary models. High-accuracy, ignores fairness DM. X-axis is fairness (lower is better). Y-axis is accuracy. The red square is a baseline binary classifier, trained only to optimize accuracy; dashed line is the fair rejection model; solid line is a fair deferring model. Yellow circle shows DM alone. In left and centre columns, dotted line is a binary model also optimizing fairness. Each experiment is a hyperparameter sweep over $\gamma_{reject}/\gamma_{defer}/\alpha_{fair}$.

We show here the results of the experiments with deferring fairly to a low accuracy, inconsistent DM with accuracy to extra information. The results are qualitatively similar to those in Figs. 4.3a and C.1a. However, it is worth noting here that the rejection learning results mostly overlap the binary model results. This means that if the DM is not taken into consideration through learning to defer, then the win of rejection learning over training a binary model can be minimal.



(a) Reject, A = 0 (b) Defer, A = 0 (c) Reject, A = 1 (d) Defer, A = 1

Figure C.6: Deferral rate for a range of hyperparameter settings, COMPAS dataset. X-axis is cutoff $\in [0, 1]$, line shows percentage of runs which had deferral rate below the cutoff. Blue solid line is models trained with biased DM, purple dashed line is with standard DM. Left column is rejection learning, right column is learning-to-defer. Top and bottom row split by value of the sensitive attribute A.

C.7 Results: Deferral Rates with a Biased DM

In Fig. C.6, we further analyze the difference in deferral and rejection rates between models trained with the biased DM (Fig. 4.3b) and standard DM (Fig. 4.3a). We ran the model for over 1000 different hyperparameter combinations, and show the distribution of the deferral rates of these runs on the COMPAS dataset, dividing up by defer/reject models, biased/standard DM, and the value of the sensitive attribute.

Notably, the deferring models are able to treat the two DM's differently, whereas the rejecting models are not. In particular, notice that the solid line (biased DM) is much higher in the low deferral regime, meaning that the deferring model, given a biased DM, almost defers on fewer than 20% of examples since the DM is of lower quality. Secondly, we see that the deferring model is also able to adapt to the biased DM by deferring at differing rates for the two values of the sensitive attribute — an effective response to a (mildly or strongly) biased DM who may already treat the two groups differently. This is another way in which a model that learns to defer can be more flexible and provide value to a system.

Bibliography

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. {TensorFlow}: A system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016a.

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016b.

Tameem Adel, Isabel Valera, Zoubin Ghahramani, and Adrian Weller. One-network adversarial fairness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2412–2420, 2019.

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. Towards a unified framework for fair and stable graph representation learning. In *Uncertainty in Artificial Intelligence*, pages 2114–2124. PMLR, 2021.

Blaise Aguera y Arcas, Margaret Mitchell, and Alexander Todorov. Physiognomy's new clothes, 2017. URL `https://medium.com/@blaisea/physiognomys-new-clothes-f2d4b59fdd6a`.

Ahmed Alaa and Mihaela Van Der Schaar. Discriminative jackknife: Quantifying uncertainty in deep learning via higher-order influence functions. In *International Conference on Machine Learning*, pages 165–174. PMLR, 2020.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Julia Angwin and Jeff Larson. Bias in criminal risk scores is mathematically inevitable, researchers say. In *https://www.propublica.org/article/bias-in-criminal-risk-scores-is-mathematically-inevitable-researchers-say*, 2016.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. In *https:www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing*, 2017.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

Josh Attenberg, Panagiotis G Ipeirotis, and Foster J Provost. Beat the machine: Challenging workers to find the unknown unknowns. *Human Computation*, 11(11), 2011.

James Atwood, Yoni Halpern, Pallavi Baljekar, Eric Breck, D Sculley, Pavel Ostyakov, Sergey I Nikolenko, Igor Ivanov, Roman Solovyev, Weimin Wang, et al. The inclusive images competition. In *The NeurIPS'18 Competition*, pages 155–186. Springer, 2020.

Tobias Baer and Vishnu Kamalnath. Controlling machine-learning algorithms and their biases. *McKinsey Insights*, 2017.

Chloé Bakalar, Renata Barreto, Stevie Bergman, Miranda Bogen, Bobbie Chern, Sam Corbett-Davies, Melissa Hall, Isabel Kloumann, Michelle Lam, Joaquin Quiñonero Candela, et al. Fairness on the ground: Applying algorithmic fairness approaches to production systems. *arXiv preprint arXiv:2103.06172*, 2021.

Aparna Balagopalan, David Yang, David Madras, Dylan Hadfield-Menell, Marzyeh Ghassemi, and Gillian Hadfield. Descriptive data labelling practices distort normative machine learning judgments. *In submission.*, 2021.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11405–11414, 2021.

Michelle Bao, Angela Zhou, Samantha Zottola, Brian Brubach, Sarah Desmarais, Aaron Horowitz, Kristian Lum, and Suresh Venkatasubramanian. It's compaslicated: The messy relationship between rai datasets and algorithmic fairness benchmarks. *arXiv preprint arXiv:2106.05498*, 2021.

Marco Barreno, Blaine Nelson, Anthony D Joseph, and J Doug Tygar. The security of machine learning. *Machine Learning*, 81(2):121–148, 2010.

Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. Relatif: Identifying explanatory training samples via relative influence. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1899–1909. PMLR, 26–28 Aug 2020. URL `https://proceedings.mlr.press/v108/barshan20a.html`.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

Tony Bates. What's right and what's wrong about coursera-style moocs. *EdTech in the Wild*, 2019.

Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of Marchine Learning Research*, 18:1–43, 2018.

Yahav Bechavod and Katrina Ligett. Learning fair classifiers: A regularization-inspired approach. *arXiv preprint arXiv:1707.00044*, 2017.

Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Richard Berk. *Criminal justice forecasts of risk: A machine learning approach*. Springer Science & Business Media, 2012.

Dimitris Bertsimas, Vishal Gupta, and Nathan Kallus. Data-driven robust optimization. *Mathematical Programming*, 167(2):235–292, 2018.

Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*, 2017.

Jean-Emmanuel Bibault, Philippe Giraud, and Anita Burgun. Big data and machine learning in radiation oncology: state of the art and future prospects. *Cancer letters*, 382(1):110–117, 2016.

John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128. Association for Computational Linguistics, 2006.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, pages 4349–4357, 2016.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Amanda Bower, Sarah N. Kitchen, Laura Niss, Martin J. Strauss, Alexander Vargas, and Suresh Venkatasubramanian. Fair Pipelines. *Workshop on Fairness, Accountability, and Transparency in Machine Learning*, July 2017. arXiv: 1707.00391.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Tim Brennan, William Dieterich, and Beate Ehret. Evaluating the predictive validity of the compas risk and needs assessment system. *Criminal Justice and behavior*, 36(1):21–40, 2009.

Liming Brotcke. Time to assess bias in machine learning models for credit decisions. *Journal of Risk and Financial Management*, 15(4):165, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Erik Brynjolfsson and Tom Mitchell. What can machine learning do? workforce implications. *Science*, 358(6370):1530–1534, 2017.

Nadia Burkart and Marco F Huber. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317, 2021.

Lowell W Busenitz and Jay B Barney. Differences between entrepreneurs and managers in large organizations: Biases and heuristics in strategic decision-making. *Journal of business venturing*, 12 (1):9–30, 1997.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. *Advances in neural information processing systems*, 30, 2017.

Giovanni Campagna, Rakesh Ramesh, Silei Xu, Michael Fischer, and Monica S Lam. Almond: The architecture of an open, crowdsourced, privacy-preserving, programmable virtual assistant. In *Proceedings of the 26th International Conference on World Wide Web*, pages 341–350, 2017.

Ray Worthy Campbell. Artificial intelligence in the courtroom: The delivery of justice in the age of machine learning. *Colo. Tech. LJ*, 18:323, 2020.

Ran Canetti, Aloni Cohen, Nishanth Dikkala, Govind Ramnarayan, Sarah Scheffler, and Adam Smith. From soft classifiers to hard decisions: How fair can we be? In *Proceedings of the conference on fairness, accountability, and transparency*, pages 309–318, 2019.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Dallas Card, Michael Zhang, and Noah A Smith. Deep weighted averaging classifiers. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 369–378, 2019.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 319–328, 2019.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

Won Ik Cho, Ji Won Kim, Seok Min Kim, and Nam Soo Kim. On measuring gender bias in translation of gender-neutral pronouns. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 173–181, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3824. URL `https://aclanthology.org/W19-3824`.

Hyunsun Choi, Eric Jang, and Alexander A Alemi. Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

C. Chow. An optimum character recognition system using decision function. *IEEE T. C.*, 1957.

C. Chow. On optimum recognition error and reject trade-off. *IEEE T. C.*, 1970.

C West Churchman. Guest editorial: Wicked problems, 1967.

CNA. Nursing Statistics - Canadian Nurses Association. `https://www.cna-aiic.ca/en/nursing/regulated-nursing-in-canada/nursing-statistics`, 2020. [Accessed 24-May-2022].

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

Jeff Conklin. *Wicked problems & social complexity*, volume 11. CogNexus Institute Napa, USA, 2006.

R Dennis Cook and Sanford Weisberg. *Residuals and influence in regression*. New York: Chapman and Hall, 1982.

Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *International Conference on Algorithmic Learning Theory*, pages 67–82. Springer, 2016.

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4): 547–553, 2009.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

Zac Cranko, Zhan Shi, Xinhua Zhang, Richard Nock, and Simon Kornblith. Generalised lipschitz regularisation equals distributional robustness. In *International Conference on Machine Learning*, pages 2178–2188. PMLR, 2021.

Elliot Creager, David Madras, Jörn-Henrik Jacobsen, Marissa Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. Flexibly fair representation learning by disentanglement. In *International Conference on Machine Learning*, pages 1436–1445. PMLR, 2019.

Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Exchanging lessons between algorithmic fairness and domain generalization. 2020a.

Elliot Creager, David Madras, Toniann Pitassi, and Richard Zemel. Causal modeling for fairness in dynamical systems. In *International Conference on Machine Learning*, pages 2185–2195. PMLR, 2020b.

Jan JM Cuppen. A divide and conquer method for the symmetric tridiagonal eigenproblem. *Numerische Mathematik*, 36(2):177–195, 1980.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020.

Pranav Dandekar, Ashish Goel, and David T Lee. Biased assimilation, homophily, and the dynamics of polarization. *Proceedings of the National Academy of Sciences*, 110(15):5791–5796, 2013.

Shai Danziger, Jonathan Levav, and Liora Avnaim-Pesso. Extraneous factors in judicial decisions. *Proceedings of the National Academy of Sciences*, 108(17):6889–6892, 2011.

Robyn M Dawes, David Faust, and Paul E Meehl. Clinical versus actuarial judgment. *Science*, 243 (4899):1668–1674, 1989.

Abir De, Paramita Koley, Niloy Ganguly, and Manuel Gomez-Rodriguez. Regression under human assistance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2611–2620, 2020.

Abir De, Nastaran Okati, Ali Zarezade, and Manuel Gomez-Rodriguez. Classification under human assistance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5905–5913, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arxiv. *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 17, 2019.

Inderjit S Dhillon. A new o (n2) algorithm for the symmetric tridiagonal eigenvalue/eigenvector problem. Technical report, CALIFORNIA UNIV BERKELEY GRADUATE DIV, 1997.

William Dieterich, Christina Mendoza, and Tim Brennan. Compas risk scales: Demonstrating accuracy equity and predictive parity. *Northpointe Inc*, 7(4), 2016.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

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*, pages 55–70, 2018.

Michele Donini, Luca Oneto, Shai Ben-David, John S Shawe-Taylor, and Massimiliano Pontil. Empirical risk minimization under fairness constraints. *Advances in Neural Information Processing Systems*, 31, 2018.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

John Duchi and Hongseok Namkoong. Variance-based regularization with convex objectives. *The Journal of Machine Learning Research*, 20(1):2450–2504, 2019.

John Duchi, Peter Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv preprint arXiv:1610.03425*, 2016.

Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

Cynthia Dwork and Christina Ilvento. Fairness under composition. *arXiv preprint arXiv:1806.06122*, 2018.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. ACM, 2012.

Alexander D'Amour, Peng Ding, Avi Feller, Lihua Lei, and Jasjeet Sekhon. Overlap in observational studies with high-dimensional covariates. *Journal of Econometrics*, 221(2):644–654, 2021.

Harrison Edwards and Amos Storkey. Censoring representations with an adversary. *International Conference on Learning Representations*, 2016.

Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

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*, pages 160–171. PMLR, 2018.

Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

Lydia Fischer and Thomas Villmann. A probabilistic classifier model with adaptive rejection option. Technical Report 1865-3960, January 2016. URL `https://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_01_2016.pdf`.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, 20(177):1–81, 2019.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

Walter Bryce Gallie. Essentially contested concepts. In *Proceedings of the Aristotelian society*, volume 56, pages 167–198. JSTOR, 1955.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Rui Gao, Xi Chen, and Anton J Kleywegt. Distributional robustness and regularization in statistical learning. *arXiv preprint arXiv:1712.06050*, 2017.

Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *International Conference on Machine Learning*, pages 2151–2159. PMLR, 2019.

Mohammad Ghavamzadeh, Marek Petrik, and Yinlam Chow. Safe policy improvement by minimizing robust baseline regret. In *Advances in Neural Information Processing Systems*, pages 2298–2306, 2016.

Judy Wawira Gichoya, Imon Banerjee, Ananth Reddy Bhimireddy, John L Burns, Leo Anthony Celi, Li-Ching Chen, Ramon Correa, Natalie Dullerud, Marzyeh Ghassemi, Shih-Cheng Huang, et al. Ai recognition of patient race in medical imaging: a modelling study. *The Lancet Digital Health*, 2022.

Ryan Giordano, William Stephenson, Runjing Liu, Michael Jordan, and Tamara Broderick. A swiss army infinitesimal jackknife. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1139–1147, 2019.

Luc Giraud, Julien Langou, Miroslav Rozložník, and Jasper van den Eshof. Rounding error analysis of the classical gram-schmidt orthogonalization process. *Numerische Mathematik*, 101(1):87–100, 2005.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Joel Goh and Melvyn Sim. Distributionally robust optimization and its tractable approximations. *Operations research*, 58(4-part-1):902–917, 2010.

Ian M Goldstein, Julie Lawrence, and Adam S Miner. Human-machine collaboration in cancer and beyond: the centaur care model. *JAMA oncology*, 3(10):1303–1304, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014a.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

Joseph K Goodman, Cynthia E Cryder, and Amar Cheema. Data collection in a flat world: The strengths and weaknesses of mechanical turk samples. *Journal of Behavioral Decision Making*, 26 (3):213–224, 2013.

Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2007.

Nina Grgić-Hlaca, Muhammad Bilal Zafar, Krishna P. Gummadi, and Adrian Weller. On Fairness, Diversity and Randomness in Algorithmic Decision Making. *Workshop on Fairness, Accountability, and Transparency in Machine Learning*, June 2017. arXiv: 1706.10208.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *International Conference on Learning Representations*, 2021.

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*, pages 1321–1330. JMLR. org, 2017.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.

Dylan Hadfield-Menell and Gillian K Hadfield. Incomplete contracting and ai alignment. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 417–422, 2019.

Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. In *Advances in neural information processing systems*, pages 3909–3917, 2016.

Sara Hajian, Josep Domingo-Ferrer, Anna Monreale, Dino Pedreschi, and Fosca Giannotti. Discrimination-and privacy-aware patterns. *Data Mining and Knowledge Discovery*, 29(6):1733–1782, 2015.

Malay Haldar, Mustafa Abdool, Prashant Ramanathan, Tao Xu, Shulin Yang, Huizhong Duan, Qing Zhang, Nick Barrow-Williams, Bradley C Turnbull, Brendan M Collins, et al. Applying deep learning to airbnb search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1927–1935, 2019.

Steve Halligan, Douglas G Altman, and Susan Mallett. Disadvantages of using the area under the receiver operating characteristic curve to assess imaging tests: a discussion and proposal for an alternative approach. *European radiology*, 25(4):932–939, 2015.

Kelly Hannah-Moffat. Actuarial sentencing: An "unsettled" proposition. *Justice Quarterly*, 30(2): 270–296, 2013.

Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pages 111–122, 2016a.

Moritz Hardt, Eric Price, Nati Srebro, et al. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016b.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL `https://doi.org/10.1038/s41586-020-2649-2`.

David Harrison Jr and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102, 1978.

Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2018.

Taha Hassan. Trust and trustworthiness in social recommender systems. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 529–532, 2019.

Sue Ellen Haupt, Jim Cowie, Seth Linden, Tyler McCandless, Branko Kosovic, and Stefano Alessandrini. Machine learning for applied weather prediction. In *2018 IEEE 14th international conference on e-science (e-Science)*, pages 276–277. IEEE, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Ursula Hébert-Johnson, Michael P Kim, Omer Reingold, and Guy N Rothblum. Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, 2018.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018.

Tianqi Hou, KY Michael Wong, and Haiping Huang. Minimal model of permutation symmetry in unsupervised learning. *Journal of Physics A: Mathematical and Theoretical*, 52(41):414001, 2019.

Lily Hu and Yiling Chen. A short-term intervention for long-term fairness in the labor market. In *Proceedings of the 2018 World Wide Web Conference*, pages 1389–1398, 2018.

Lily Hu and Issa Kohler-Hausmann. What's sex got to do with machine learning? In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 513–513, 2020.

Ruitong Huang, Bing Xu, Dale Schuurmans, and Csaba Szepesvári. Learning with a strong adversary. *arXiv preprint arXiv:1511.03034*, 2015.

Peter J Huber. Robust statistics. In *International encyclopedia of statistical science*, pages 1248–1251. Springer, 2011.

Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 19(2):433–450, 1990.

Aapo Hyvarinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR, 2019.

Christina Ilvento. Metric learning for individual fairness. In *1st Symposium on Foundations of Responsible Computing*, 2020.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 624–635, 2021.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *International Conference on Learning Representations*, 2016.

Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. *Advances in neural information processing systems*, 31, 2018.

Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International conference on machine learning*, pages 3020–3029, 2016.

Fredrik D Johansson, David Sontag, and Rajesh Ranganath. Support and invertibility in domain-invariant representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 527–536. PMLR, 2019.

Fredrik D Johansson, Uri Shalit, Nathan Kallus, and David Sontag. Generalization bounds and representation learning for estimation of potential outcomes and causal effects. *arXiv preprint arXiv:2001.07426*, 2020.

Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*, pages 325–333, 2016.

Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th annual international symposium on computer architecture*, pages 1–12, 2017.

Christopher Jung, Michael Kearns, Seth Neel, Aaron Roth, Logan Stapleton, and Zhiwei Steven Wu. An algorithmic framework for fairness elicitation. In *2nd Symposium on Foundations of Responsible Computing (FORC 2021)*, 2021.

Nathan Kallus and Angela Zhou. Residual unfairness in fair machine learning from prejudiced data. In *International Conference on Machine Learning*, pages 2439–2448. PMLR, 2018.

F. Kamiran and T. Calders. Classifying without discriminating. In *2nd International Conference on Computer, Control and Communication, 2009. IC4 2009*, pages 1–6, February 2009. doi: 10.1109/IC4.2009.4909197.

Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer, 2012.

Sanjay Kariyappa and Moinuddin K Qureshi. Improving adversarial robustness of ensembles with diversity training. *arXiv preprint arXiv:1901.09981*, 2019.

Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *35th International Conference on Machine Learning, ICML 2018*, pages 4008–4016. International Machine Learning Society (IMLS), 2018.

Vijay Keswani, Matthew Lease, and Krishnaram Kenthapadi. Towards unbiased and accurate deferral to multiple experts. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 154–165, 2021.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

Michael P Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254, 2019.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Lauren Kirchner and Jeff Larson. How we analyzed the compas recidivism algorithm. In *https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm*, 2016.

Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ziad Obermeyer. Prediction policy problems. *American Economic Review*, 105(5):491–95, 2015.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67, page 43. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR. org, 2017.

Junpei Komiyama, Akiko Takeda, Junya Honda, and Hajime Shimao. Nonconvex optimization for regression with fairness constraints. In *International conference on machine learning*, pages 2737–2746. PMLR, 2018.

O Krafft and N Schmitz. A note on hoeffding's inequality. *Journal of the American Statistical Association*, 64(327):907–912, 1969.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021.

James Kuczmarski. Reducing gender bias in Google Translate — blog.google. `https://blog.google/products/translate/reducing-gender-bias-google-translate/`, 2018. [Accessed 24-May-2022].

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, pages 4069–4079, 2017.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Cornelius Lanczos. *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. United States Governm. Press Office Los Angeles, CA, 1950.

Tosca Lechner, Shai Ben-David, Sushant Agarwal, and Nivasini Ananthakrishnan. Impossibility results for fair representations. *arXiv preprint arXiv:2107.03483*, 2021.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2:18, 2010.

Yoonho Lee, Huaxiu Yao, and Chelsea Finn. Diversify and disambiguate: Learning from underspecified data. *arXiv preprint arXiv:2202.03418*, 2022.

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. *Advances in neural information processing systems*, 32, 2019.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

Jorge M Lobo, Alberto Jiménez-Valverde, and Raimundo Real. Auc: a misleading measure of the performance of predictive distribution models. *Global ecology and Biogeography*, 17(2):145–151, 2008.

Francesco Locatello, Gabriele Abbati, Thomas Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. On the fairness of disentangled representations. *Advances in Neural Information Processing Systems*, 32, 2019.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 6348–6359. PMLR, 2020.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard S Zemel. The variational fair autoencoder. In *ICLR*, 2016.

Chaochao Lu, Yuhuai Wu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. Invariant causal representation learning for out-of-distribution generalization. In *International Conference on Learning Representations*, 2021.

Zhiyun Lu, Eugene Ie, and Fei Sha. Uncertainty estimation with infinitesimal jackknife, its distribution and mean-field approximation. 2020.

Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. In *NIPS Workshop on Adversarial Training*, 2016.

David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *International Conference on Learning Representations*, 2017.

David Madras and Richard Zemel. Identifying and benchmarking natural out-of-context prediction problems. *Advances in Neural Information Processing Systems*, 34, 2021.

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pages 3384–3393. PMLR, 2018a.

David Madras, Toniann Pitassi, and Richard Zemel. Predict responsibly: improving fairness and accuracy by learning to defer. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6150–6160, 2018b.

David Madras, James Atwood, and Alexander D'Amour. Detecting extrapolation with local ensembles. In *International Conference on Learning Representations*, 2019a.

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Fairness through causal awareness: Learning causal latent-variable models for biased data. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 349–358, 2019b.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Vladimir A Marčenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, 1967.

Benjamin M Marlin and Richard S Zemel. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the third ACM conference on Recommender systems*, pages 5–12, 2009.

Charles H Martin and Michael W Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021.

Donald Martin Jr., Vinodkumar Prabhakaran, Jill Kuhlberg, Andrew Smart, and William S Isaac. Participatory problem formulation for fairer machine learning through community based system dynamics. *arXiv preprint arXiv:2005.07572*, 2020.

Charles Marx, Flavio Calmon, and Berk Ustun. Predictive multiplicity in classification. In *International Conference on Machine Learning*, pages 6765–6774. PMLR, 2020.

David A McAllester. Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

Liam McCoy, Jacquelyn Burkell, Dallas Card, Brent Davis, Judy Gichoya, Sophie LePage, and David Madras. On meaningful human control in high-stakes machine-human partnerships. *WeRobot*, 2020.

Reid McIlroy-Young, Siddhartha Sen, Jon Kleinberg, and Ashton Anderson. Aligning superhuman ai with human behavior: Chess as a model system. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1677–1687, 2020.

Katrina McLaughlin, Orla T Muldoon, and Marianne Moutray. Gender, gender roles and completion of nursing education: A longitudinal study. *Nurse education today*, 30(4):303–307, 2010.

Daniel McNamara, Cheng Soon Ong, and Robert C Williamson. Provably fair representations. *arXiv preprint arXiv:1710.04394*, 2017.

Andrew McStay. Emotional ai and edtech: serving the public good? *Learning, Media and Technology*, 45(3):270–283, 2020.

Nicolai Meinshausen. Causality from a distributional robustness point of view. In *2018 IEEE Data Science Workshop (DSW)*, pages 6–10. IEEE, 2018.

Prem Melville and Raymond J Mooney. Diverse ensembles for active learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 74. ACM, 2004.

Aditya Krishna Menon and Robert C Williamson. The cost of fairness in binary classification. In *Conference on Fairness, Accountability and Transparency*, pages 107–118, 2018.

Vahid Balazadeh Meresht, Abir De, Adish Singla, and Manuel Gomez-Rodriguez. Learning to switch between machines and humans. *arXiv preprint arXiv:2002.04258*, 2020.

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*, pages 230–239, 2019.

Jovana Mitrovic, Brian McWilliams, Jacob C Walker, Lars Holger Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. In *International Conference on Learning Representations*, 2020.

Eric Mjolsness and Dennis DeCoste. Machine learning for science: state of the art and future prospects. *science*, 293(5537):2051–2055, 2001.

Daniel Moyer, Shuyang Gao, Rob Brekelmans, Aram Galstyan, and Greg Ver Steeg. Invariant representations without adversarial training. *Advances in Neural Information Processing Systems*, 31, 2018.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, pages 7076–7087. PMLR, 2020.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.

Razieh Nabi and Ilya Shpitser. Fair inference on outcomes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2018.

Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. *Advances in neural information processing systems*, 29, 2016.

Hongseok Namkoong and John C Duchi. Variance-based regularization with convex objectives. *Advances in neural information processing systems*, 30, 2017.

Arvind Narayanan. Translation tutorial: 21 fairness definitions and their politics. In *Proc. Conf. Fairness Accountability Transp., New York, USA*, volume 1170, page 3, 2018.

Warwick J Nash, Tracy L Sellers, Simon R Talbot, Andrew J Cawthorn, and Wes B Ford. The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait. *Sea Fisheries Division, Technical Report*, 48, 1994.

Shamim Nemati, Andre Holder, Fereshteh Razmi, Matthew D Stanley, Gari D Clifford, and Timothy G Buchman. An interpretable machine learning model for accurate prediction of sepsis in the icu. *Critical care medicine*, 46(4):547, 2018.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2018.

Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

Nastaran Okati, Abir De, and Manuel Rodriguez. Differentiable learning under triage. *Advances in Neural Information Processing Systems*, 34, 2021.

Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

Luca Oneto, Michele Donini, Massimiliano Pontil, and Andreas Maurer. Learning fair and transferable representations with theoretical guarantees. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 30–39. IEEE, 2020.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pages 4970–4979. PMLR, 2019.

Nicolas Papernot and Patrick McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018.

Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfar Erlingsson. Scalable private learning with pate. In *International Conference on Learning Representations*, 2018.

Giambattista Parascandolo, Niki Kilbertus, Mateo Rojas-Carulla, and Bernhard Schölkopf. Learning independent causal mechanisms. In *International Conference on Machine Learning*, pages 4036–4044. PMLR, 2018.

Emilio Parrado-Hernández, Amiran Ambroladze, John Shawe-Taylor, and Shiliang Sun. Pac-bayes bounds with data dependent priors. *The Journal of Machine Learning Research*, 13(1):3507–3531, 2012.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Barak A Pearlmutter. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *International Conference on Machine Learning*, pages 7599–7609. PMLR, 2020.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. *Advances in neural information processing systems*, 30, 2017.

Marcelo OR Prates, Pedro H Avelar, and Luís C Lamb. Assessing gender bias in machine translation: a case study with google translate. *Neural Computing and Applications*, 32(10):6363–6381, 2020.

Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12556–12565, 2020.

Novi Quadrianto, Viktoriia Sharmanska, and Oliver Thomas. Discovering fair representations in the data domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8227–8236, 2019.

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

Maithra Raghu, Katy Blumer, Rory Sayres, Ziad Obermeyer, Bobby Kleinberg, Sendhil Mullainathan, and Jon Kleinberg. Direct uncertainty prediction for medical second opinions. In *International Conference on Machine Learning*, pages 5281–5290. PMLR, 2019b.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018.

Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358, 2019.

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

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

Manoel Horta Ribeiro, Raphael Ottoni, Robert West, Virgílio AF Almeida, and Wagner Meira Jr. Auditing radicalization pathways on youtube. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 131–141, 2020.

Horst WJ Rittel and Melvin M Webber. Dilemmas in a general theory of planning. *Policy sciences*, 4(2):155–169, 1973.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning, 2018.

Yuji Roh, Geon Heo, and Steven Euijong Whang. A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, 33 (4):1328–1347, 2019.

Yangjun Ruan, Yann Dubois, and Chris J Maddison. Optimal representations for covariate shift. In *International Conference on Learning Representations*, 2021.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Robb B Rutledge, Adam M Chekroud, and Quentin JM Huys. Machine learning and big data in psychiatry: toward clinical applications. *Current opinion in neurobiology*, 55:152–159, 2019.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2019.

Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Samira Samadi, Uthaipon Tantipongpipat, Jamie H Morgenstern, Mohit Singh, and Santosh Vempala. The price of fair pca: One extra dimension. In *Advances in Neural Information Processing Systems*, pages 10976–10987, 2018.

Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with gram matrices. In *International Conference on Machine Learning*, pages 8491–8501. PMLR, 2020.

Robert R Schaller. Moore's law: past, present and future. *IEEE spectrum*, 34(6):52–59, 1997.

Sebastian Schelter and Julia Stoyanovich. Taming technical bias in machine learning pipelines. *Bulletin of the Technical Committee on Data Engineering*, 43(4), 2020.

Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling up influence functions. *AAAI*, 2021.

Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural computation*, 4 (6):863–879, 1992.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

Peter Schulam and Suchi Saria. Auditing pointwise reliability subsequent to training. In *Proceedings of the 22nd International Conference on Ar-tificial Intelligence and Statistics (AISTATS)*, 2019.

Andrew D Selbst, Danah Boyd, Sorelle A Friedler, Suresh Venkatasubramanian, and Janet Vertesi. Fairness and abstraction in sociotechnical systems. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 59–68, 2019.

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. *arXiv preprint arXiv:1908.01755*, 2019.

Uri Shaham, Yutaro Yamada, and Sahand Negahban. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomputing*, 307:195–204, 2018.

Reza Shokri, Martin Strobel, and Yair Zick. On the privacy risks of model explanations. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 231–241, 2021.

Dylan Slack, Sorelle A Friedler, and Emile Givental. Fairness warnings and fair-maml: learning fairly with minimal data. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 200–209, 2020.

Jiaming Song, Pratyusha Kalluri, Aditya Grover, Shengjia Zhao, and Stefano Ermon. Learning controllable fair representations. In *AISTATS*, 2019.

Megha Srivastava, Tatsunori Hashimoto, and Percy Liang. Robustness to spurious correlations via human annotations. In *International Conference on Machine Learning*, pages 9109–9119. PMLR, 2020.

Harald Steck. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 713–722, 2010.

Ryan Steed and Aylin Caliskan. Image representations learned with unsupervised pre-training contain human-like biases. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 701–713, 2021.

Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. *Advances in neural information processing systems*, 30, 2017.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Harini Suresh and John V Guttag. A framework for understanding unintended consequences of machine learning. *arXiv preprint arXiv:1901.10002*, 2019.

Latanya Sweeney. Discrimination in online ad delivery. In *CoRR abs/1301.6822 (2013). http://arxiv.org/abs/1301.6822*, 2013.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

Ahmad P Tafti, Eric LaRose, Jonathan C Badger, Ross Kleiman, and Peggy Peissig. Machine learning-as-a-service and its application to medical informatics. In *International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 206–219. Springer, 2017.

Omer Tene and Jules Polonetsky. A theory of creepy: technology, privacy and shifting social norms. *Yale JL & Tech.*, 16:59, 2013.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018.

Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf, and Stefan Bauer. On disentangled representations learned from correlated data. In *International Conference on Machine Learning*, pages 10401–10412. PMLR, 2021.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

Thanasis Vafeiadis, Konstantinos I Diamantaras, George Sarigiannidis, and K Ch Chatzisavvas. A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*, 55:1–9, 2015.

Vladimir Vapnik, Esther Levin, and Yann Le Cun. Measuring the vc-dimension of a learning machine. *Neural computation*, 6(5):851–876, 1994.

Kush R Varshney and Homa Alemzadeh. On the safety of machine learning: Cyber-physical systems, decision sciences, and data products. *Big data*, 5(3):246–255, 2017.

Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations: Why and how to pass stress tests. *arXiv preprint arXiv:2106.00545*, 2021.

Rajeev Verma and Eric Nalisnick. Calibrated learning to defer with one-vs-all classifiers. *arXiv preprint arXiv:2202.03673*, 2022.

R Alan Walks and Larry S Bourne. Ghettos in canada's cities? racial segregation, ethnic enclaves and poverty concentration in canadian urban areas. *The Canadian Geographer/Le Géographe canadien*, 50(3):273–297, 2006.

Chuang Wang and Yue M Lu. Online learning for sparse pca in high dimensions: Exact dynamics and phase transitions. In *2016 IEEE Information Theory Workshop (ITW)*, pages 186–190. IEEE, 2016.

Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. Online object tracking with sparse prototypes. *IEEE transactions on image processing*, 22(1):314–325, 2012.

Xin Wang, Yujia Luo, Daniel Crankshaw, Alexey Tumanov, and Joseph E. Gonzalez. IDK Cascades: Fast Deep Learning by Learning not to Overthink. *Conference on Uncertainty in Artificial Intelligence*, June 2017. arXiv: 1706.00885.

Yixin Wang and Michael I Jordan. Desiderata for representation learning: A causal perspective. *arXiv preprint arXiv:2109.03795*, 2021.

Eugene P Wigner. Random matrices in physics. *SIAM review*, 9(1):1–23, 1967.

Bryan Wilder, Eric Horvitz, and Ece Kamar. Learning to complement humans. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 1526–1533, 2021.

Julia K Winkler, Christine Fink, Ferdinand Toberer, Alexander Enk, Teresa Deinlein, Rainer Hofmann-Wellenhof, Luc Thomas, Aimilios Lallas, Andreas Blum, Wilhelm Stolz, et al. Association between surgical skin markings in dermoscopic images and diagnostic performance of a deep learning convolutional neural network for melanoma recognition. *JAMA dermatology*, 155(10):1135–1141, 2019.

Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro. Learning non-discriminatory predictors. In *Conference on Learning Theory*, pages 1920–1953. PMLR, 2017.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Huanrui Yang, Jingyang Zhang, Hongliang Dong, Nathan Inkawhich, Andrew Gardner, Andrew Touchet, Wesley Wilkes, Heath Berry, and Hai Li. Dverge: diversifying vulnerabilities for enhanced robust generation of ensembles. *Advances in Neural Information Processing Systems*, 33:5505–5515, 2020.

Jitka Žáčková. On minimax solutions of stochastic linear programming problems. *Časopis pro pěstování matematiky*, 91(4):423–430, 1966.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1171–1180. International World Wide Web Conferences Steering Committee, 2017a.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970. PMLR, 2017b.

Meike Zehlike, Ke Yang, and Julia Stoyanovich. Fairness in ranking: A survey. *arXiv preprint arXiv:2103.14000*, 2021.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018.

Han Zhao and Geoffrey J Gordon. Inherent tradeoffs in learning fair representations. *Journal of Machine Learning Research*, 23:1–26, 2022.