Fairness in Artificial Intelligence based Decision Making

Thesis submitted in partial
fulfillment of the requirements of the degree of

Doctor of Philosophy
in
Computer Science and Engineering

by

P Manisha
20172145
manisha.padala@research.iiit.ac.in

Advised by Dr. Sujit P Gujar

International Institute of Information Technology
Hyderabad - 500 032, INDIA
May, 2023

International Institute of Information Technology

Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Fairness in Artificial Intelligence based Decision Making” by P Manisha, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Prof. Sujit P Gujar

Acknowledgments

I deeply owe to my research advisor, Dr. Sujit Gujar. I am grateful for his invaluable guidance, support, and encouragement throughout my PhD journey. His expert knowledge, wise counsel, and patient mentorship have been instrumental in helping me complete my research and write this thesis.

I would like to thank Prof T. E. S. Raghavan, Prof. Haris Aziz, Prof. C. V. Jawahar, Prof. Milind Tambe for providing their invaluable inputs and guidance. I would like to thank Dr. Ramasuri Narayanam and Dr. Aparna Taneja for being a wonderful mentors at Adobe Research and Google Research India. I would also like to thank my collaborators and colleagues Sankarshan Damle, Shaily Mishra, Samhita Kanaparthy and Debojit Das from IIIT Hyderabad for their hard work, constructive feedback, and ongoing support. I am grateful to all the faculty at IIIT Hyderabad and IIT Jodhpur for teaching me the fundamentals. I would also like to thank all the administrative members and staff who have helped me with the submission of the thesis and throughout my stay at IIIT.

I am grateful to Bharat Electronics Limited India for providing the financial and logistical support that made this work possible.

Most importantly, I want to express my heartfelt appreciation to my Mom (B Sreedevi) and Dad (P Mohan Rao) for believing in me and being there for me through all the hard times. I would like to thank all my friends for their love, encouragement, and understanding throughout this process. Finally, I thank everyone who have made my stay at IIIT Hyderabad one of the most pleasant experiences.

Thank you all for being a part of my academic journey.

Abstract

AI systems are ubiquitous in the current times, facilitating numerous real-world even real-time applications. Such sophistication is the consequence of advancement in algorithmic research and concurrent up-gradation of computational resources. The existing models achieve near-optimal results for specific performance measures. Such perfection is often obtained at the cost of *Fairness*. By fairness, we try to quantify the impact an application has on an individual user (*Individual Fairness*) or a group of users (*Group Fairness*). In this work, we shift our focus from a single performance measure and explore the fairness of existing algorithms specifically in two settings, i) Fair resource allocation with strategic agents and ii) Fair classification models. We divide our work and discuss it in the following two parts.

Part A – Fair Allocations with Strategic Agents. We consider the setting of resource allocation, where there are multiple items and multiple agents who have preferences for these items. The agents are rational and strategic and may manipulate their preferences to obtain higher gains. The social planner must find allocations that satisfy certain desirable fairness properties and are resistant to manipulation, i.e. ensure *strategy-proofnesss*. Researchers have proposed algorithms that charge agents in order to prevent manipulations. However, analytically designing payments which are fair and strategy-proof is challenging. In this part, we propose data-driven approach to learn payments that are fair and strategy-proof. We additionally consider resource allocation settings wherein charging payments is not feasible. We analyze the existence of strategy-proof algorithms that ensure

fair allocations. We consider certain well known fairness notions like envy-freeness, proportionality and max-min share allocations. Such notions only ensure individual fairness of the agents involved.

Part B – Fair Decisions for Groups. We consider machine learning-based classification algorithms. The accuracy of such algorithms has been the primary concern and is widely researched. More recently, researchers have uncovered the prejudiced predictions of such models towards certain demographic groups. Due to existing bias against certain race, gender or age, the data available is often biased. The prejudices in the data, amplified by the algorithms trained only for achieving higher accuracy, lead to unfair decisions to certain groups. Moreover, such algorithms made public on various online platforms potentially leak private information of the individual data used in training. Ensuring fairness and privacy in a machine learning framework gives rise to a non-convex and complex optimization with multiple constraints. Towards this, we rely on learning-based approaches and exploit neural networks' immense capacity to get closer to the goal.

Research Papers Based on the Thesis Work

Conference Papers

1. Padala Manisha and Sujit Gujar. “Mechanism Design without Money for Fair Allocations.” To Appear in the proceedings of the 20th IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology 2021 **(WI-IAT ’21)**

2. Padala Manisha, Sankarshan Damle, and Sujit Gujar. “Federated Learning Meets Fairness and Differential Privacy”. To Appear in the Proceedings 28th International Conference on Neural Information Processing of the Asia-Pacific Neural Network Society 2021 **(ICONIP’21)**.

3. Padala Manisha, Sujit Gujar. “FNNC: Achieving Fairness through Neural Networks.” In Proceedings of the Twenty-ninth International Joint Conference on Artificial Intelligence, 2020 **(IJCAI’20)**

4. Padala Manisha, and Sujit Gujar. “Thompson Sampling Based Multi-Armed-Bandit Mechanism Using Neural Networks.” Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, 2019 **(AAMAS’19)**

5. Padala Manisha, C. V. Jawahar, and Sujit Gujar. “Learning optimal redistribution mechanisms through neural networks.” Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, 2018 **(AAMAS’18)**

Peer-reviewed non-archival Publications

1. Padala Manisha, Sankarshan Damle, Sujit Gujar. “Building Ethical AI: Federated Learning meets Fairness and Privacy.” Conference First Indian Conference on Deployable AI, 2021. **Best Paper Award (DAI’21)**

Research Papers not in Thesis Work

Conference Papers

1. Sankarshan Damle, Padala Manisha, Sujit Gujar. "Combinatorial Civic Crowdfunding with Budgeted Agents: Welfare Optimality at Equilibrium and Optimal Deviation." In 37th Association for the Advancement of Artificial Intelligence 2023 **(AAAI'23)**

2. Shaily Mishra, Padala Manisha, Sujit Gujar. "Fair Allocation with Special Externalities." In The 19th Pacific Rim International Conferences on Artificial Intelligence 2022 **Best Paper Award Runner Up (PRICAI'22)**

3. Shaily Mishra, Padala Manisha, Sujit Gujar. "EEF1-NN: Efficient and EF1 Allocations Through Neural Networks." In The 19th Pacific Rim International Conferences on Artificial Intelligence 2022 **(PRICAI 2022)**

4. Samhita Kanaparthy, Padala Manisha, Sankarshan Damle, Sujit Gujar. "Fair Federated Learning for Heterogeneous Data." In Young Researcher's Symposium Track **(COMAD/CODS'22)**

5. Padala Manisha, Sankarshan Damle, and Sujit Gujar. "Learning Equilibrium Contributions in Multi-project Civic Crowdfunding." In the 20th IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology 2021 **(WI-IAT '21)**.

6. Padala Manisha, Debojit Das, Sujit Gujar. "Effect of Input Noise Dimension in GANs." In the 28th International Conference on Neural Information Processing **(ICONIP'21)**

Peer-reviewed non-archival Publications

1. Sankarshan Damle, Padala Manisha, Sujit Gujar. "Welfare Optimal Combinatorial Civic Crowdfunding with Budgeted Agents. **(GAIW@AAMAS '22)**

Contents

List of Figures

Figure Page

List of Tables

Table Page

Chapter 1

Introduction

> "If machine learning is our way into studying institutional decision making, fairness is the moral lens through which we examine those decisions." – Barocas et al. 2019 [24]

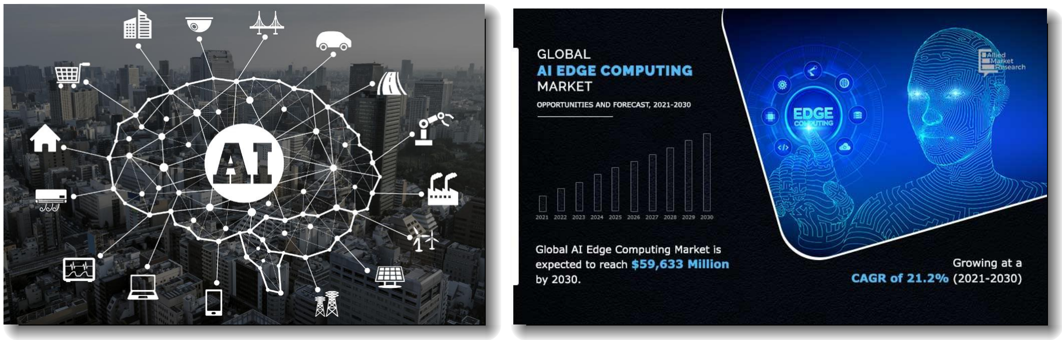

AI
GLOBAL
AI EDGE COMPUTING
MARKET
OPPORTUNITIES AND FORECAST, 2021-2030
2021 2022 2023 2024 2025 2026 2027 2028 2029 2030
Global AI Edge Computing Market is expected to reach $59,633 Million by 2030.
EDGE COMPUTING
Allied Market Research
Growing at a CAGR of 21.2% (2021-2030)

Figure 1.1: Ubiquitous AI systems [Image Credits: [7, 8]]

Artificial Intelligence (AI) has become a core component of technological innovation. AI is now ubiquitous in crucial sectors like automobile, health care, retail, finance, e-commerce and even entertainment. Many of these applications directly rely on the decisions from the AI model. As a result, AI-based decision making is having a huge impact on each of our lives. AI helps us decide what routes to chose while traveling via *Smart Navigation* apps,

Figure 1.2: Cake Cutting [Image Credits: [44]]

what products to buy via *Online Advertisements* and even what social media content to follow via *Recommendation Systems*. AI has a role in what we write (*Language Modelling*), how we look (*Photo Editors*) and even what opportunities we get (e.g., do we qualify for a job, a loan or medical treatment) via *Predictive Models*.

The global AI market was valued at USD 328.34 Billion in 2021 and will likely expand at a compound annual growth rate (CAGR) of 40.2% from 2021 to 2028 [7]. Such progression is possible due to intensive research and computational resources. In general, industries are driving the AI market towards faster and more efficient solutions to maximize revenue. As such, research towards improving certain established performance measures (e.g., the accuracy of a learned model), even marginally, is highly rewarded. In this "race" for perfection, we have significantly succeeded in narrow spaces. However, we still need holistic and continually productive solutions.

We believe that *inclusion* is the way to long-term solutions. Considering the interests of the user of any application is the first step towards inclusive thinking. In other words, a user must perceive that the solution offered is *fair* to it. Let us consider the classic example of cake cutting (Figure 1.2) and understand how appropriate methods can achieve a subjective notion like fairness. Consider a fancy cake with different layers and toppings which we must distribute between two people. Each person has different preferences for the different parts of the cake. In order to divide the cake fairly among the two, we ask one person to cut the cake and the other to choose. The person who cuts the cake ensures that he divides it so that either of the pieces is equally valuable to it. The second person will choose the piece which is more valuable than the other. Likewise, both persons feel they have received their fair share.

Resource Allocation is the classic setting where fairness has been extensively studied. The standard resource allocation model consists of agents interested in multiple yet limited resources. The social planner is responsible for designing appropriate allocations that satisfy specific efficiency and fairness criteria. There can be multiple aspects of fairness tailored to a specific resource allocation application. For example, it may not be fair to charge large payments to the agents in many auctions. Consider the auction of public property like spectrum auctions or government land, or factories. The main motive here is to find a suitable candidate for the said item and not profit.

On the one hand, the agents being *strategic* may lie about their valuations if the items are offered for free. On the other hand, charging large payments may discourage the participation of agents who genuinely value the property. In contrast, there are other situations, like crowdsourcing or online advertisement, where the social planner chooses suitable candidates to complete a specific task. It may not be fair that the planner pays an enormous amount to select an appropriate candidate in such scenarios. There is a notion of a fair share of allocation in other settings where there are no payments or incentives, like in the cake-cutting example. Each individual must be offered a fair share of the allocation, applicable in many web-based solutions for a course assignment, land allocation, or splitting the fare. In these settings, the mechanisms must ensure truthfulness without using payments.

The above applications warrant *individual fairness* of the agents or the social planner. In practice, situations may arise where a community or a population sub-group is subjected to a particular bias. Racial, gender-based, and even caste-based subjugation has been prevalent for a long time. It is no mystery that such prejudices are reflected in our gathered data. In recent times, with ease in accessibility of a wide variety of data and computational efficiency, learning algorithms have gained popularity. When trained on biased data, such learning algorithms definitely lead to biased predictions that are biased [25, 30, 53]. E.g., ProPublica conducted its study of the risk assessment tool, which was widely used by the judiciary system in the USA. ProPublica observed that the risk values for recidivism estimated for African-American defendants were, on average, higher than for Caucasian defendants. Hence, learning algorithms that are often trained to improve specific performance measures like accuracy must also incorporate notions of *group fairness*. Governments worldwide have adopted laws to enforce the same. E.g., the 80% Disparate Impact rule introduced in the US labor laws [77].

Naturally, any approach which ensures group fairness requires the knowledge of sensitive attributes (e.g., gender, race). These attributes often comprise the most critical information. The law regulations at various places prohibit using such attributes to develop models. The EU General Data Protection Regulation prevents the collection of sensitive user attributes [189]. Thus, it is imperative to address discrimination while preserving the leakage of sensitive attributes from the data samples. In other words, *privacy* of the data must not be violated in order to obtain high accuracy or even fairness. To ensure holistic and productive solutions, the evaluation of a model must depend on (i) primary performance measures (like accuracy), (ii) the fairness notion that it satisfies, and (iii) the privacy guarantees it provides. We provide some real-world applications and examine them through the lens of fairness, as discussed below.

1.1 AI Applications under Fairness Lens

Various applications use AI for decision-making. We discuss these in broadly two different categories of mechanism design and predictive modelling. In the prior, the agents are strategic, and hence appropriate incentives must be designed. In the latter, the agents are not strategic, and we are concerned with performance across groups of agents. Hence, we

ONLINE ADVERTISING
ONE PRICE FITS
ONLINE GOVERNMENT SURPLUS AUCTIONS NOVEMBER 8-16
- Airport Lost & Found Items
- Vehicles & Equipment
- Miscellaneous Items
auction.com
TNT AUCTION
Two Day Preview: Nov. 12-13
TAXI
Split Fare
Fairly split taxi fare, or the cost of an Uber or Lyft ride, when sharing a ride with friends.

Figure 1.3: Mechanism design is applied in auctions, dynamic pricing, crowdsourcing, sharing resources and online-advertisements [Image Credits: [12]]

divide the applications in two parts based on the type of fairness studied, **Part A** – Fair Allocations with Strategic Agents., we discuss certain applications of individual fairness in resource allocation. In **Part B** – Fair Decisions for Groups., we discuss group fair and private classification using neural network based classifiers.

1.1.1 Part A – Fair Allocations with Strategic Agents

We often face the problem of allocating public resources/objects among multiple agents who desire them. These strategic agents have their private values or preferences over the resources. The agents may manipulate their preferences to obtain favourable outcomes, even more so with AI-based agents. The allocations must satisfy specific desirable game-theoretic properties. *Mechanism design* is a field in economics and game theory that designs economic mechanisms or incentives toward desired objectives in strategic settings where players act rationally. It is widely applicable in a lot of complex settings (Figure 1.3), as discussed below,

Figure 1.4: Mechanism Design Fairness Challenges [Image Credits: [12]]

- *Auctions* are widely used in e-commerce and by government organizations. They provide a natural setting for mechanism design with money. In auctions, various agents bid for resources (like property, frequency spectrum, airport slots, etc.). The social planner decides an appropriate allocation and payment scheme for certain objectives. The objective could be maximizing revenue or social welfare. There is a significant increase in web-based auctions that require designing mechanisms for large and complex settings.
- *Internet advertisement* has become a million-dollar market in the current times. In various search engines like Google or Bing, when a user enters a keyword, multiple links related to the keyword are displayed. Alongside, various sponsored links that correspond to the advertisements of selected advertisers are also displayed. The user is directed to the corresponding page when such a link is clicked. The advertiser is charged a certain amount for directing the user to its page. Generally, these search engines employ Mutli-armed based (MAB) mechanism design [16, 116, 175]. These mechanisms select suitable advertisements to display (based on clicks obtained) and conduct an auction to determine the payments they should charge for the slots.
- *Expertsourcing* is a popular method for requesters to obtain information or collect opinions from a large group of people. It enables a crowd with plenty of diverse expertise to contribute to any outsourced task. Tasks include rating products online, testing applications for a company, or collecting real-world data. Many of these

tasks do not have answers, i.e., we do not know the ground truth and we cannot verify the correctness of the players' contributions. Due to this fact, strategic players may have incentives to manipulate the system by providing arbitrary data without actually performing the tasks. Researchers propose several incentive mechanisms to discourage such undesirable strategies based on specific reward schemes [87, 133].

- *Dynamic Pricing* has become an integral part of various applications to help adjust to market conditions. We commonly find it in ride-sharing in Uber and other taxi services, airline and hotel bookings, and various e-commerce stores. With an increase in online shopping, market conditions have become highly volatile. The pricing is already adjusted based on competitive pricing, demand, supply, sales, and requirements. The question is whether the pricing is perceived as fair by both the supplier and the customer [65].
- *Sharing Resources* is applicable in real-world settings, such as division of investments and inheritance, vaccines, or tasks. Web-based applications such as Spliddit, The Fair Proposals System, Coursematch, Divide Your Rent Fairly, among others, are used for credit assignment, land allocation, division of property, course allocation, and even task allotment. All these applications assure certain fairness and efficiency guarantees.

Money is involved in many applications discussed above, like auctions and dynamic pricing. Hence, mechanism design focuses on designing payment schemes to achieve the objective. Often these objectives are aggregated statistics like overall welfare or revenue generated. Consequently, the mechanisms' solutions may be perceived as unfair by each individual agent involved. E.g., surge pricing in Uber, high prices in auctions for giveaway goods, and very high payment for sourcing quality workers (Figure 1.4). On the other hand, there is a host of applications of resource division where money is not involved. E.g., division of inheritance, splitting room rent among the members, and distribution of central revenue among the states. In these scenarios, it is impossible to introduce payments to

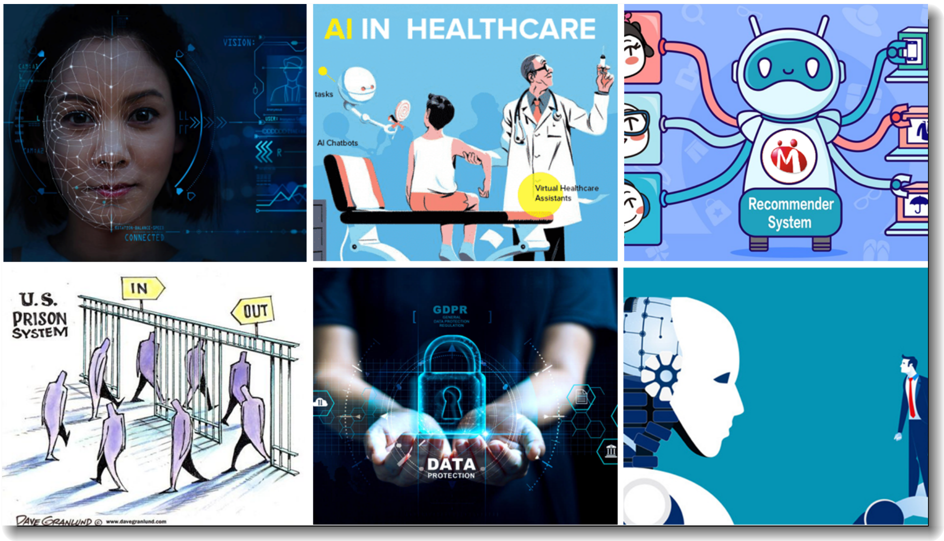


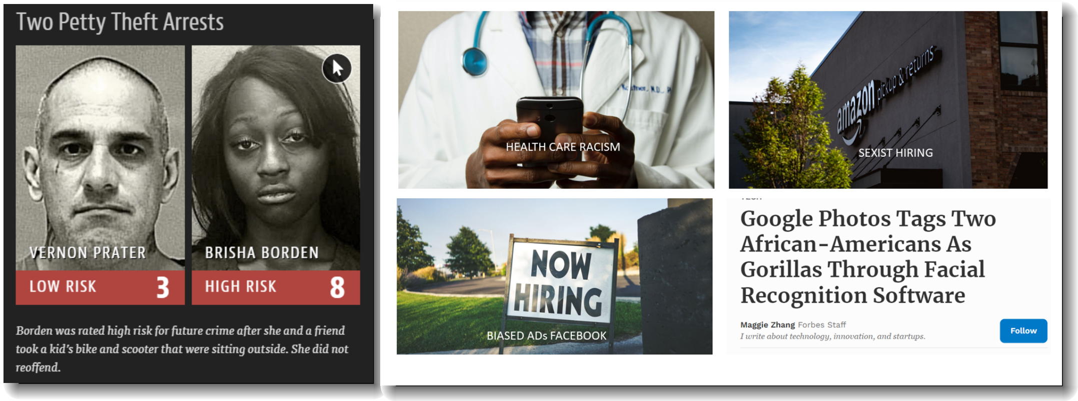
Figure 1.5: Machine Learning Applications in Health Care, Reccomender Systems, Recruitment, Criminal Justice, Privacy, Recognition [Image Credits: [12]]

incentivize strategic agents. Much of the existing literature focuses on the fair division of resources but does not consider strategic agents. We believe that when agents act selfishly, having fair algorithms would still lead to unfair solutions (Figure 1.4).

1.1.2 Part B – Fair Decisions for Groups

In recent years machine learning models have been popularized as prediction models to supplement the process of decision-making. Decision-making in crucial sectors like recruitment, criminal justice, and even health care (Figure 1.5). We list some use cases below,

- *Recruitment.* Many companies use machine learning models in the process of hiring. Machine learning models assist in the entire process, from predicting hiring needs to candidate assessment. Even displaying job advertisements strategically is also decided using previous data. Often the goal of such companies is increasing short-term profit.



Borden was rated high risk for future crime after she and a friend took a kid's bike and scooter that were sitting outside. She did not reoffend.

Google Photos Tags Two African-Americans As Gorillas Through Facial Recognition Software

Maggie Zhang Forbes Staff
I write about technology, innovation, and startups.

Figure 1.6: Bias in Machine Learning [Image Credits: [34, 35]]

They look for the best candidates with the highest performance based on certain features, including sensitive information. Often these models are trained on biased data, which reflect in the predictions and hence find their way into the decision-making.

- *Criminal Justice.* AI-based predictive models have found their way into criminology, law, and forensics. Algorithms provide decisions regarding bail, sentence and parole, and risk assessments. Such models play a crucial role in the prisoners' lives; incorrect decisions could also immediately affect the lives of many others.
- *Healthcare.* In healthcare, there is always a contest for limited and costly resources. In healthcare, machine learning is used to develop better diagnostic tools, predict the side effects of new drugs, schedule appointments, and even prioritize the patients for getting certain treatments. Invariably these models are trained to produce the most effective results for minimum costs. Although, we must ask if the predictions are biased towards certain population subgroups, causing a further rift in society.
- *Recommender Systems.* Recommender systems have played a vital role in our lives with the rise of Youtube, Amazon, and Netflix. In e-commerce, these systems suggest to buyers articles that could interest them. In online advertising, they suggest to users the right content which matches their preferences. These systems aim to recommend relevant

items to the users based on the users' profiles. The efficiency of a recommender system is often the major focus. However, we ignore the bias it creates via unfair recommendations. Also, sensitive data used by these algorithms may lead to privacy issues.

- *Security and Privacy.* Many security software depend on facial recognition and identification. While these algorithms must have good overall performance, they should also perform equally well across different demographics of people. Unless this is ensured, security systems unethically become biased towards certain groups of people. Likewise, the data used for training machine learning algorithms contain sensitive information. Hence privacy guarantees of these applications must be considered before deployment.

These machine learning models unknowingly introduce a societal bias through their predictions [25, 30, 53] (Figure 1.6). E.g., ProPublica conducted its study of the risk assessment tool, which was widely used by the judiciary system in the USA. ProPublica observed that the risk values for recidivism estimated for African-American defendants were, on average, higher than for Caucasian defendants (Figure 1.6). More recently, several examples indicate the disparity in the hiring process, even at Amazon and Facebook. Some examples also highlight the disparity in income levels among different genders and health care opportunities for different races. Models trained on historically prejudiced data often propagate and amplify the existing bias.

1.2 Contributions

It is common to consider and optimize over a single performance measure in any AI-based solution. Here we shift our focus from a single performance measure and consider fairness while designing algorithms. Such inclusion comes with significant challenges, calling for a customized approach for each application. The process majorly involves the following steps. (i) The first step is quantifying the notion of *Fairness* or identifying what is a fair solution in the context of an application. (ii) Then, we assess the deployed system in terms of the fairness notion. (iii) We next identify the challenges involved in ensuring fairness. (iv) Finally, we propose modifying the existing system to ensure a fairer solution. We now discuss five different applications where we show that the current solutions can be improved and made fairer to an individual or a group of individuals.

1.2.1 Part A – Fair Allocations with Strategic Agents

We summarize our contributions in three applications of mechanism design consisting of mechanisms with and without payments.

1.2.1.1 Redistribution Mechanism

We consider a social setting where public resources are allocated among competing and strategic agents to maximize social welfare (the objects should be allocated to those who value them the most). This setting is called *Allocative Efficiency* (AE). We need the agents to report their valuations for obtaining these resources, truthfully referred to as *Dominant Strategy Incentive Compatibility* (DSIC).

Example 1: Government Auctions

The Government often acquires old abandoned buildings, factories, and lands which do not belong to anyone. The Government then finds owners for these structures to ensure their better utilization. The aim of the Government is not to earn money but find optimal welfare allocation of these resources among the different interested parties. In many cases, the interested parties may try to manipulate their valuations to obtain these resources. Redistribution mechanisms help us prevent such malpractices while charging minimum money.

Challenges. Typically, we use auction-based mechanisms to achieve AE and DSIC. However, due to Green-Laffont Impossibility Theorem [93], we cannot ensure budget balance in the system while ensuring AE and DSIC. That is, the net transfer of money cannot be zero. Hence agents might end up paying large amounts of money to convince the social planner of the value of the resources. This problem has been addressed by designing a "redistribution" mechanism to ensure a minimum surplus of money, allocative efficiency and incentive compatibility. The objective could be minimizing surplus in expectation (or worst case). The objects can be homogeneous or heterogeneous. Designing such mechanisms is non-trivial. More concretely, designing redistribution mechanisms that perform well in expectation becomes analytically challenging for heterogeneous settings [136].

Contributions

- We train a neural network to determine an optimal redistribution mechanism for the given settings with both objectives, optimal in expectation and in the worst case.
- We also propose a loss function to train a neural network to optimize the worst case.
- We design neural networks with the underlying rebate functions being linear and nonlinear in terms of agents' bids.
- We demonstrate that our networks' performances mimic theoretical guarantees. We observe that a neural network based redistribution mechanism for homogeneous settings which uses nonlinear rebate functions outperforms linear rebate functions when the objective is optimal in expectation.
- We also show that our approach yields an optimal in expectation redistribution mechanism for heterogeneous settings.

1.2.1.2 MAB Mechanisms

Similar to the above setting, we have multiple resources and bidders although the value the bidders derive from the resources is stochastic. Consider expertsourcing, a popular method for requesters to obtain information or collect opinions from a large group of agents. The actual quality of the agent is known to neither the agent nor the auctioneer. Such parameters are not deterministic but are subject to various environmental conditions, hence are stochastic or could even be adversarial. In such a setting, it becomes necessary to figure out the average values of these parameters through exploration and at the same time ensure that the agents do not misreport their cost. *Multi-Armed-Bandit* algorithms are learning based algorithms, they aim to minimize the difference between the qualities of the best possible agents and the agents selected, also referred to as *regret*. However, in the presence of such learning algorithms, the strategic agents have more freedom to manipulate. Hence it is required to design novel mechanisms that also learn the environmental parameters. Such mechanisms are referred to as *Multi-Armed-Bandit (MAB) Mechanisms*.

Example 2: Online Advertisements

Online Advertisements have become a major source of income for various search engines like Google, Bing etc. Interested parties bid for certain slots on the websites to display ads. They are then charged according to their slots of choice, while they obtain a fixed value for every click their ad receives. The goal is to select bidders offering competitive prices while maximizing the number of clicks. Often the utilities obtained by the advertisers vary a lot, depending on the payment scheme used. Using multi-armed bandits based mechanisms, we could learn payments schemes which maximize revenue, and ensure utilities with reduced variance

Challenges. Most of the MAB mechanisms focus on frequentist approaches like upper confidence bound algorithms. Recent work shows that Bayesian approaches like *Thompson sampling* ensure lower regret. The resulting mechanism satisfies a weaker game theoretic property, namely, *Within-Period Dominant Strategy Incentive Compatibility* (WP-DSIC). The existing payment rules in the Thompson sampling based mechanisms may be unfair to the auctioneer causing a negative utility. Besides, if we wish to minimize the cost to the auctioneer, it is challenging to design payment rules that satisfy WP-DSIC while learning through Thompson sampling [135].

Contributions.

- We propose to use a data-driven approach for designing MAB mechanisms.
- Specifically, we use neural networks for designing a WP-DSIC payment rule, while the allocation rule is modelled using Thompson sampling.
- For the setting of crowd-sourcing for recruiting quality workers, our results indicate that the learned payment rule guarantees better revenue while maximizing the social welfare and also ensuring reduced variance in the utilities to the agents.

1.2.1.3 Fair Division of Resources.

Fairness is well studied in the context of resource allocation.

Example 3: Division of Inheritance

Division of inheritance often leads to major rifts among the family members. Dividing the inheritance in a way that satisfies each member often helps in reducing ill-feeling. Moreover, every member may have different values for different items, which are often private. Since it is not possible to charge each member to elicit their true valuations, it is important to study mechanisms that ensure fair division while truthful reporting of valuations.

Challenges. Researchers have proposed different fairness notions primarily *envy-freeness* (EF), and its relaxations, *proportionality* and *max-min share* (MMS). There is a vast literature on the existential and computational aspects of such notions. While computing fair allocations, any algorithm assumes agents' truthful reporting of their valuations towards the resources. Whereas in real-world web-based applications for fair division, the agents involved are strategic and may manipulate for individual utility gain.

Contributions.

- In our work [159], we study DSIC mechanisms also referred to as strategy-proof mechanisms without monetary transfer, which satisfy the various fairness criteria.
- We know that for additive valuations, designing truthful mechanisms for EF, MMS and proportionality is impossible. Here we show that there cannot be a truthful mechanism for EFX and the existing algorithms for EF1 are manipulable.
- We then study the special case of single-minded agents. For this case, we provide a *Serial Dictatorship Mechanism* that is DSIC and satisfies all the fairness criteria except EF.

In two out of the three applications we discussed, we employed learning based approaches to ensure fairness to the agents or the social planner involved in the resource allocation. Till now, we have considered different types of individual fairness. In the next part of our work, we further investigate other group fairness and related biases introduced by general machine learning algorithms.

1.2.2 Part B – Fair Decisions for Groups.

We summarize our contributions towards building applications which require group fairness and privacy guarantees.

1.2.2.1 FNNC: Fair Neural Network Classifier

Classification algorithms are used in decision making but could often suffer from unfair predictions.

Example 4: Loan Approval

Loans are the core business of banks. The loan companies grant a loan after an intensive process of verification and validation. However, they still are doubtful if the applicant is able to repay the loan with no difficulties. Machine learning is used to predict whether an applicant is able to repay the loan. Often due to existing biases in the data, applicants belonging to a minority race and/or gender are often not approved for loans. Hence, a fair classifier is needed that does not discriminate based on attributes like gender/race.

We focus on existing fairness notions widely studied in the community which comply with various fairness laws in place. The authors in [202] propose *group fairness*, which requires different sensitive groups to receive beneficial outcomes in similar proportions. We are concerned with group fairness like: *Demographic Parity* (DP) [71], *Disparate Impact* (DI) [77] and *Equalized odds* (EO) [105].

Challenges. We note that achieving a perfectly unbiased model is impossible [53]. Hence various approaches minimize the bias while maintaining high accuracy [5, 36, 132, 158]. In classification models, fairness can be ensured by solving a constrained optimization problem. In this work [158], we focus on fairness constraints like Disparate Impact, Demographic Parity, and Equalized Odds, which are non-decomposable and non-convex. Researchers define convex surrogates of the constraints and then apply convex optimization frameworks to obtain fair classifiers. Surrogates serve as an upper bound to the actual constraints, and convexifying fairness constraints is challenging.

Contributions.

- We propose a neural network-based framework, FNNC, to achieve fairness while maintaining high accuracy in classification. The above fairness constraints are included in the loss using Lagrangian multipliers. The network is optimized using two-step mini-batch stochastic gradient descent.
- We prove bounds on generalization errors for the constrained losses which asymptotically go to zero.

- Our experiments show that FNNC outperforms the state-of-the-art. The experimental evidence supplements our theoretical guarantees.
- In summary, we have an automated solution to achieve fairness in classification, which is easily extendable to many fairness constraints.

1.2.2.2 Towards Building Ethical AI – Fair and Private Classifier

In the previous work we consider fairness in a single neural network model. In this work, [157] we consider both privacy and fairness in a general training setting for large datasets. For handling big datasets and pertaining computational challenges, researchers have proposed *distributed* training process, referred to as *Federated learning* (FL) [139].

Example 5: Face Recognition in Mobile Phones

Facial recognition is used by almost everyone today to unlock their phones. To make robust predictions, federated learning is used since sharing the images violates privacy. While the data is not shared, it is still possible to reconstruct the input image given the trained model hence, one must provide differential privacy guarantees. Moreover, the software should perform equally well across different genders, races and age groups. Hence there must be both privacy and fairness guarantees.

Challenges. Invariably all approaches guaranteeing fairness require the information of the sensitive attribute. This information comprises any individual information based on the training data [1] and any information related to the sensitive attribute [189]. Typically, the law regulations at various places prohibit using such attributes to develop models such as EU General Data Protection Regulation prevents the collection of protected user attributes. The aggregator in FL has no direct access to private data, which prima facie preserves privacy. However, there exist several attacks that highlight the information leak in an FL setting. To address privacy concerns existing literature either uses cryptographic solutions based mainly on complex *Partial Homomorphic Encryption* (PHE) or through

Differential Privacy (DP). While private FL solutions using PHE exist in the literature [76, 144, 200, 204], these suffer from computational inefficiency and post-processing attacks.

Contributions.

- We propose a framework that ensures predictions that are socially *fair* towards all demographic groups even when trained on imbalanced data and preserves the privacy of (often) delicate individual information present in the dataset.
- To this end, we use the rigorous privacy guarantees provided by a *differentially-private* solution [154, 162, 177, 196].
- We demonstrate the trade-off between accuracy, fairness and privacy obtained while using our approach on various real-world datasets.

1.3 Organisation of the Thesis

As discussed above, we divide the work into two parts, Part A and Part B.

1 In Part A, we establish the preliminaries for game theory, mechanism design and automated mechanism design in Chapter 2. We discuss our work on redistribution mechanism in Chapter 3. Next, we discuss our work on Thompson sampling based mechanism design in Chapter 4 Finally we discuss fair resource allocation under strategic agents in Chapter 5.

2 Next in Part B, Chapter 6, we discuss preliminaries and definitions related to fairness in machine learning and differential privacy. In Chapter 7, we discuss our proposed Fair Neural Network Classifier (FNNC) and Fair and Private Federated Learning framework (FPFL) in Chapter 8.

Part A – Fair Allocations with Strategic Agents

Chapter 2

Part A - Preliminaries and Related Work

Game Theory models strategic interactions between rational and intelligent players. The aim is to predict the outcome of the interaction between the players who are maximizing their individualistic payoff. There are various real-world applications where the desired outcome is either overall efficiency or revenue obtained. Along with these properties, the fairness of the outcome is also well-studied. In game theory, various equilibrium analyses characterize the outcome and its properties. On the other hand, we are more concerned with *Mechanism Design* or reverse-engineered game theory. In mechanism design, the social planner specifies the interactions among self-interested players for a desirable outcome. Typically the social planner either incentivizes them (mechanisms with money) or designs appropriate rules (mechanisms without money) for the players to act accordingly. In this chapter, we discuss the preliminaries of game theory and mechanism design, especially for the setting of resource allocation.

2.1 Game Theory

Game theory models the interaction between rational players. Each player is capable of and intends to maximize its pay-off. The players' interests may be conflicting or cooperative. Game theory provides us with appropriate tools to predict the outcome of such interactions. A prevalent approach used to represent a game is the *Strategic Form*.

Definition 2.1 (*Strategic Form Game*)**.** *A strategic form game* Γ *is a tuple* $\langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$ *where,*

- $N = \{1, 2, \ldots, n\}$ *is the set of players*
- $S_1, S_2, \ldots, S_n$ *are the strategies of the players*
- $u_i : S_1 \times S_2 \times \ldots \times S_n \to \mathbb{R}$ *for* $i \in N$ *are the utility functions*

The players all select their strategies simultaneously in the strategic form game and report this to the social planner. The planner then computes the outcome and individual utilities. Below we provide an example of a commonly studied game between the two prisoners, also known as the *Prisoner's Dilemma*.

Example 6: Prisoner's Dilemma

In this problem, there are two prisoners $N = \{1, 2\}$. The prosecutors have no evidence to convict them. However, the prosecutors question each of the prisoners separately to obtain a confession. The prisoners cannot communicate with each other and are offered the following choices,

- If both confess they will each receive five-year imprisonment
- If prisoner 1 confesses and prisoner 2 defects then, prisoner 1 gets 1 year imprisonment and 2 gets ten years
- If prisoner 2 confesses and prisoner 1 defects then, prisoner 2 gets 1 year imprisonment and 1 gets ten years
- If both defect they get two years imprisonment each

$P2$ $P1$	D	C
D	(-2, -2)	(-10, -1)
C	(-1, -10)	(-5, -5)

Table 2.1: Player Utilities for Prisoner's Dilemma

We obtain the following payoff matrix for the prisoners ($P1$ and $P2$) from the above. The row player is $P1$, and the first value corresponds to its utility, while $P2$ is the column player, and the second value corresponds to its utility. C denotes the prisoner's choice to confess, and D denotes the prisoner's choice to defect. The representation in Table 2.1 is also called the matrix form of the game. Given the game of prisoner's dilemma, what would the prisoners choose and hence what would be the outcome? This question leads us to the various equilibrium concepts present in game theory.

Given a game Γ, players may choose different strategies based on the utility. Here we define two different equilibrium strategies for the players, *Dominant Strategy Equilibrium* and *Nash Equilibrium*. A strategy maximizing the utility of a player irrespective of the strategy of the other players is the dominant strategy. A dominant strategy equilibrium is the dominant strategy profile of all players. Any rational, intelligent players choose to play the dominant strategy if such an equilibrium exists. Such a strategy may not exist; hence Nash proposed a weaker notion of equilibrium referred to as the Nash equilibrium. A strategy profile is Nash equilibrium if every player following the strategy maximizes their utility, given that every other player also follows the given strategy. In the following definitions, we consider n players having a strategy profile $(s_1, \ldots, s_n)$ where s_{-i} denotes the strategy tuple of all the players except player i. More formally,

Definition 2.2 (*Dominant Strategy Equilibrium*)**.** *Given a game* Γ *a strategy profile* $(s_1^*, \ldots, s_n^*)$ *called a dominant strategy equilibrium if* $\forall i \in N$*, the strategy* s_i^* *is a dominant strategy,*

$$u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i}), \forall s_i \in S_i,\ \ s_i \neq s_i^* \text{ and } s_{-i} \in S_{-i}$$

while it is $u_i(s_i^*, s_{-i}) > u_i(s_i, s_{-i})$ *for some* $s_{-i} \in S_{-i}$*. This is also referred to as weakly dominant strategy equilibrium. A strict inequality for all* s_i *in the above equation would correspond to a strongly dominant strategy equilibrium.*

Definition 2.3 (*Nash Equilibrium*)**.** *Given a game* Γ*, a strategy profile* $(s_1^*, \ldots, s_n^*)$ *is called a Nash equilibrium if* $\forall i \in N$*, the strategy* s_i^* *satisfies,*

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*), \forall s_i \in S_i$$

In the prisoner's dilemma game with pay-off matrix as given by Table 2.1, note that choosing D is strongly dominated by C for $P1$ since,

$$u_1(C, D) > u_1(D, D) \text{ and } u_1(C, C) > u_1(D, C)$$

Similarly for $P2$ it is,

$$u_2(D, C) > u_2(D, D) \text{ and } u_2(C, C) > u_2(C, D)$$

Thus (C, C) is a strongly dominant strategy. Further, it is also the unique Nash equilibrium. Although (C, C) is the natural prediction, (D, D) is the best outcome jointly for the players. Hence, the dominant strategy outcome does not maximize the players' overall welfare or the sum of utilities. Equipped with the essential game-theoretic tools, we try to model real-world situations similarly to obtain desirable outcomes. Generally, in many situations, decisions are made considering the interests of a group of people, especially in public decision-making and resources/task allocation in any organization. Here we consider all the people involved as players and enlist the set of possible outcomes for the given situation. The players harbour different interests in these outcomes. The social planner is tasked with designing a game among these self-interested players by designing specific rules/incentives. Using game theory, we can now analyze each player's strategies and, thus, the outcome. The social planner must design appropriate rules to facilitate a conducive outcome. Such reverse engineering of game theory is called *Mechanism Design*.

Specifically, consider the resource allocation setting, where there are multiple players and multiple but finite items. Every player has a specific private valuation for these items. The social planner wants to find the optimal outcome or, in this case, the optimal allocation. An allocation is considered optimal only when it satisfies certain fairness and/or efficiency criteria. There are two underlying problems; the first problem is of *preference elicitation* where the social planer must design a game as discussed above such that the true valuation of the players is revealed. Typically, the planner either provides monetary incentives (mechanisms with money) or poses certain conditions (mechanisms without money) to enable favourable interactions. In other words, at equilibrium, the players report their true valuations. The second problem is of *preference aggregation* or finding the optimal allocation given the player's valuations. We discuss the preliminaries in two parts, Part I - Mechanism Design with Money and Part II - Mechanism Design without Money. In both cases, we also discuss the fairness and efficiency properties we wish the mechanism to satisfy. We first discuss the essential mechanism design environment required to model the resource allocation problem as a game.

2.2 The Mechanism Design Environment

The following provides a general setting for formulating, analyzing, and solving mechanism design problems.

- There are n players denoted by the set $N = \{1, 2, \ldots, n\}$. The players are rational and intelligent

- X is a set of *alternatives* or *outcomes*. The players are required to make a collective choice from the set X
- Players have privately observed preferences over the alternatives in X. That is, the preference player i derives is determined by the privately observed signal or type θ_i
- We denote by Θ_i the set of private values of player i, $i = 1, 2, \ \ldots, \ n$. The set of all type profiles is given by $\Theta = \Theta_1 \times \ \ldots \ \ \times \Theta_n$. A typical type profile is represented as $\theta = (\theta_1, \ \ldots, \ \theta_n)$
- The private values of the players have a common prior distribution $\Phi \in \Delta(\Theta)$
- A player's preference over the outcomes is represented by a utility function $u_i : X \times \Theta_i \rightarrow \mathbb{R}$. Given $x \in X$ and $\theta_i \in \Theta_i$, the value $u_i(x, \theta_i)$ denotes the payoff that player i having type $\theta_i \in \Theta_i$ receives from a decision $x \in X$
- The set of outcomes X, the set of players N, the type sets Θ_i $(i = 1, \cdots, n)$, the common prior distribution $\Phi \in \Delta(\Theta)$, and the payoff functions u_i $(i = 1, \cdots, n)$ are assumed to be *common knowledge* among all the players. The specific value θ_i observed by player i is private information of player i

Social Choice Functions

The function that makes a collective decision based on the player's private valuations is formally defined as,

Definition 2.4 (*Social Choice Function* (SCF))**.** *Given a set of players $N = \{1, 2, \ldots, n\}$, their type sets $\Theta_1, \Theta_2, \ldots, \Theta_n$, and a set of outcomes X, a social choice function is a mapping*

$$f : \Theta_1 \times \cdots \times \Theta_n \rightarrow X$$

that assigns to each possible type profile $(\theta_1, \theta_2, \ldots, \theta_n)$ a collective choice from the set of alternatives.

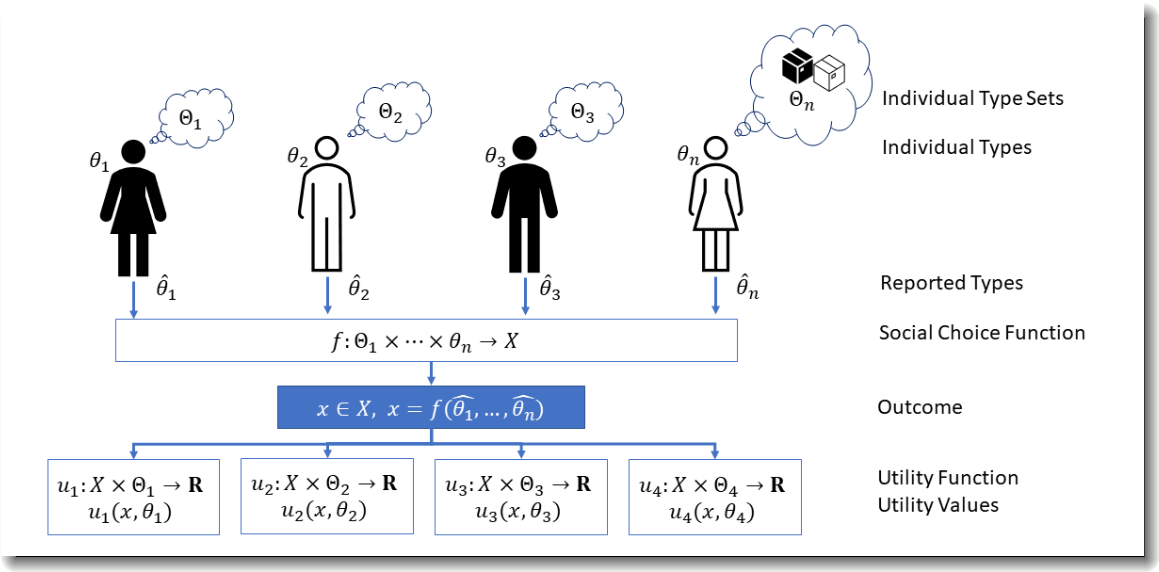


Figure 2.1: Mechanism Design Environment

Preference Elicitation Problem

Given a social choice function, $f : \Theta_1 \times \ldots \times \Theta_n \to X$, the players may chose to misreport their true individual types $\theta_1, \cdots, \theta_n$. This is because they are rational and aim to maximize their utilities. The problem of ensuring that it is in the player's best interest to reveal its true types is called the *preference elicitation* problem or the *information revelation* problem.

Preference Aggregation Problem

Let θ_i be the true type and $\hat{\theta}_i$ the reported type of agent i $(i = 1, \ldots, n)$. The process of computing $f(\hat{\theta}_1, \ldots, \hat{\theta}_n)$ is called the *preference aggregation* problem. The preference aggregation problem is usually an optimization problem.

Figure 2.1 provides a pictorial representation of all the elements making up the mechanism design environment.

Direct and Indirect Mechanisms

In a broad sense, mechanism design models incomplete optimization problems where the parameters are partially known. Therefore the first step is eliciting the unknown values, i.e., the type information. For truthful elicitation, there are broadly two kinds of mechanisms, *indirect mechanisms* and *direct mechanisms*. We define these below. In these definitions, we assume that the set of players N, the set of outcomes X, the sets of types $\Theta_1, \ldots, \Theta_n$, a common prior $\Phi \in \Delta(\Theta)$, and the utility functions $u_i : X \times \Theta_i \to \mathbb{R}$ are given and are common knowledge.

Definition 2.5 (*Direct Mechanism*)**.** *Given a social choice function* $f : \Theta_1 \times \Theta_2 \times \ldots \times \Theta_n \to X$*, a direct (revelation) mechanism consists of the tuple* $(\Theta_1, \Theta_2, \ldots, \Theta_n, f(.))$.

The idea of a direct mechanism is to *directly* seek the type information from the players by asking them to reveal their true types.

Definition 2.6 (*Indirect Mechanism*)**.** *An indirect (revelation) mechanism consists of a tuple* $(S_1, S_2, \ldots, S_n, g(.))$ *where* S_i *is a set of possible actions for player* i *(*$i = 1, 2, \ldots, n$*) and* $g : S_1 \times S_2 \times \ldots \times S_n \to X$ *is a function that maps each action profile to an outcome.*

The idea of an indirect mechanism is to provide a choice of actions to each player and specify an outcome for each action profile. Thus inducing a game among the players and the strategies played by the players in an equilibrium of this game will indirectly reflect their original types. More formally, the mechanism induces a Bayesian game formally defined below,

Definition 2.7 (*Bayesian Game*)**.** *Given a set of players* N*, with types* $(\Theta_1, \ldots, \Theta_n)$ *having a common prior* $\phi \sim \Delta(\Theta)$ *and a set of outcomes* X*. If each player* i *has an utility* $u_i : X \times \Theta_i \to \mathbb{R}$*, then a mechanism* $M = (S_1, \ldots, S_n, g(.))$ *induces a Bayesian game* $\Gamma^b : (N, (\Theta_i), (S_i), (p_i), (U_i))$ *among the players where,* $U_i(\theta_1, \ \ldots, \ \theta_n, s_1, \ \ldots, \ s_n) = u_i(g(s_1, \ \ldots, \ s_n), \theta_i)$

Given the basic components, we discuss certain properties that the mechanism must satisfy in the next section. Firstly we are concerned with the truthfulness of a mechanism for preference elicitation. Secondly, we also look for certain efficiency and fairness criteria it must satisfy while preference aggregation.

2.3 Properties of a Mechanism

We have already seen that mechanism design involves preference revelation (or elicitation) and aggregation problems. For truthful elicitation, there is a need to make true revelation the best response for the players, consistent with rationality and intelligence assumptions. Offering incentives is a way of doing this; incentive compatibility essentially refers to offering the right amount of incentive to induce truth revelation by the players.

2.3.1 Incentive Compatibility

There are broadly two types of incentive compatibility: (1) Truth revelation is the best response for each player irrespective of what is reported by the other players; (2) Truth revelation is the best response for each player whenever the other players also reveal their true types. The first one is called dominant strategy incentive compatibility (DSIC), and the second one is called Bayesian Nash incentive compatibility (BIC). Since truth revelation is always with respect to types, only direct revelation mechanisms are relevant when formalizing the notion of incentive compatibility. The notion of incentive compatibility was first introduced by Hurwicz 1973 [115].

Definition 2.8 (*Incentive Compatibility*)**.** *A social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$ is said to be incentive compatible (or truthfully implementable) if the Bayesian game induced by the direct revelation mechanism $\mathscr{D} = ((\Theta_i)_{i \in N}, f(\cdot))$ has a pure strategy equilibrium $s^*(\cdot) = (s_1^*(\cdot), \ldots, s_n^*(\cdot))$ in which $s_i^*(\theta_i) = \theta_i, \forall \theta_i \in \Theta_i, \forall i \in N$.*

That is, truth revelation by each player constitutes an equilibrium of the game induced by $\mathscr{D}$. It is easy to infer that if an SCF $f(\cdot)$ is incentive compatible then the direct revelation mechanism $\mathscr{D} = ((\Theta_i)_{i \in N}, f(\cdot))$ can implement it. That is, directly asking the players to report their types and using this information in $f(\cdot)$ to get the social outcome will solve both the problems, namely, preference elicitation and preference aggregation.

Based on the type of equilibrium concept used, we have,

Definition 2.9 (*Dominant Strategy Incentive Compatibility* (DSIC))**.** *A social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$ is said to be dominant strategy incentive compatible (or truthfully implementable in dominant strategies) if the direct revelation mechanism $\mathscr{D} = ((\Theta_i)_{i \in N}, f(\cdot))$ has a* weakly dominant strategy equilibrium *$s^*(\cdot) = (s_1^*(\cdot), \ldots, s_n^*(\cdot))$ in which $s_i^*(\theta_i) = \theta_i, \forall \theta_i \in \Theta_i, \forall i \in N$.*

That is, truth revelation by each player constitutes a dominant strategy equilibrium of the game induced by $\mathscr{D}$.

Definition 2.10 (*Bayesian Incentive Compatibility* (BIC))**.** *A social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$ is said to be Bayesian incentive compatible (or truthfully implementable in Bayesian Nash equilibrium) if the direct revelation mechanism $\mathscr{D} = ((\Theta_i)_{i \in N}, f(\cdot))$ has a* Bayesian Nash equilibrium *$s^*(\cdot) = (s_1^*(\cdot), \ldots, s_n^*(\cdot))$ in which $s_i^*(\theta_i) = \theta_i, \forall \theta_i \in \Theta_i, \forall i \in N$.*

That is, truth revelation by each player constitutes a Bayesian Nash equilibrium of the game induced by $\mathscr{D}$.

With the widespread internet, large-scale resource allocations happen every second. Especially in online auctions for internet advertisements, and resource allocation among various software nodes. Often the players involved could be software bots; henceforth we refer to our players as agents. Let us understand the above concepts better by taking the example of a first price auction.

First and Second Price Auction

Consider the problem of buying a single indivisible item or resource. We have a buying agent (agent 0) and two selling agents (agents 1 and 2), so we have $N = \{0, 1, 2\}$. An outcome here can be represented by $x = (y_0, y_1, y_2, t_0, t_1, t_2)$. For $i = 0$, we have

$$\begin{aligned} y_0 &= 0 \quad \text{if the buyer buys the good} \\ &= 1 \quad \text{otherwise} \\ t_0 &= \text{monetary transfer received by the buyer.} \end{aligned}$$

For $i = 1, 2$, we have

$$\begin{aligned} y_i &= 1 \quad \text{if agent } i \text{ supplies the goods to the buyer} \\ &= 0 \quad \text{if agent } i \text{ does not supply the good} \\ t_i &= \text{monetary transfer received by the agent } i. \end{aligned}$$

The set X of all feasible outcomes is given by

$$X = \{(y_0, y_1, y_2, t_0, t_1, t_2) : y_i \in \{0, 1\}, \sum_{i=0}^{2} y_i = 1, t_i \in \mathbb{R}, \sum_{i=0}^{2} t_i \leq 0\}.$$

The constraint $\sum_i t_i \leq 0$ implies that the total money received by all the agents are less than or equal to zero. That is, the total money paid by all the agents are greater than or equal to zero (that is, the buyer pays at least as much as the sellers receive. The excess between the payment and receipts is the surplus). For $x = (y_0, y_1, y_2, t_0, t_1, t_2)$, define the utilities to be of the form:

$$u_i(x, \theta_i) = u_i((y_0, y_1, y_2, t_0, t_1, t_2), \theta_i) = -y_i\theta_i + t_i \ ; \ \ i = 1, 2$$

where $\theta_i \in \mathbb{R}$ can be viewed as seller i's valuation of the good.

Below, we design two different indirect mechanisms for the above scenario and analyze their properties

Example 2.1 (*First Price Procurement Auction*)**.** *Here each seller submits a sealed bid, $b_i \geq 0$ $(i = 1, 2)$. The sealed bids are examined and the seller with the lower bid is declared the winner. If there is a tie, seller 1 is declared the winner. The winning seller receives an amount equal to his bid from the buyer. The losing seller does not receive anything.*

Let us make the following assumptions:

1. *θ_1, θ_2 are independently drawn from the uniform distribution on $[0, 1]$.*

2. *The sealed bid of seller i takes the form $b_i(\theta_i) = \alpha_i\theta_i + \beta_i$, where $\alpha_i \in [0, 1], \beta_i \in [0, 1 - \alpha_i]$. He has to make sure that $b_i \in [0, 1]$. The term β_i is like a fixed cost whereas $\alpha_i\theta_i$ indicates a fraction of the true cost.*

Seller 1's problem is now to bid in a way to maximize his payoff:

$$\max_{1 \geq b_1 \geq 0} (b_1 - \theta_1) P\{b_2(\theta_2) \geq b_1\}$$

$$\begin{aligned} P\{b_2(\theta_2) \geq b_1\} &= 1 - P\{b_2(\theta_2) < b_1\} \\ &= 1 - P\{\alpha_2\theta_2 + \beta_2 < b_1\} \\ &= 1 - \frac{b_1 - \beta_2}{\alpha_2} \ \textit{if } b_1 \geq \beta_2 \qquad (2.1) \\ &\quad \textit{since } \theta_2 \textit{ is uniform over } [0, 1] \end{aligned}$$

$$(2.2)$$

Thus seller 1's problem is:

$$\max_{b_1 \geq \beta_2} (b_1 - \theta_1)(1 - \frac{b_1 - \beta_2}{\alpha_2}).$$

The solution to this problem is

$$b_1(\theta_1) = \frac{\alpha_2 + \beta_2}{2} + \frac{\theta_1}{2}. \qquad (2.3)$$

We can show on similar lines that

$$b_2(\theta_2) = \frac{\alpha_1 + \beta_1}{2} + \frac{\theta_2}{2}. \qquad (2.4)$$

As the bid of seller i takes the form $b_i(\theta_i) = \alpha_i\theta_i + \beta_i$, where $\alpha_i \in [0,1], \beta_i \in [0, 1-\alpha_i]$, from the equations (2.3) and (2.4), we obtain $\alpha_1 = \alpha_2 = \frac{1}{2}$. As the goal of each seller is to maximize the profit and $\beta_i \in [0, 1-\alpha_i]$, $\beta_1 = \beta_2 = \frac{1}{2}$. Then we get

$$\begin{aligned} b_1(\theta_1) &= \frac{1+\theta_1}{2} \qquad \forall\ \theta_1 \in \Theta_1 = [0,1] \\ b_2(\theta_2) &= \frac{1+\theta_2}{2} \qquad \forall\ \theta_2 \in \Theta_2 = [0,1]. \end{aligned}$$

Note that if $b_2(\theta_2) = \frac{1+\theta_2}{2}$, the best response of seller 1 is $b_1(\theta_1) = \frac{1+\theta_1}{2}$ and vice-versa. Hence the profile $\left(\frac{1+\theta_1}{2}, \frac{1+\theta_2}{2}\right)$ is a Bayesian Nash equilibrium of an underlying Bayesian game. In other words, there is a Bayesian Nash equilibrium of an underlying game (induced by the indirect mechanism called the first price procurement auction) that (indirectly) yields the outcome

$$f(\theta) = (y_0(\theta), y_1(\theta), y_2(\theta), t_0(\theta), t_1(\theta), t_2(\theta))$$

such that

$$\begin{aligned} y_0(\theta) &= 0 \qquad \forall\ \theta \in \Theta \\ y_1(\theta) &= 1 \qquad \text{if } \theta_1 \le \theta_2 \\ &= 0 \qquad \text{else} \\ y_2(\theta) &= 1 \qquad \text{if } \theta_1 > \theta_2 \\ &= 0 \qquad \text{else} \\ t_1(\theta) &= \frac{1+\theta_1}{2} y_1(\theta) \\ t_2(\theta) &= \frac{1+\theta_2}{2} y_2(\theta) \\ t_0(\theta) &= -(t_1(\theta) + t_2(\theta)). \end{aligned}$$

Example 2.2 (*Second Price Procurement Auction*)**.** *Here, each seller is asked to submit a sealed bid $b_i \geq 0$. The bids are examined, and the seller with the lower bid is declared the winner. In case there is a tie, seller 1 is declared the winner. The winning seller receives as payment from the buyer an amount equal to the second lowest bid. The losing bidder* *does not receive anything. In this case, we can show that $b_i(\theta_i) = \theta_i$ for $i = 1, 2$ constitutes a weakly dominant strategy for each player.*

Thus the game induced by the indirect mechanism second price procurement auction has a weakly dominant strategy in which truthful revelation is optimal.

2.3.2 Other Properties

We have seen that a mechanism provides a solution to both the preference elicitation problem and the preference aggregation problem. Here we study certain important and desirable properties that a social choice function must satisfy.

Definition 2.11 (*Ex-Post Efficiency*)**.** *The SCF $f : \Theta \to X$ is said to be ex-post efficient (or Paretian) if for every profile of agents' types, $\theta \in \Theta$, the outcome $f(\theta)$ is a Pareto optimal outcome. The outcome $f(\theta_1, \ldots, \theta_n)$ is Pareto optimal if there does not exist any $x \in X$ such that:*

$$u_i(x, \theta_i) \geq \ u_i(f(\theta), \theta_i) \ \forall \ \ i \in N \ \text{ and } \ u_i(x, \theta_i) > \ u_i(f(\theta), \theta_i) \ \text{ for some } \ i \in N.$$

Definition 2.12 (*Dictatorship*)**.** *A social choice function $f : \Theta \to X$ is said to be dictatorial if there exists an agent d (called dictator) who satisfies the following property:*

$$\forall \ \theta \in \Theta, \ \ f(\theta) \ \text{ is such that } \ u_d(f(\theta), \theta_d) \geq u_d(x, \theta_d) \ \ \forall \ x \in X.$$

A social choice function that is not dictatorial is said to be nondictatorial .

In a dictatorial SCF, every outcome that is picked by the SCF is such that it is a most favoured outcome for the dictator.

Individual Rationality

This property ensures that each agent is not worse off by participating in the mechanism. Hence it ensures non-negative utility to each participating agent. There are three stages at which individual rationality constraints may be relevant in a mechanism design situation.

- *Ex-Post Individual Rationality.* When the agent knows the types of all everyone, let $\overline{u_i}(\theta_i)$ be the utility that agent i receives by withdrawing from the mechanism when his type is θ_i. Then, f satisfies ex-post participation (or individual rationality) constraints when,

$$u_i(f(\theta_i, \theta_{-i}), \theta_i) \geq \overline{u_i}(\theta_i) \ \ \forall \ (\theta_i, \theta_{-i}) \in \Theta.$$

- *Interim Individual Rationality.* When the agent is only aware of its own type, its interim expected utility is given by $U_i(\theta_i|f) = E_{\theta_{-i}}[u_i(f(\theta_i, \theta_{-i}), \theta_i)|\theta_i]$. Thus, interim participation (or individual rationality) constraints for agent i require that

$$U_i(\theta_i|f) = E_{\theta_{-i}}[u_i(f(\theta_i, \theta_{-i}), \theta_i)|\theta_i] \geq \overline{u_i}(\theta_i) \ \ \forall \ \theta_i \in \Theta_i.$$

- *Ex-Ante Individual Rationality.* The agent is not aware of even its own type, its ex-ante expected utility is $U_i(f) = E_\theta[u_i(f(\theta_i, \theta_{-i}), \theta_i)]$ from social choice function $f(\cdot)$. Thus, ex-ante participation (or individual rationality) constraints for agent i require that

$$U_i(f) = E_\theta[u_i(f(\theta_i, \theta_{-i}), \theta_i)] \geq E_{\theta_i}[\overline{u_i}(\theta_i)].$$

The following proposition establishes a relationship among the three different participation constraints discussed above.

Proposition 2.1. *For any social choice function $f(\cdot)$, we have*

$$f(\cdot) \textit{ is ex-post IR} \Rightarrow f(\cdot) \textit{ is interim IR} \Rightarrow f(\cdot) \textit{ is ex-ante IR.}$$

2.3.3 The Gibbard-Satterthwaite Impossibility Theorem

We have seen in the last section that dominant strategy incentive compatibility is an extremely desirable property of social choice functions. However, the DSIC property, being a strong one, precludes certain other desirable properties to be satisfied. The *Gibbard–Satterthwaite impossibility theorem* (GS theorem, for short), shows that the DSIC property will force an SCF to be dictatorial if the utility environment is an unrestricted one. The GS theorem is credited independently to Gibbard 1973 [85] and Satterthwaite 1975 [172]. The GS theorem is a brilliant reinterpretation of the famous Arrow's impossibility theorem.

Theorem 2.1 (*Gibbard–Satterthwaite Impossibility Theorem*)**.** *Consider a social choice function $f : \Theta \rightarrow X$. Suppose that*

1. *The outcome set X is finite and contains at least three elements,*

2. *$\mathscr{R}_i = \mathscr{P} \;\; \forall\, i \in N$,*

3. *$f(\cdot)$ is an onto mapping, that is, the image of SCF $f(\cdot)$ is the set X.*

Then the social choice function $f(\cdot)$ is dominant strategy incentive compatible iff it is dictatorial.

For proof of this theorem, the reader is referred to Proposition 23.C.3 of the book by Mas-Colell et al. 1995 [137]. One way to get around the impossible situation described by the GS Theorem is to hope that at least one of the conditions (1), (2), and (3) of the theorem does not hold. With this intuition, we look at the most practical and widely studied assumption made on the utility function.

2.3.4 The Quasilinear Environment

This is the most extensively studied special class of environments where the Gibbard–Satterthwaite theorem does not hold. In the quasilinear environment, an alternative $x \in X$ is a vector of the form $x = (k, t_1, \ldots, t_n)$, where k is an element of a set K, which is called the set of project choices or set of allocations. The set K is usually assumed to be finite. The term $t_i \in \mathbb{R}$ represents the monetary transfer to agent i. If $t_i > 0$ then agent i will receive the money and if $t_i < 0$ then agent i will pay the money. We assume that we are dealing with a system in which the n agents have no external source of funding, i.e., $\sum_{i=1}^{n} t_i \leq 0$. This condition is known as the *weak budget balance* condition. The set of alternatives X is therefore

$$X = \left\{ (k, t_1, \ldots, t_n) : k \in K;\ t_i \in \mathbb{R} \ \ \forall\, i \in N;\ \ \sum_i t_i \leq 0 \right\}.$$

An SCF in this quasilinear environment takes the form $f(\theta) = (k(\theta), t_1(\theta), \ldots, t_n(\theta))$ where, for every $\theta \in \Theta$, we have $k(\theta) \in K$ and $\sum_i t_i(\theta) \leq 0$. Note that here we are using the symbol k both as an element of the set K and as a function going from Θ to K. It should be clear from the context as to which of these two we are referring. For a direct revelation mechanism $\mathscr{D} = ((\Theta_i)_{i\in N}, f(\cdot))$ in this environment, the agent i's utility function takes the quasilinear form

$$u_i(x, \theta_i) = u_i((k, t_1, \ldots, t_n), \theta_i) = v_i(k, \theta_i) + m_i + t_i$$

where m_i is agent i's initial endowment of the money and the function $v_i(\cdot)$ is known as agent i's valuation function. Recall from our discussion of mechanism design environment (Section 2.2) that the utility functions $u_i(\cdot)$ are common knowledge. In the context of a quasilinear environment, this implies that for any given type θ_i of any agent i, the social planner and every other agent j have a way to know the function $v_i(., \theta_i)$. In many cases, the set Θ_i of the direct revelation mechanism $\mathscr{D} = ((\Theta_i)_{i\in N}, f(\cdot))$ is actually the set of all feasible valuation functions v_i of agent i. That is, each possible function represents the possible types of agent i. Therefore, in such settings, reporting a type is the same as reporting a valuation function.

Immediate examples of quasilinear environment include many of the previously discussed examples, such as the first price and second price auctions. In the quasilinear environment, we can define two important properties of a social choice function, namely, allocative efficiency and budget balance.

Definition 2.13 (*Allocative Efficiency* (AE))**.** *We say that a social choice function $f(\cdot) = (k(\cdot), t_1(\cdot), \ldots, t_n(\cdot))$ is allocatively efficient if for each $\theta \in \Theta$, $k(\theta)$ satisfies the following* *condition*[1] *We will be using the symbol $k^*(\cdot)$ for a function $k(\cdot)$ that satisfies Equation (2.5).*

$$k(\theta) \in \begin{matrix} \arg\max \\ k \in K \end{matrix} \sum_{i=1}^{n} v_i(k, \theta_i). \tag{2.5}$$

Equivalently,

$$\sum_{i=1}^{n} v_i(k(\theta), \theta_i) = \begin{matrix} \max \\ k \in K \end{matrix} \sum_{i=1}^{n} v_i(k, \theta_i).$$

The above definition implies that for every $\theta \in \Theta$, the allocation $k(\theta)$ will maximize the sum of the values of the players. In other words, every allocation is a value maximizing allocation, or the objects are allocated to the players who value the objects most. This is an extremely desirable property to have for any social choice function. The above definition implicitly assumes that for any given θ, the function $\sum_{i=1}^{n} v_i(., \theta_i) : K \to \mathbb{R}$ attains a maximum over the set K.

Definition 2.14 (*Budget Balance* (BB))**.** *We say that a social choice function $f(\cdot) = (k(\cdot), t_1(\cdot), \ldots, t_n(\cdot))$ is budget balanced, if for each $\theta \in \Theta$, $t_1(\theta), \ldots, t_n(\theta)$ satisfy the following condition:*

$$\sum_{i=1}^{n} t_i(\theta) = 0. \tag{2.6}$$

Many authors prefer to call this property *strong budget balance*, and they refer to the property of having $\sum_{i=1}^{n} t_i(\theta) \leq 0$ as *weak budget balance*. In this thesis, we will use the term budget balance to refer to a strong budget balance.

Budget balance ensures that the total receipts are equal to the total payments. This means that the system is a closed one, with no surplus and no deficit. The weak budget balance property means that the total payments are greater than or equal to total receipts.

The following lemma establishes an important relationship of these two properties of an SCF with the ex-post efficiency of the SCF.

[1]

Lemma 2.1. *A social choice function $f(\cdot) = (k(\cdot), t_1(\cdot), \ldots, t_n(\cdot))$ is ex-post efficient in a quasilinear environment if and only if it is allocatively efficient and budget balanced.*

Proof. Let us assume that $f(\cdot) = (k(\cdot), t_1(\cdot), \ldots, t_n(\cdot))$ is allocatively efficient and budget balanced. This implies that for any $\theta \in \Theta$, we have

$$\begin{aligned} \sum_{i=1}^{n} u_i(f(\theta), \theta_i) &= \sum_{i=1}^{n} v_i(k(\theta), \theta_i) + \sum_{i=1}^{n} t_i(\theta) \\ &= \sum_{i=1}^{n} v_i(k(\theta), \theta_i) + 0 \\ &\geq \sum_{i=1}^{n} v_i(k, \theta_i) + \sum_{i=1}^{n} t_i; \;\; \forall\, x = (k, t_1, \ldots, t_n) \\ &= \sum_{i=1}^{n} u_i(x, \theta_i); \;\; \forall\, (k, t_1, \ldots, t_n) \in X. \end{aligned}$$

That is if the SCF is allocatively efficient and budget balanced then for any type profile θ of the agent, the outcome chosen by the social choice function will be such that it maximizes the total utility derived by all the agents. This will automatically imply that the SCF is ex-post efficient.

To prove the other part, we will first show that if $f(\cdot)$ is not allocatively efficient, then, it cannot be ex-post efficient and next we will show that if $f(\cdot)$ is not budget balanced then it cannot be ex-post efficient. These two facts together will imply that if $f(\cdot)$ is ex-post efficient then it will have to be allocatively efficient and budget balanced, thus completing the proof of the lemma.

To start with, let us assume that $f(\cdot)$ is not allocatively efficient. This means that $\exists\, \theta \in \Theta$, and $k \in K$ such that,

$$\sum_{i=1}^{n} v_i(k, \theta_i) > \sum_{i=1}^{n} v_i(k(\theta), \theta_i).$$

This implies that there exists at least one agent j for whom $v_j(k, \theta_i) > v_j(k(\theta), \theta_i)$. Now consider the following alternative x

$$x = \Big(k, \left(t_i = t_i(\theta) + v_i(k(\theta), \theta_i) - v_i(k, \theta_i)\right)_{i \neq j}, t_j = t_j(\theta)\Big).$$

It is easy to verify that $u_i(x, \theta_i) = u_i(f(\theta), \theta_i) \; \forall \; i \neq j$ and $u_j(x, \theta_i) > u_j(f(\theta), \theta_i)$, implying that $f(\cdot)$ is not ex-post efficient.

Next, we assume that $f(\cdot)$ is not budget balanced. This means that there exists at least one agent j for whom $t_j(\theta) < 0$. Let us consider the following alternative x

$$x = \Big(k, (t_i = t_i(\theta))_{i \neq j}, t_j = 0\Big).$$

It is easy to verify that for the above alternative x, we have $u_i(x, \theta_i) = u_i(f(\theta), \theta_i) \; \forall \; i \neq j$ and $u_j(x, \theta_i) > u_j(f(\theta), \theta_i)$ implying that $f(\cdot)$ is not ex-post efficient.

□

The next lemma summarizes another fact about social choice functions in quasilinear environment.

Lemma 2.2. *All social choice functions in quasilinear environments are nondictatorial.*

Proof. If possible, assume that a social choice function, $f(\cdot)$, is dictatorial in the quasilinear environment. This means that there exists an agent called the dictator, say $d \in N$, such that for each $\theta \in \Theta$, we have

$$u_d(f(\theta), \theta_d) \geq u_d(x, \theta_d) \; \forall \; x \in X.$$

However, because the environment is quasilinear, we have $u_d(f(\theta), \theta_d) = v_d(k(\theta), \theta_d) + t_d(\theta)$. Now consider the following alternative $x \in X$:

$$x = \begin{cases} (k(\theta), (t_i = t_i(\theta))_{i \neq d}, t_d = t_d(\theta) - \sum_{i=1}^{n} t_i(\theta)) & : \quad \sum_{i=1}^{n} t_i(\theta) < 0 \\ (k(\theta), (t_i = t_i(\theta))_{i \neq d,j}, t_d = t_d(\theta) + \epsilon, t_j = t_j(\theta) - \epsilon) & : \quad \sum_{i=1}^{n} t_i(\theta) = 0 \end{cases}$$

where $\epsilon > 0$ is any arbitrary number, and j is any agent other than d. It is easy to verify, for the above outcome x, that we have $u_d(x, \theta_d) > u_d(f(\theta), \theta_d)$, which contradicts the fact that d is a dictator. □

In view of Lemma 2.2, the social planner need not have to worry about the nondictatorial property of the social choice function in quasilinear environments and he can simply look for whether there exists any SCF that is both ex-post efficient and dominant strategy incentive compatible. Furthermore, in the light of Lemma 2.1, we can say that the social planner can look for an SCF that is allocatively efficient, budget balanced, and dominant strategy incentive compatible. Once again the question arises whether there could exist social choice functions which satisfy all these three properties — AE, BB, and DSIC.

2.4 Part I - Mechanism Design with Money (Groves Mechanisms)

An important possibility result in mechanism design is that in the quasilinear environment, there exist social choice functions that are both allocatively efficient and dominant strategy incentive compatible. These are in general called the VCG (Vickrey–Clarke–Groves) mechanisms.

2.4.1 VCG Mechanisms

The VCG mechanisms are named after their famous inventors William Vickrey, Edward Clarke, and Theodore Groves. It was Vickrey who introduced the famous Vickrey auction (second price sealed bid auction) in 1961 [191]. To this day, the Vickrey auction continues to enjoy a special place in the annals of mechanism design. Clarke [54] and Groves [94] came up with a generalization of the Vickrey mechanisms and helped define a broad class of dominant strategy incentive compatible mechanisms in the quasilinear environment. VCG mechanisms are by far the most extensively used among quasilinear mechanisms. They are popular due to their mathematical elegance and the strong properties they satisfy.

Groves' Theorem

The following theorem provides a sufficient condition for an allocatively efficient social function in quasilinear environment to be dominant strategy incentive compatible.

Theorem 2.2 (*Groves Theorem* [94])**.** *Let the SCF $f(\cdot) = (k^*(\cdot), t_1(\cdot), \ldots, t_n(\cdot))$ be allocatively efficient. Then $f(\cdot)$ is dominant strategy incentive compatible if it satisfies the following payment structure (popularly known as the Groves payment (incentive) scheme):*

$$t_i(\theta) = \left[\sum_{j \neq i} v_j(k^*(\theta), \theta_j) \right] + h_i(\theta_{-i}) \;\; \forall \, i = 1, \ldots, n \tag{2.7}$$

where $h_i : \Theta_{-i} \to \mathbb{R}$ is any arbitrary function that honors the feasibility condition $\sum_i t_i(\theta) \leq 0 \; \forall \, \theta \in \Theta$.

The proof for the above theorem can be found in [151, Theorem 18.1]. A *Groves Mechanism* is a direct revelation mechanism in which the implemented SCF is allocatively efficient and satisfies the Groves payment scheme.

Definition 2.15 (*Groves Mechanisms*)**.** *A direct mechanism, $\mathscr{D} = ((\Theta_i)_{i \in N}, f(\cdot))$ in which $f(\cdot) = (k(\cdot), t_1(\cdot), \ldots, t_n(\cdot))$ satisfies allocative efficiency (2.5) and Groves payment rule (2.7) is known as a Groves mechanism.*

In mechanism design parlance, Groves mechanisms are popularly known as Vickrey–Clarke–Groves (VCG) mechanisms.

The Groves theorem provides a sufficiency condition under which an allocatively efficient (AE) SCF will be DSIC. The following theorem due to Green et al. 1979 [93] provides a set of conditions under which the condition of Groves Theorem also becomes a necessary condition for an AE SCF to be DSIC. In this theorem, we let $\mathscr{F}$ denote the set of all possible functions $f : K \to \mathbb{R}$.

Theorem 2.3 (*First Characterization Theorem of Green–Laffont*)**.** *Suppose for each agent $i \in N$ that $\{v_i(., \theta_i) : \theta_i \in \Theta_i\} = \mathscr{F}$, that is, every possible valuation function from K to $\mathbb{R}$ arises for some $\theta_i \in \Theta_i$. Then any allocatively efficient social choice function $f(\cdot)$ will be dominant strategy incentive compatible if and only if it satisfies the Groves payment scheme given by Equation (2.7).*

Note that in the above theorem, every possible valuation function from K to $\mathbb{R}$ arises for any $\theta_i \in \Theta_i$. In the following characterization theorem, again due to Green et al. 1979 [93], $\mathscr{F}$ is replaced with with $\mathscr{F}_c$ where $\mathscr{F}_c$ denotes the set of all possible continuous functions $f : K \to \mathbb{R}$.

Theorem 2.4 (*Second Characterization Theorem of Green–Laffont*)**.** *Suppose for each agent $i \in N$ that $\{v_i(.,\theta_i) : \theta_i \in \Theta_i\} = \mathscr{F}_c$, that is, every possible continuous valuation function from K to $\mathbb{R}$ arises for some $\theta_i \in \Theta_i$. Then any allocatively efficient social choice function $f(\cdot)$ will be dominant strategy incentive compatible if and only if it satisfies the Groves payment scheme given by Equation (2.7).*

2.4.2 Clarke (Pivotal) Mechanisms

A special case of Groves mechanism was developed independently by Clarke in 1971 [54] and is known as the *Clarke*, or the *pivotal* mechanism. It is a special case of Groves mechanisms in the sense of using a natural special form for the function $h_i(\cdot)$. In the Clarke mechanism, the function $h_i(\cdot)$ is given by the following relation:

$$h_i(\theta_{-i}) = -\sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j) \;\; \forall\, \theta_{-i} \in \Theta_{-i}, \forall\, i = 1, \ldots, n \tag{2.8}$$

where $k^*_{-i}(\theta_{-i}) \in K_{-i}$ is the choice of a project that is allocatively efficient if there were only the $n-1$ agents $j \neq i$. Formally, $k^*_{-i}(\theta_{-i})$ must satisfy the following condition.

$$\sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j) \geq \sum_{j \neq i} v_j(k, \theta_j) \; \forall\, k \in K_{-i} \tag{2.9}$$

where the set K_{-i} is the set of project choices available when agent i is absent. Substituting the value of $h_i(\cdot)$ from Equation (2.8) in Equation (2.7), we get the following expression for agent i's transfer in the Clarke mechanism:

$$t_i(\theta) = \left[\sum_{j \neq i} v_j(k^*(\theta), \theta_j)\right] - \left[\sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j)\right]. \tag{2.10}$$

The above payment rule has an appealing interpretation: Given a type profile $\theta = (\theta_1, \ldots, \theta_n)$, the monetary transfer to agent i is given by the total value of all agents other than i under an efficient allocation when agent i is present in the system minus the total value of all agents other than i under an efficient allocation when agent i is absent in the system.

2.4.3 Groves Mechanisms and Budget Balance

Note that a Groves mechanism always satisfies the properties of AE and DSIC. Therefore, if a Groves mechanism is budget balanced, then it will solve the problem of the social planner because it will then be ex-post efficient and dominant strategy incentive compatible. By looking at the definition of the Groves mechanism, one can conclude that it is the functions $h_i(\cdot)$ that decide whether or not the Groves mechanism is budget balanced. The natural question that arises now is whether there exists a way of defining functions $h_i(\cdot)$ such that the Groves mechanism is budget balanced. In what follows, we present one possibility result and one impossibility result in this regard.

2.4.4 Green Laffont Impossibility Result for Quasilinear Environments

Green and Laffont [93] showed that in a quasilinear environment, if the set of possible types for each agent is sufficiently rich then ex-post efficiency and DSIC cannot be achieved together. The precise statement is given in the form of the following theorem.

Theorem 2.5 (*Green–Laffont Impossibility Theorem*)**.** *Suppose for each agent $i \in N$ that $\mathscr{F} = \{v_i(., \theta_i) : \theta_i \in \Theta_i\}$, that is, every possible valuation function from K to $\mathbb{R}$ arises for some $\theta_i \in \Theta_i$. Then there is no social choice function that is ex-post efficient and DSIC.*

Thus, the above theorem says that if the set of possible types for each agent is sufficiently rich then there is no hope of finding a way to define the functions $h_i(\cdot)$ in Groves payment scheme so that we have $\sum_{i=1}^{n} t_i(\theta) = 0$. Hence, in the next section, we discuss Redistribution Mechanisms, which satisfy EPE, DSIC and minimizes the budget imbalance by redistributing the money back to agents.

2.4.5 Redistribution Mechanism

Consider that p resources are available and each of $n > p$ agents is interested in utilizing one of them. Naturally, we should assign these resources to those agents who value them the most. Since Vickery, Clarke and Groves mechanisms [54, 94, 191] have attractive game theoretic properties such as dominant strategy incentive compatibility (DSIC) and allocative efficiency (AE), Groves mechanisms are quite appealing to use in this context. However, in general, a Groves mechanism need not be budget balanced. That is, the total transfer of money in the system may not be zero. So the system will be left with a surplus or deficit. Using Clarke's mechanism [54], we can ensure under fairly weak conditions, that there is no deficit of money (that is the mechanism is weakly budget balanced). In such a case, the system or the auctioneer will be left with some money.

Often, surplus money is not really needed in many social settings such as allocations by the Government among its departments, etc. Since strict budget balance cannot coexist with DSIC and AE (Green-Laffont theorem [93]), we would like to redistribute the surplus to the participants as far as possible, preserving DSIC and AE. This idea was originally proposed by Laffont ([138]). The total payment made by the mechanism as a redistribution will be referred to as the *rebate* to the agents. More formally,

Definition 2.16 (*Redistribution Mechanism*)**.** *We call a Groves mechanism as* Groves redistribution mechanism *or simply* redistribution mechanism*, if it allocates objects to the agents in an allocatively efficient way and redistributes the Clarke surplus in the system in the form of rebates to the agents such that the net payment made by each agent still follows Groves payment structure.*

Given that the agents report their bids $b = (b_1, b_2, \ldots, b_n)$ where $b_i = (b_{i1}, \ldots, b_{ip})$ is the bid submitted by agent i for p items. For a bid profile b, the rebate to an agent i is denoted by $r_i(b)$. Further $t_i(b)$ is the payment made by i in Clarke pivotal mechanism i.e., $t_i(b) = v_i(k^*(b)) - (v(k^*(b)) - v(k^*_{-i}(b)))$, where $k^*(b)$ is an allocatively efficient allocation and

$k^*_{-i}(b)$ is allocatively efficient allocation without agent i. The rebated offered is calculated using a rebate function as formally defined below.

Definition 2.17 (*Linear Rebate Function*)**.** *We say a rebate to an agent is* linear rebate function*, if it is linear combination of bid vectors of all the remaining agents. Moreover, if a redistribution mechanism uses linear rebate functions for all the agents, we say the mechanism is* linear redistribution mechanism*.*

Definition 2.18 (*Redistribution Index*)**.** *A* redistribution index *of a redistribution mechanism is defined to a worst case fraction of Clarke's surplus that gets redistributed among the agents. That is,*

$$e = \inf_{b:t(b)\neq 0} \frac{\sum r_i(b)}{t(b)}$$

Designing rebate functions have been explored both when items are *Homogeneous* and *Heterogneous*. Homogeneous implies that the items are identical, and *Worst Case Optimal Redistribution* (WCO) provides a linear rebate function which maximizes the redistribution index. Heterogeneous implies non-identical items, for which the linear rebate function cannot be useful. [95] prove the impossibility of the existence of a linear rebate function with a non-zero redistribution index. Further, the authors propose HETERO a mechanism with optimal non-zero redistribution index as proven in [99].

2.4.5.1 Optimal Worst Case Redistribution for Homogeneous Items

When the objects are identical, every agent i has the same value for each object, call it v_i. Without loss of generality, we will assume, $v_1 \geq v_2 \geq \ldots \geq v_n$. In Clarke's pivotal mechanism, the first p agents will receive the objects and each of these p agents will pay v_{p+1}. So, the surplus in the system is $p \times v_{p+1}$. For this situation, Moulin [147] and [103] have independently designed a redistribution mechanism.

Guo et al. 2009 [103] maximize the worst case fraction of the total surplus which gets redistributed. This mechanism is called the WCO mechanism. Moulin 2009 [149] minimizes the ratio of budget imbalance to the value of an optimal allocation, that is the value of an allocatively efficient allocation. The WCO mechanism coincides with Moulin's feasible and individually rational mechanism. Both the above mechanisms work as follows. After receiving bids from the agents, bids are sorted in decreasing order. The first p agents receive the objects. Each agent's Clarke payment is calculated, say t_i. Every agent i pays, $p_i = t_i - r_i$, where, r_i is the rebate function for an agent i.

$$\begin{aligned} r_i^{WCO} &= c_{p+1}v_{p+2} + c_{p+2}v_{p+3} + \ldots + c_{n-1}v_n & i = 1, \ldots p+1 \\ r_i^{WCO} &= c_{p+1}v_{p+1} + \ldots + c_{i-1}v_{i-1} + c_i v_{i+1} + \ldots + c_{n-1}v_n & i = p+2, \ldots n \end{aligned} \tag{2.11}$$

where,

$$c_i = \frac{(-1)^{i+p-1}\,(n-p)\binom{n-1}{p-1}}{i\binom{n-1}{i}\sum_{j=p}^{n-1}\binom{n-1}{j}} \left\{ \sum_{j=i}^{n-1} \binom{n-1}{j} \right\}; \quad i = p+1, \ldots, n-1 \tag{2.12}$$

Suppose $y_1 \geq y_2 \geq \ldots \geq y_{n-1}$ are the bids of the $(n-1)$ agents excluding the agent i, then equivalently the rebate to the agent i is given by,

$$r_i^{WCO} = \sum_{j=p+1, j \neq i}^{n-1} c_j y_j \tag{2.13}$$

The redistribution index of this mechanism is e^*, where e^* is given by,

$$e^* = 1 - \frac{\binom{n-1}{p}}{\sum_{j=p}^{n-1}\binom{n-1}{j}}$$

This is an optimal mechanism since there is no other mechanism which can guarantee more than e^* fraction redistribution in the worst case. Next, we discuss the heterogeneous setting and impossibility of having a linear rebate function.

2.4.5.2 Impossibility of Linear Rebate Function with Non-Zero Redistribution Index for Heterogeneous Items

We have seen that the WCO mechanism is a linear function of the types of agents. We now explore the general case. In the homogeneous case, the bids are real numbers which can be arranged in decreasing order. The Clarke surplus is a linear function of these ordered bids. For the heterogeneous scenario, this would not be the case. Each bid b_i belongs to $\mathbb{R}^p_+$; hence, there is no unique way of defining an order among the bids. Moreover, the Clarke surplus is not a linear function of the received bids in the heterogeneous case. So, there cannot be any linear/affine rebate function of types to work well at all type profiles. The following theorems state this more formally. The symbol $\succcurlyeq$ denotes the order over the bids of the agents, as defined in [95].

Theorem 2.6. *In Groves redistribution mechanism, any deterministic, anonymous rebate function f is DSIC iff,*

$$r_i = f(v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n) \;\; \forall i \in N \tag{2.14}$$

where, $v_1 \succcurlyeq v_2 \succcurlyeq \ldots \succcurlyeq v_n$.

Theorem 2.7. *If a redistribution mechanism is feasible and individually rational, then there cannot exist a linear rebate function which is DSIC, deterministic, anonymous and provides non-zero redistribution index.*

Proof. Assume that there exists a linear function, say f, which satisfies the above properties. Let $v_1 \succcurlyeq v_2 \succcurlyeq \ldots \succcurlyeq v_n$. Then according to Theorem 2.6, for each agent i,

$$\begin{aligned} r_i &= f(v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n) \\ &= (c_0, e_p) + (c_1, v_1) + \ldots + (c_{n-1}, v_n) \end{aligned}$$

where, $c_i = (c_{i1}, c_{i2}, \ldots, c_{ip}) \in \mathbb{R}^p$, $e_p = (1, 1, \ldots, 1) \in \mathbb{R}^p$, and $(\cdot, \cdot)$ denotes the inner product of two vectors in $\mathbb{R}^p$. Now, we will show that the worst case performance of f will be zero. To this end, we will study the structure of f, step by step. □

Observation 1: Consider type profile $(v_1, v_2, \ldots, v_n)$ where $v_1 = v_2 = \ldots = v_n = (0, 0, \ldots, 0)$. For this type profile, the total Clarke surplus is zero and $r_i = (c_0, e_p) \; \forall i \in N$. Individual rationality implies,

$$(c_0, e_p) \geq 0 \tag{2.15}$$

Feasibility implies the total redistributed amount is less than the surplus, that is,

$$\sum_i r_i = n(c_0, e_p) \leqslant 0 \tag{2.16}$$

From, (2.15) and (2.16), it is easy to see that, $(c_0, e_p) = 0$.

<u>Observation 2:</u> Consider type profile $(v_1, v_2, \ldots, v_n)$ where $v_1 = (1, 0, 0, \ldots, 0)$ and $v_2 = \ldots, v_n = (0, 0, \ldots, 0)$. For this type profile, $r_1 = 0$ and if $i \neq 1$, $r_i = c_{11} \geq 0$ for individual rationality. For this type profile, it can be seen through straightforward calculations that the Clarke surplus is zero. Thus, for feasibility, $\sum_i r_i = (n-1)c_{11} \leq t = 0$. This implies, $c_{11} = 0$.

In the above profile, by considering $v_1 = (0, 1, 0, \ldots, 0)$, we get $c_{12} = 0$. Similarly, one can show $c_{13} = c_{14} = \ldots = c_{1p} = 0$.

<u>Observation 3:</u> Continuing like above with, $v_1 = v_2 = \ldots = v_i = e_p$, and $v_{i+1} = (1, 0 \ldots, 0)$ or $(0, 1, 0 \ldots, 0), \ldots$ or $(0, \ldots, 0, 1)$, we get, $c_{i+1} = (0, 0, \ldots, 0)\ \forall\ i \leq p-1$. Thus,

$$r_i = \begin{cases} (c_{p+1}, v_{p+2}) + \ldots + (c_{n-1}, v_n) & : \text{if } i \leq p+1 \\ (c_{p+1}, v_{p+1}) + \ldots + (c_{i-1}, v_{i-1}) & \\ \qquad +(c_i, v_{i+1}) + \ldots + (c_{n-1}, v_n) & : \text{otherwise} \end{cases} \tag{2.17}$$

Thus a rebate function in any linear redistribution mechanism has to be of the form in the Equation (2.17). We now claim that the redistribution index of such mechanism is zero. For any individually rational redistribution mechanism, a trivial lower bound on the redistribution index is zero. We prove that in a linear redistribution mechanism, there exists a type profile, at which the fraction of the Clarke surplus that gets redistributed is zero. Consider the type profile:

$$\begin{aligned}
v_1 &= (2p-1, 2p-2, \ldots, p+1, p) \\
v_2 &= (2p-2, 2p-3, \ldots, p, p-1) \\
&\vdots \\
v_{p-1} &= (p+1, p, \ldots, 3, 2) \\
v_p &= (p, p-1, \ldots, 2, 1)
\end{aligned}$$

and $v_{p+1} = v_{p+2} \ldots = v_n = (0, 0, \ldots, 0)$.

Now it can be seen through straight forward calculations of Clarke's payment, with this type profile, agent 1 pays $(p-1)$, agent 2 pays $(p-2), \ldots,$ agent $(p-1)$ pays 1 and the remaining agents pay 0. Thus, the Clarke payment received is non-zero but it can be seen that $r_i = 0$ for all the agents. Hence, the redistribution index for any linear redistribution mechanism has to be zero.

The above theorem provides disappointing news. It rules out the possibility of a linear redistribution mechanism for the heterogeneous settings which will have non-zero redistribution index. However, there are two ways to get around it.

1. The domain of types under which Theorem 2.7 holds is, $\Theta_i = \mathbb{R}_+^p$, $\forall i \in N$. One idea is to restrict the domain of types.

2. Explore the existence of a rebate function which is not linear and yields a non-zero performance.

In the next section, we state the results corresponding to non-linear rebate functions.

2.4.5.3 Non-linear Redistribution Mechanisms for the Heterogeneous Setting

We should note that the homogeneous objects case is a special case of the heterogeneous objects case in which each bidder submits the same bid for all objects. Thus, we cannot expect any redistribution mechanism to perform better than the homogeneous objects case. For $n \le p+1$, the worst case redistribution is zero for the homogeneous case and so will be for the heterogeneous case ([103, 149]). In this section, we discuss *HETERO*, which provides a non-linear rebate function when $n > p + 1$.

When the objects are identical, the WCO mechanism is given by the equation (2.13). We give a novel interpretation to it. Consider the scenario in which one agent is absent from the scene. Then Clarke's payment received is either pv_{p+1} or pv_{p+2} depending upon which agent is absent. If we remove two agents, the surplus is pv_{p+1} or pv_{p+2} or pv_{p+3}, depending upon which two agents are removed. Till $(n-p-1)$ agents are removed, we get non-zero surplus. If we remove $(n-p)$ or more agents from the system, there is no need for any mechanism for the assignment of the objects. So, we will consider the cases when we remove k agents, where, $1 \leq k < n-p$.

Now let $t^{-i,k}$ be the average payment received when agent i is removed along with k other agents that is, a total of $(k+1)$ agents are removed comprising of i. The average is taken over all possible selections of k agents from the remaining $(n-1)$ agents. We can rewrite the WCO mechanism in terms of $t^{-i}, t^{-i,k}$. Observe that, $t^{-i}, t^{-i,k}$ can be defined in heterogeneous settings as well. The rebate function for HETERO is defined as,

$$r_i^H = \alpha_1 t^{-i} + \sum_{k=2}^{k=n-p-1} \alpha_k t^{-i,k-1} \tag{2.18}$$

where α_k are the suitable weights assigned to the surplus generated when a total of k agents are removed from the system. By using different α_ks, we get different mechanisms. The HETERO mechanism satisfies individual rationality, and feasibility, as proven in, [95, Conjecture 1]. Further, HETERO is worst case optimal and has redistribution index same as WCO as proven in [99, Proposition 2].

In the next section, we consider the problem of resource allocation where money is not involved. In this setting, we aim to find allocations that are fair. We discuss many fairness notions discussed in literature and algorithms proposed to find fair allocations.

2.5 Part II - Mechanism Design without Money (Fair Resource Allocation)

In many applications like inheritance division, course allocation, and division of resources or tasks among the workforce, we encounter the problem of resource division. In contrast to the auction setting considered previously, here, the goal is to divide resources among the interested agents optimally without any monetary transactions. Furthermore, we consider that the allocation satisfies certain fairness constraints so that each agent is ensured its fair share. Most of the literature assumes that agent valuations are public knowledge and thereby focuses on preference aggregation. In other words, many existing works propose meaningful fairness notions and algorithms to achieve allocations that satisfy them. Nevertheless, in reality, the agents' valuations are often private. The agents may manipulate the existing algorithms to maximize their utility at the cost of overall fairness. This section discusses the possibility/impossibility of having an algorithm that cannot be manipulated for certain fairness notions. First, we introduce some notations required.

2.5.1 Fair Resource Allocation Environment

Consider the problem of division of indivisible resources. We represent each instance by $\langle N, M, V \rangle$ which are formally defined below,

- Finite set of agents $N = \{1, \ldots, n\}$
- Finite set of indivisible goods $M = \{1, \ldots, m\}$
- Valuation functions V where $v \in V$, $v = (v_1, \ldots, v_n) = (v_i, v_{-i})$ denotes a particular profile and $\forall i \in N$, $v_i : 2^M \to \mathbb{R}_+$ and v_{-i} be the valuation profile of all agents excluding i
- The valuations are normalized, i.e., $v_i(\emptyset) = 0$. We denote the valuation of item $k \in M$ for any agent $i \in N$ as $v_i(\{k\})$ or v_{ik}

- We assume v_i is monotonic, $\forall i \in N, \forall S \subseteq T \subseteq M, v_i(S) \leq v_i(T)$
- The set of all possible complete allocations, $\mathcal{A}$, $A \in \mathcal{A}$ denotes a specific allocation and A_i is allocation per agent. A_{-i} denotes allocation of all agents except i
- We only allow complete allocation, and no two agents can receive the same item. That is, $A = (A_1, A_2, \ldots, A_n)$, s.t., $\forall i, j \in N$, $i \neq j; A_i \cap A_j = \emptyset$ and $\bigcup_i A_i = M$

Different Types of Valuation Functions

Most of the literature assumes that the valuation functions of agents are *monotonic* for goods. The utility to an agent for a bundle increases every additional good in the bundle. More formally,

Definition 2.19 (*Monotonicity*). *A valuation function v_i is monotonic if, $\forall S \subseteq T \subseteq M, v_i(S) \leq v_i(T)$.*

In general, the agents derive certain valuation for every possible subset of goods i.e., all possible bundles. Representing such a rich valuation function requires exponential space hence it is common to consider specific kinds of valuation functions. The most popular ones are additive and identical valuations.

Definition 2.20 (*Additive Valuations*). *Agents have additive valuations if, $\forall i \in N$ values any non-empty bundle A_i as $v_i(A_i) = \sum_{k \in A_i} v_{ik}$.*

A valuation instance is said to be identical when all agents have the same valuation for all subsets of items, formally,

Definition 2.21 (*Identical Valuations*). *The valuations are identical if, $\forall i, j \in N, \forall S \subseteq M, v_i(S) = v_j(S)$.*

Often, agents do not have identical valuations, but they order the items likewise, i.e., agents have the same rank for the items. We call such valuations Identical Ordering (IDO).

Definition 2.22 (*Identical Ordering (IDO)*)**.** *Valuations are IDO when all agents agree on the same ranking of the items, i.e., for* $\forall i \in N$, $v_{i1} \geq v_{i2} \ldots \geq v_{im}$.

The cardinal cost functions of agents in an IDO instance may still differ. In approval-based settings, agents have binary valuations, where they either approve or disapprove an item.

Definition 2.23 (*Binary Valuations*)**.** *The valuations are binary, if* $\forall i \in N$, $\forall k \subseteq M$, $v_{ik} \in \{1, 0\}$.

Given the basic components for resource allocation, we now state the common fairness notions considered in the literature below.

2.5.2 Fairness Notions and Algorithms

Researchers have studied fair division formally since the 1940s. The problem of cake cutting, where a heterogeneous continuous resource has to be divided among multiple agents. Further, each agent must be guaranteed a fair share of the cake, which led to proportional and envy-free cake cutting. There are various approaches proposed for fair division in the case of divisible goods. Although in many real-world situations like course allocation, division of inheritance, and divorce settlement, the goods may not be divisible. Here we are focusing on indivisible resources and studying the fairness notions of envy-freeness, proportionality and max-min share. Since allocations that satisfy these notions may not exist, there are relaxations and corresponding algorithms to achieve them.

2.5.2.1 Envy Freeness and its Relaxations

The concept of envy-freeness (EF) introduced by Foley 1966 [78] is a well-established notion of fairness. It ensures that no agent envies the bundle of any other agent. Unfortunately, an EF allocation of indivisible items may not exist; for example, when there is a single good and two agents, the agent who doesn't get the good feels envious of the agent that does. The researchers were interested in relaxing the concept of EF to EF1 and EFX in order to limit the envy of every agent. For goods, an allocation is EF1 when all agents value their bundle at least as much as they value another agent's bundle with their most valued item removed. EFX is stronger than EF1 and requires that all agents value their bundle no less than the other agents' bundle with their least valued item removed. A standard definition is as follows,

Definition 2.24 (*Envy-free (EF) and relaxations*)**.** *For the items (goods or chores), an allocation A that satisfies $\forall i, j \in N$,*

$$\begin{aligned} &v_i(A_i) \geq v_i(A_j) \ \textit{is}\ \text{EF} \\ &v_i(A_i) \geq v_i(A_j \setminus \{k\}); \forall k \in A_j \ \textit{is}\ \text{EFX} \\ &v_i(A_i) \geq v_i(A_j \setminus \{k\}); \exists k \in A_j \ \textit{is}\ \text{EF1} \end{aligned} \tag{2.19}$$

We now explore some existential results and algorithms for finding EFX allocations. Then we see that relaxing EFX to EF1 guarantees existence. Further, we describe some polynomial time algorithms for finding EF1 allocations.

Envy-Freeness up to any item (EFX)

As given by Definition 2.19, in the case of goods, EFX implies that for any pair of agents i and j, if i envies j, the envy can be eliminated by hypothetically removing any good from j's bundle. EFX is a very compelling notion of fairness however, its existence is still an open problem in fair division. It is known that EFX allocation always exists in the case of three agents with additive valuations [50] and two agents with general monotone valuations [164]. Despite the ongoing efforts, the question of EFX's existence remains unanswered for any valuation system that involves more than three agents.

Plaut et al. 2020 [164] showed that when agents have *general but identical valuations*, a modification of the Leximin, i.e, leximin++ solution is EFX. The Leximin++ (Algorithm

1) selects the allocation that maximizes the minimum individual utility; further, if multiple allocations achieve it, it chooses the allocation that maximizes the size of the minimum utility agent's bundle. Further, if multiple allocations achieve it, it chooses the allocation that maximizes the second minimum individual utility, followed by the cardinality maximum of the second minimum agent's bundle, and so forth. Even when two agents have identical submodular valuations, Plaut et al. 2020 [164] showed that finding EFX takes exponential time. On the other hand, the algorithm of Lipton et al. 2004 [130] finds an EF1 allocation in polynomial time for any number of agents with monotone valuations. Thus, EFX is indeed significantly stronger than EF1. In addition, Plaut et al. 2020 [164] proposed an algorithm for EFX allocation in polynomial time, i.e., $\mathcal{O}(mn^3)$ when agents have additive valuations with identical ranking, which relies on envy-cycle elimination [130].

Algorithm 1 Leximin++

```
Set $\forall i, A_i = \emptyset$
$A \leftarrow \max_{A' \in \prod_n(M)} \min_{i \in N} v_i(A'_i)$
while $|A| > 1$ do
    $i \leftarrow \min_{j \in N} v_j(A_j)$
    $A \leftarrow \max_A |A_i|$
    $A \leftarrow \max_{A' \in A} \min_{i \in N} v_i(A'_i)$
end whilereturn Allocation $A$
```

Envy-Freeness up to one item (EF1)

As defined in Definition 2.19, in the case of goods, Envy-free up to one item (EF1) implies that for any pair of agents i and j, if agent i envies agent j, the envy can be eliminated by virtually taking out agent i's most valuable item from j's bundle, i.e., $v_i(A_i) \geq v_i(A_j \setminus \{g\})$, $\exists g \in A_j$. EF1 allocation always exists and can be obtained in polynomial time.

Theorem 2.8 (Budish, Lipton et al. 2011, 2004 [43, 130])**.** *EF1 allocation always exists and can be found in polynomial time.*

When agents have additive valuations, round-robin algorithm gives EF1 for goods in polynomial time, i.e., $\mathcal{O}(mn \log m)$ [47]. We formally describe the steps in Algorithm 2.

Algorithm 2 Round Robin

```
Set $\forall i, A_i = \emptyset$
Every agent sort the items in decreasing order
Arrange agents in an arbitrary sequence
while $M \neq \emptyset$ do
    for $i \leftarrow 1$ to $n$ do
        $A_i \leftarrow A_i \cup \max_{k \in M} v_{ik}$
        $M \leftarrow M \setminus \{k\}$
    end for
end whilereturn Allocation $A$
```

When agents have *general monotone valuations*, Lipton et al. 2004 [130] proposed an *envy-cycle elimination algorithm* that gives EF1 allocation for goods in polynomial time $\mathcal{O}(mn^3)$. This algorithm bounds the envy of any agent by the maximum marginal value of any good; it corresponds to the notion of EF1 for goods. We define the envy graph first. An envy graph of allocation A consists of nodes for each agent and directed edges from agent i to agent j if i envies j, i.e., $v_i(A_i) < v_i(A_j)$. The algorithm selects an unenvied agent in each iteration, i.e., an agent with no edges directed towards them, and assigns an arbitrary good. If there are no such agents, there must be cycles of envy, and the agents can exchange bundles until no more cycles remain. Upon receiving the good, other agents may envy this agent; we can eliminate this envy by removing the good they just received since they were previously unenvied. Thus, if agent i envy agent j, the envy is bounded up to one good, i.e., by the recently added good in agent j's bundle. After each round of partial allocation, this algorithm ensures that EF1 is satisfied. In contrast, Bérczi et al. 2020 [29] showed that the envy-cycle elimination algorithm fails to find an EF1 allocation when agents have general non-monotone valuations.

Algorithm 3 Envy-Cycle Elimination

```
Set $\forall i, A_i = \emptyset$
for $k \leftarrow 1$ to $m$ do
    Find an unenvied agent $i$
    $A_i \leftarrow A_i \cup k$
    if Envy-Cycle exists then
        Swap bundles to resolve
    end if
end forreturn Allocation $A$
```

2.5.2.2 Proportionality and its Relaxations

In addition to EF and its relaxations, proportional allocations are also well studied in the context of fairness. As introduced by Steihaus 1948 [180], Proportionality requires that each agent gets at least $1/n$ of their share of the total value. Furthermore, even proportional allocations may not always exist; for this reason, researchers have considered relaxations. Similar to EF, we have proportionality up to one item, PROP1 requires that every agent is guaranteed to receive their proportionality guarantee if they lose their least valued chore or receive their most valuable good from any other agent's bundle. Proportionality up to any item, PROPX requires that every agent is guaranteed to receive their proportionality guarantee if they lose their most valued chore or receive their least valued good allocated to another agent.

Definition 2.25 (*Proportionality (PROP)* [14, 58, 180])**.** *For the items (chores or goods), an allocation A that satisfies $\forall i \in N$,*

$$\begin{aligned}
&v_i(A_i) \geq 1/n \cdot v_i(M) \textit{ is } \text{PROP} \\
&v_{ik} < 0, v_i(A_i \setminus \{k\}) \geq 1/n \cdot v_i(M); \forall\ k \in A_i \textit{ is } \text{PROPX} \\
&v_i(A_i \setminus \{k\}) \geq 1/n \cdot v_i(M); \exists\ k \in A_i \textit{ is } \text{PROP1}
\end{aligned} \tag{2.20}$$

Proportionality up to one item (PROP1):

Conitzer et al. 2017 [58] introduced the notion of PROP1 in the setting of Public Decision Making, which is more generic than indivisible item allocations. PROP1 requires each agent to receive a utility at least their proportional share if we add the largest good allocated to another agent to their bundle, as stated in Definition 2.25. Each agent receives its proportional share after hypothetically including one extra good from another agent's allocation into its bundle.

In the setting of Public Decision Making, Conitzer et al. 2017 [58] shows that a PROP1 allocation always exists and can be found in polynomial time. When agents have subadditive valuations, Envy-freeness is a stronger notion than proportionality, i.e., EF implies PROP. Similarly, when agents have additive valuations, EF1 implies PROP1. It may seem counter-intuitive at first glance that EFX implies PROPX when agents have additive valuations.

2.5.2.3 Maxmin Share Allocations

We explore Maximin share (MMS) introduced by Budish 2011 [43], extending the concept of Cut and Choose to indivisible goods. Let's say we ask an agent to divide m items into n bundles and take the bundle that they're least interested in. This risk-averse agent would divide the bundles to maximize the minimum utility, which is the MMS share of the agent. An MMS allocation guarantees every agent their MMS share. Note that MMS is a weaker fairness property than proportionality; a proportional allocation is always MMS. Even though MMS is weaker, MMS is still too demanding in the case of indivisible items.

Definition 2.26 (*Maxmin Share MMS* [43]). *An allocation A is said to be MMS if $\forall i \in N, u_i(A_i) \geq \mu_i$, where*

$$\mu_i = \max_{(A_1,A_2,\ldots,A_n)\in\prod_n(M)} \min_{j\in N} u_i(A_j)$$

An MMS allocation exists in instances with two agents with additive valuations using the ”Cut-and-Choose” protocol but Kurokawa et al. 2018 [128] presented an intricate example in which every allocation fails to achieve MMS guarantees for more than two agents. Bouveret et al. 2016 [41] showed that an MMS allocation need not exist when agents have general valuations, even in the case of two agents.

Despite its appealing formulation, MMS does have a computational disadvantage. For one, even the computation of the maximin share for an agent with additive valuations is an NP-Complete problem [41]. And computing an MMS allocation is strongly NP-Hard. The problem is weakly NP-hard even for two agents [15, 41]. However, a PTAS for computing MMS exists [198]. Originally, Woeginger 1997 [198] gave PTAS to compute a maximum partition for a particular agent within the context of job scheduling. However, the problem is identical to computing a maximin partition for the given agent.

2.5.3 Efficiency Notions

In the previous subsection, we discussed some popular fairness notions. Note that not assigning any item to any agent is trivially EF. However, we also desire efficiency, so fairness is considered in connection with efficiency criteria as well. One of the most frequently studied efficiency criteria in fairness literature is Pareto-Optimality. A Pareto optimal (PO) allocation ensures that there is no other allocation which Pareto dominates, i.e., better for all agents and strictly better for at least one. It is interesting to consider PO and fair allocations.

Definition 2.27 (*Pareto-Optimal (PO)*). *An allocation A' is said to Pareto dominate allocation A, if $\forall i \in N$, $v_i(A'_i) \geq v_i(A_i)$ and $\exists i \in N$, $v_i(A'_i) > v_i(A_i)$. When there is no other allocation that dominates allocation A, it is said to be Pareto-optimal.*

We then consider utilitarian welfare, the sum of agents' utilities. On the other hand, Nash welfare corresponds to the product of agents' utilities and egalitarian welfare, to the minimum of individual agents' utility.

Definition 2.28. *Given an instance $(N, M, \mathcal{V})$, an allocation A^* satisfies,*

Maximum Utilitarian Welfare, $MUW(u)$, *if*

$$A^* \in \max_A \sum_{i=1}^{n} u_i(A_i) \tag{2.21}$$

Maximum Nash Welfare, $MNW(u)$ *if*

$$A^* \in \max_A \prod_{i=1}^{n} u_i(A_i) \tag{2.22}$$

Maximum Egalitarian Welfare, $MEW(u)$ *if*

$$A^* = \max_A \min_i u_i(A_i) \tag{2.23}$$

2.5.4 Strategyproof Fair Allocations

In fair resource allocation, the majority of the algorithms proposed in the literature do not assume strategic agents. Hence the social planner must know the valuations of the agents upfront to ensure fair allocations using such algorithms. Although in reality, agents may misreport their true valuations. As described in Section 2.3, to ensure truthfulness of agents, we must look for incentive compatible mechanisms 2.8. Incentive compatible mechanisms are also referred to as strategyproof mechanisms. In fairness literature, there are two kinds of strategyproof mechanisms explored, i) *Deterministic Strategyproof Mechanisms* and ii) *Randomized Strategyproof Mechanisms*. We define them formally given a resource allocation instance $\langle N, M, V \rangle$ and $A \in \mathcal{A}$ denotes a specific allocation. We define a direct mechanism which given the agent valuations v, determines allocation, i.e., $h_i(v_1, \ldots, v_n) = A_i, \forall i \in N$.

Definition 2.29 (*Deterministic Strategyproof Mechanism* (DSM))**.** *A direct mechanism (Definition 2.5) h is called strategyproof if,*

$$\forall v_1, \ldots, v_n, \forall i, \forall v_i^{'} : v_i(h_i(v_1, \ldots, v_n)) \geq v_i(h_i(v_1, \ldots, v_i^{'}, \ldots, v_n))$$

Definition 2.30 (*Group Strategyproof Mechanism* (GSM))**.** *A d direct mechanism h is called strategyproof iff for each subset of agents $S \subseteq N$ and valuation profile $v^{'}$, where $v_i^{'} = v_i, \forall i \notin S$,*

$$\forall v, v^{'}, \exists i \in S, : v_i(h_i(v)) \geq v_i(h_i(v^{'}))$$

Definition 2.31 (*Randomized Strategyproof Mechanism* (RSM))**.** *A randomized direct mechanism h is called strategyproof if,*

$$\forall v_1, \ldots, v_n, \forall i, \forall v_i^{'} : \mathbb{E}\left[v_i(h_i(v_1, \ldots, v_n))\right] \geq \mathbb{E}\left[v_i(h_i(v_1, \ldots, v_i^{'}, \ldots, v_n))\right]$$

The expectation is over the randomness of the mechanism.

Impossibility Results

Unfortunately, in fairness literature, there are a lot of impossibility results known for DSM for certain fairness notions. In Lipton et al. 2004 [130], the authors prove that it is impossible to design a truthful mechanism that achieves minimum envy or EF by providing a counterexample. We will see the example in detail in Chapter 5. Menon et al. 2017 [142] prove that it is impossible to have any DSP which is (even approximately) proportional and ensures complete allocation. By complete allocation, we mean non-wastefulness i.e., all resources are allocated. Amanatidis et al. 2016 [10] prove that for 2 agents, there is no truthful mechanism that ensures better than $\frac{1}{m/2}$-MMS allocation, m is the total number of items. On the other hand, a strategyproof mechanism for Pareto Optimal (PO) allocations is very simple. Serial dictatorship is the DSM that ensures PO.

Criteria	DSM	RSM
EF + EQ	✓	✓
PO + EF	✓	✓
PO + EQ	✗	✗
PO + EF + EQ	✗	✗
PROP + PO	✓[2]	✓
PROP + EF	✗	✓
PROP + EQ	UNK	✓
PROP + PO + EF	✗	UNK
PROP + PO + EQ	✗	✗
PROP + EF + EQ	✗	✓
PROP + PO + EF + EQ	✗	✗

Table 2.2: Summary for Strategyproof Fair Mechanisms (Divisible Resources)

Apart from DSM, researchers have identified Randomized SP mechanisms (RSM) to ensure fairness and efficiency. Chen et al., Mossel et al. 2013, 2010 [52, 145] show that there is a randomized SP mechanism that always returns an allocation which is EF and EQ. The Table 2.2 summarizes the known results for the existence of RSM and DSM for various fairness notions. In the table, UNK refers to unknown, i.e., the existence of such a mechanism is not known. From the table, we see that there exist deterministic strategyproof mechanisms for allocations that are both EF and EQ or PO and EQ or PROP and PO. Whereas for the rest either there are randomized SP mechanisms or impossibilities. Given these impossibilities for finding fair allocations in general valuations, researchers look for DSMs in restricted valuation domains like binary valuations.

Binary Valuations

Halpern et al. 2020 [104] provide a group strategy-proof mechanisms for binary additive valuations. Their mechanism is based on maximizing the Nash social welfare (i.e., the geometric mean of the agents' valuations) with a lexicographic tie-breaking rule. Nash optimal allocations are Pareto efficient. Also, under binary additive valuations, such allocations are known to be MMS as well as EF1. Hence, for binary additive valuations, the work of Halpern et al. 2020 [104] achieves all the three desired properties; their mechanism in fact can be executed in polynomial time. For the broader class of binary submodular valuations, Babaioff et al. 2021 [17] obtain a truthful, PO, and fair mechanism. This work considers Lorenz domination as a fairness criterion and, hence as implications, obtains EF1 and 1/2-MMS guarantees. Further strengthening the claim, Barman et al. 2022 [23] shows that under matroid-rank valuations it is possible to have a group strategyproof mechanism that is PO and EF1.

In the above two sections, we have seen some examples of mechanisms with money i.e., redistribution mechanisms and mechanisms without money for fair allocations. In all the examples, the mechanisms are designed analytically and proven to satisfy the required properties. Designing mechanisms analytically is very challenging for complex real-world scenarios. To overcome this issue, researchers have proposed *Automated Mechanism Design*. In the next section, we elaborate on this and provide some applications.

2.6 Automated Mechanism Design

Typically mechanisms are designed to achieve specific objective under certain conditions. For example,

- VCG mechanism only maximizes social welfare and cannot be extended to any other objective
- Myerson's expected revenue maximization is only for single item

- Redistribution mechanisms proposed are worst-case optimal i.e, redistribute maximum money in the worst case. It is challenging to design these mechanism that perform well in expectation

In 2002, Conitzer and Sandholm introduced *Automated Mechanism Design* (AMD) approach, where the mechanism is computationally created for the specific problem instance at hand [59]. Compared to analytically designing mechanisms, AMD has several advantages: 1) it can yield better mechanisms than the ones known to date, 2) it is often applicable to complex real-world setting, and 4) it inculcates learning in machine using existing data instead of manual efforts. The problem is modelled as a optimization problem,

Definition 2.32 (*Automated Mechanism Design* [171])**.** *In an automated mechanism design setting, there are a finite set of agents N, and a finite set of outcomes X. For each agent $i \in N$, a finite set of types Θ_i sampled from the distribution Φ_i and a utility function $u_i : \Theta \times X \rightarrow \mathbb{R}$. Finally there is an objective function that the social planner aims to maximize along with certain constraints.*

The objective function can aim to maximize social welfare, revenue or any other metric as desired. At the same time, each agent is maximizing its own utility. Here we consider AMD for designing mechanisms with money. For this specific case, the AMD requires appropriate outcome selection and payment selection functions that maximize the objective while satisfying certain basic constraints of IR and DSIC. More formally,

Definition 2.33 (*AMD with Money* [171])**.** *A deterministic mechanism with payments consists of an outcome selection function $f : \Theta \rightarrow X$ and for each agent i and a payment selection function $t_i : \Theta \rightarrow \mathbb{R}$, where $t_i(\Theta)$ gives the payment made to agent i when the reported types are Θ. When the ex-post IR and DSIC constraints can be given by the following, for any agent i*

$$\forall \theta \in \Theta, u_i(\theta_i, f(\Theta)) + t_i(\theta)) \geq 0 \quad \textit{(ex-post IR)}$$

$$\forall \theta \in \Theta, \forall \theta_i^{'} \in \Theta_i^{'}, u_i(\theta_i, f(\theta_i, \theta_{-i})) + t_i(\theta_i, \theta_{-i})) \geq u_i(\theta_i, f(\theta_i^{'}, \theta_{-i})) + t_i(\theta_i^{'}, \theta_{-i})) \quad \textit{(DSIC)}$$

Designing a deterministic mechanism is easy if the designer's objective is social welfare (the VCG mechanism suffices), but NP-complete more generally (for example, if the objective is to maximize the expected revenue collected from the bidders [59]—as is the objective in some auctions). All of these hardness results apply even with a uniform prior over types. Given the complexity results, we use learning-based approaches to obtain a good empirical approximations of the optimal solutions. We first discuss the basic framework required to apply learning based approaches before discussing existing work on the same. First we discuss the basic components in Deep Learning Models.

2.6.1 Deep Learning Models

Neural Networks are biologically inspired paradigms which learn optimal functions from data. The main components that customize such a network for a specific task are its architecture and the objective function which guides its training. Neural networks have been successful in learning complex, nonlinear functions accurately, given adequate data [110]. There is a theoretical result which states a neural network can approximate any continuous function on compact subspace of $\mathbb{R}^n$ [112]. However, designing such a network has been elusive until recent times. The latest theoretical developments ensure that stochastic gradient descent (SGD) converges to globally optimal solutions [125, 179]. In recent times, with advent in computing technology, neural networks have become one of the most widely used learning models. They have outperformed many of the traditional models in the tasks of classification and generation etc [108, 92]. We now discuss the basic components of neural networks.

2.6.1.1 Perceptron

The most fundamental unit of a deep neural network is called an artificial neuron. McCulloch (neuroscientist) and Pitts (logician) proposed a highly simplified computational

$$g(x) = \sum_{i=0}^{m} w_i \cdot x_i, \;\; x_i \in \mathbb{R}$$

$$y = \phi(g(x)) = 1 \quad \text{if } g(x) \geq \theta$$

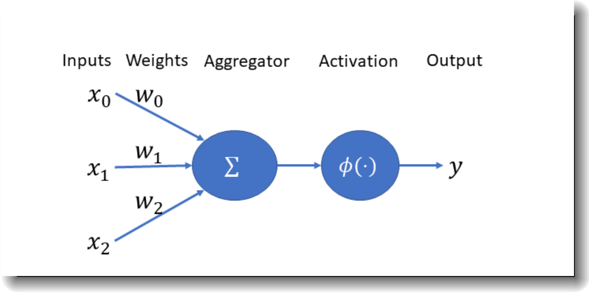


Figure 2.2: Simple Perceptron

model of the neuron (1943). It consists of an aggregator function g that aggregate the boolean inputs, $x = (x_0, \ldots, x_m), x_i \in \{0, 1\}$. The function ϕ takes the aggregated output and makes a binary decision $y \in \{0, 1\}$. More formally,

$$g(x) = \sum_{i=0}^{m} x_i, \;\; x_i \in \{0, 1\}$$

$$y = \phi(g(x)) = 1 \quad \text{if } g(x) \geq \theta$$

Frank Rosenblatt, an American psychologist, proposed the classical perceptron model (1958) which is more general computational model than McCulloch–Pitts neurons. In this model the inputs are not restricted to be binary and weights are introduced. More formally, it takes $x = (x_0, \ldots, x_m), x_i \in \mathbb{R}$ as inputs and outputs y with weight w_i associated with i^{th} input. The activation function ϕ is for thresholding the output to introduce nonlinearity in the network. The formal definition is given beside an illustration in Figure 2.2.

Now we provide the *Perceptron Learning Algorithm* to learn the weights given (x, y).

Perceptron Learning Algorithm

Algorithm 4 Perceptron Learning Algorithm

```
Input: $(X, Y)$
Let $X^1$ set of samples with $y = 1$
Let $X^0$ set of samples with $y = 0$
Initialize $w$ randomly
while !convergence do
    Pick a random sample $x \in X^1 \cup X^0$
    if $x \in X^1$ and $\sum_{i=0}^{m} w_i x_i < 0$ then
        $w = w + x$
    end if
    if $x \in X^0$ and $\sum_{i=0}^{m} w_i x_i \geq 0$ then
        $w = w - x$
    end if
end while
The algorithm converges when all inputs are correctly classified
```

Using the simple model of Rosenblatt's perceptron, we now look at the algorithm to learn the weights w_i. The w_i's are learnt such that the inputs x are correctly classified according to the given labels y. When the two classes are linearly separable, the perceptron algorithm (Algorithm 4) converges in finite number of steps as stated and proven in [37]. Although real-world data is complex and noisy, hence the different classes may not be linearly separable. Towards this the Rosenblatt's perceptron is extended to include *non-linear activations* and multi-layered neuron stacked together as we discuss below.

2.6.1.2 Multi-Layered Perceptrons

In general to mimic complex functions, a simple perceptron is not sufficient. Firstly, the basic thresholding function ϕ is modified, which hard threshold's the aggregator output w.r.t., θ. Different non-linear thresholding functions or also known as *activation* functions are discussed below.

Non-Linear Activations. There are numerous activation functions proposed in the literaure. The goal of these functions is to transform the output from the aggregator function g, to the final output which takes values only within a specified range. The hard thresholding function used in the Rosenblatt's perceptron only returns either 1 or 0 based on the threshold θ. Here we discuss some popular activation functions.

- *Linear.* The simplest activation function is linear activation i.e., effectively having no activation function, it is an identity function where $a = \sum_i w_i, x_i$, $\phi^{Linear}(a) = a$.

- *Sigmoid.* The sigmoid or logistic function outputs a value in the range $[0, 1]$, given the input x and $a = \sum_i w_i, x_i$, the function is defined as,

$$\phi^{sigmoid}(a) = \frac{1}{1 + \exp(-a)}$$

  The output is smoother compared to the threholding function and can be interpreted as probability. The function is smooth continuous and differentiable. Although the gradients vanish for very large or very small values of the input.

- *Tanh* The tanh function outputs a value in the range $[-1, 1]$, given the input x and $a = \sum_i w_i, x_i$, the function is defined as,

$$\phi^{tanh}(a) = \frac{\exp(a) - \exp(-a)}{\exp(a) + \exp(-a)}$$

  The output is smoother compared to the threholding function and the mean of the activations is centered around 0 unlike sigmoid. The function is smooth continuous and differentiable yet shares the same issue of vanishing gradients as sigmoid.

- *ReLU* The most widely used activation function currently is ReLU. It is faster to compute and its gradients do not saturate. Although not as smooth but still works in practice,

  $$\phi^{ReLU}(a) = \max(0, a)$$

  The only disadvantage is that the derivative is equal to zero when the input is negative. The problem is known as the dying Relu. If the weights in the network always lead to negative inputs into a Relu neuron, that neuron won't be effectively contributing to the network training. To overcome this, researchers have proposed leakyReLU which is given by $\phi^{LReLU}(a) = \max(0.01 \cdot a, a)$.

- *Softmax.* The softmax activation function is used in neural networks when we want to build a multi-class classifier which solves the problem of assigning an instance to one class when the number of possible classes is larger than two(otherwise we can simply use sigmoid if possible classes=2). When $a_i = \sum_j w_{ji} x_j$, softmax is formally given by,

  $$\phi_i^{softmax} = \frac{\exp{(a_i)}}{\sum_j \exp((a_j))}$$

Given various non-linear activations, we now look at how to stack multiple perceptrons to obtain feed-forward neural networks.

Feed-forward Network

Consider that the input x is an m-dimensional vector and we want an output which is k-dimensional (k-class classification) and we stack the perceptrons in $L-1$ hidden layers. Such an organization is referred to as *Multi-layered Perceptrons* (MLPs) or *Feed-forward Neural Networks* (NN). Each hidden layer consists of n neurons each. The input layer can be called the 0^{th} layer and the output layer is the L^{th} layer. At each neuron, there is an aggregation step and then the non-linear activation is applied. Let $h^l(x)$ represent the vector of neuron at any step $l \in 0, 1, \ldots, L$. Therefore, $h^0(x) = x$ and $h^L(x) = f(x)$

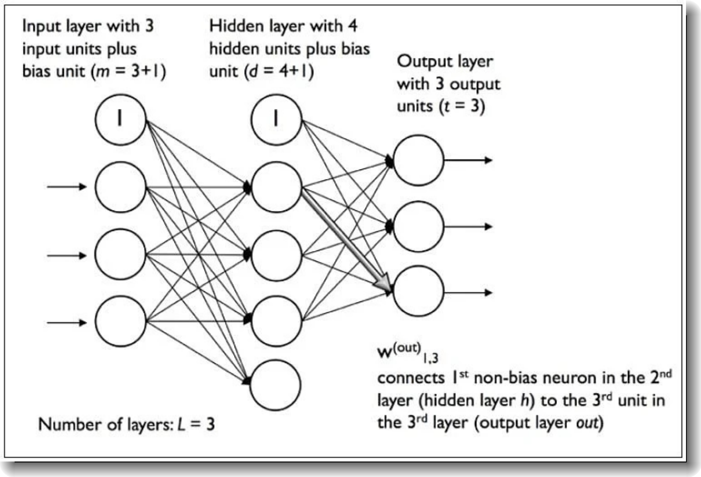

Input layer with 3 input units plus bias unit (m = 3+1)
Hidden layer with 4 hidden units plus bias unit (d = 4+1)
Output layer with 3 output units (t = 3)
1
1
$w^{(out)}{}_{1,3}$
connects 1st non-bias neuron in the 2nd layer (hidden layer h) to the 3rd unit in the 3rd layer (output layer out)
Number of layers: L = 3

Figure 2.3: Feed Forward Neural Network

assuming that the network learns the function f to map x to y. For any hidden layer $h^l(x) = (h_1^l(x), \dots, h_n^l(x))$ assuming n neurons in all the hidden layers,

$$h_j^l(x) = \phi(\sum_{i \in n} w_i^{(l-1,j)} h_i^{l-1}(x) + b_j^l), \quad l \in \{1, \dots, L-1\}, \forall j \in [n]$$

where ϕ is any non-linear activation function described above, $b^l = (b_1^l, \dots, b_n^l)$ is the bias added at layer l and $w^{(l,j)} = (w_1^{(l,j)}, \dots, w_n^{(l,j)})$ is the weights with which the output at layer $l-1$, i.e., $h^{l-1}(x)$ is multiplied to obtain the value of j^{th} neuron at l^{th} layer. Let W^l denote all the weights connecting layer l to $l+1$, i.e., $W^l = [w_i^{(l,j)}]_{(i,j)}$, where $i, j \in [n]$, or is referred to as the weight matrix. The final function learnt by the network is,

$$f(x; W, b) = \phi^L(W^{L-1} \cdot h^{L-1}(x) + b^L)$$

where ϕ^L represents the final activation typically a *Softmax Function* for multi-class classification or even a linear activation function for regression. The illustration for the described feed forward network with 1 hidden layer is given in Figure 2.3, containing $m = 3$ input features and $n = 4$ hidden neurons and $k = 3$ output classes. We next state a powerful theorem that discusses the representational power of MLPs.

Representation power of MLPs

The theorem states that, there is a guarantee that for any function $f^r(x) : \mathbb{R}^n \to \mathbb{R}^m$ we can always find a neural network (with 1 hidden layer containing enough neurons) whose output $f(x)$ satisfies $|f(x) - f^r(x)| < \epsilon$. More formally,

Theorem 2.9 (Cybenko 1989 [61])**.** *A multi-layer network of neurons with a single hidden layer can be used to approximate any continuous function to any desired precision.*

The proof for the theorem is given in [111, 61].

Despite such strong theorem, it is crucial to note that, it is practically challenging to design and learn the parameters of a single layered network that can approximate any function. Hence, the neural networks consists of many hidden layers and are trained using *Gradient Descent*. Below we discuss different steps involved in the learning the parameters (W, b) i.e. the weights and biases of the NNs.

2.6.1.3 Learning of MLP Parameters

Given the input, target pairs $(X, y) = \{(x_1, y_1) \ldots, (x_n, y_n)\}$, we consider the task of classification, where $y_i \in \{0, 1\}$. We build an MLP which learns the function parameterized by (W, b), $f_{W,b}(x_i) = \hat{y}_i$ such that $\hat{y}_i \in \{0, 1\}$. We first consider an objective function that we need to minimize to ensure y and $\hat{y}$ are close. We consider the binary cross-entropy loss given by,

$$\mathcal{L}_{(W,b)}(f(X; W, b), y) = -\frac{1}{n} \sum_i y_i \log \hat{y}_i$$

The goal is to find optimal (W^*, b^*) that minimizes $\mathcal{L}$, hence the objective is to $\min_{W,b} \mathcal{L}_{(W,b)}$. One of the approaches to minimize a given function is the *Gradient Descent Method* [3]. As its name suggests, gradient descent involves calculating the gradient of the target function. The pseudo code for the algorithm is given in Algorithm 5.

Algorithm 5 Gradient Descent

```
Input: $(X, Y)$, Learning Rate: $\eta$
Initialize $t = 0$, $W_t$ and $b_t$
while $t{+}{+} <$ max iterations do
    $W_{t+1} = W_t - \eta \nabla_{W_t} \mathcal{L}$
    $b_{t+1} = b_t - \eta \nabla_{b_t} \mathcal{L}$
end while
```

In order to compute the gradients w.r.t the weights and biases at a given layer l, i.e., $(\nabla_{W^l}\mathcal{L}, \nabla_{b^l}\mathcal{L})$ we apply the chain rule as follows,

$$\left(\frac{\partial \mathcal{L}}{\partial W^l} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial h^{L-1}} \frac{\partial h^{L-1}}{\partial h^{L-2}} \cdots \frac{\partial h^{l+1}}{\partial W^l}, \quad \frac{\partial \mathcal{L}}{\partial b} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial h^{L-1}} \frac{\partial h^{L-1}}{\partial h^{L-2}} \cdots \frac{\partial h^{l+1}}{\partial b^l} \right)$$

Computing the gradients of the weights at a layer at layer l requires the gradients of later layers w.r.t the corresponding next layer till the very end. Hence the gradients are computed starting from the last layer, which popularly known as *Back-propagation*. Notice that the gradient descent algorithm (Algorithm 5) goes over the entire data once before updating the parameters for computing ∇_W, ∇_b. Imagine, a dataset of millions of samples, the algorithm must make millions of calculations every step. Hence in practice an approximation to this approach is used which is called *Stochastic Gradient Descent* (SGD) [38]. It is a stochastic approximation of gradient descent, that reduces the high computational burden, achieving faster iterations in trade for a lower convergence rate. Essentially at each iteration, we sample a mini-batch of training samples and compute the gradient w.r.t. the mini-batch. Hence the time required for each iteration significantly reduces, although lot many iterations are required compared to gradient descent. Bottou et al. 2008 [39] provide proven arguments to justify the practicality of SGD. Hence the final algorithm for NN training using SGD is given by Algorithm 6.

Algorithm 6 Stochastic Gradient Descent

```
Input: $(X, Y)$, Learning Rate: $\eta$
Initialize $t = 0$, $W_t$ and $b_t$                                ▷ Weight Initialization
while $t{+}{+} <$ max iterations do
    Sample a mini-batch of samples $\{(x_1, y_1), \dots, (x_b, y_b)\} \subset (X, Y)$
    Compute $f(x_i; W, b), \forall i \in \{1, \dots, b\}$              ▷ Forward Pass
    Compute gradient estimates
        $\hat{g} = \frac{1}{b} \nabla_{(W,b)} \sum_{i=1}^{b} \mathcal{L}(f(x_i; W, b), y_i)$
                                                                   ▷ Backward Pass
    $[W_{t+1}, b_{t+1}] = [W_t, b_t] - \eta \hat{g}$                   ▷ Weight Update
end while
```

It mostly consists of the following four important steps,

- ***Weight Initialization.*** It is used to define the initial values for the parameters in neural network models prior to training the models on a dataset. Historically, weight initialization involved using small random numbers, although over the last decade, more specific heuristics have been developed that use information, such as the type of activation function that is being used and the number of inputs to the node. When activation function like Sigmoid or Tanh are used, then *Xavier Weight Initialization* [86] and *Normal Xavier Weight initialization* are used. With ReLU, *He Weight Initialization* is used [107]. For the weights connecting layer l having n_{in} number of nodes to layer $l+1$ having n_{out} number of nodes, the weights are sampled in the following way,
    - Xavier: $W \sim Unif\left[-\frac{1}{\sqrt{n_{in}}}, \frac{1}{\sqrt{n_{in}}}\right]$
    - Normalized Xavier: $W \sim Unif\left[-\frac{\sqrt{6}}{\sqrt{n_{in}+n_{out}}}, \frac{\sqrt{6}}{\sqrt{n_{in}+n_{out}}}\right]$

- He: $W \sim \mathcal{N}(0, \sqrt{2/n_{in}})$

The above initializations help the network to converge to good solutions.

- ***Forward Pass.*** It refers to calculation process, values of the output layers from the inputs data. It's traversing through all neurons from first to last layer to obtain the function $f(X; W, b)$ given the inputs, weights and biases as described before. A loss function is calculated from the output values.

- ***Backward Pass.*** It refers to the computation of gradients which is made from the last layer, backward to the first layer.

- ***Weight Update.*** After computing the gradients, this is the crucial step from gradient descent that updates the weights in the negative direction of the gradients. This is to minimize the overall loss. The learning rate specifies the magnitude of the descent step in the negative direction of the gradients. In practice, there are better optimizers proposed in the literature to overcome training challenges in deep neural networks, described in detail in [90][Chapter 8]. In addition to the simple step performed in SGD, the *Momentum Method* ([166]) and *Nestrov Accelerated Momentum* [184] are designed to accelerate learning. To further improve the optimizers, researcher proposed adaptive learning methods, which adaptively modify the learning rate and training progresses individually for each parameter. The most popular of these are AdaGrad ([68]), RMSProp ([188]) and Adam ([127]) optimizers.

Now that we have studied some of the basic components involved in training of a neural network, we now try to place it in the context of Automated Mechanism Design (AMD). In the next section, we study how to model AMD using neural networks before discussing some recently proposed approaches for deep learning based mechanism design.

2.6.2 AMD via Deep Learning

In this subsection, we discuss the basic framework for learning any mechanism via DL. This framework is not the absolute framework that one must follow but it is a first attempt at formalizing a mechanism design problem in terms of machine learning framework. The following are the major components involved,

- ***Input Data.*** In various auctions, the samples are based on the type profile of the agents. Hence it is very common the use the valuations θ as the input to the network and for each valuation θ , the network maps it to a particular outcome which may correspond to (k, t) or the utility value obtained by the agent. Since, real world data is not available for auctions, it is common to assume that the valuations θ come from a predefined distribution. It is very common to assume the valuations are independent and identically distributed, since most machine learning frameworks assume that the input samples are i.i.d from a certain distribution. Although in [176], the authors explore the case where the valuations are correlated. Such assumption of the distribution is definitely a shortcoming and real-world data would be more reflective and useful for better results. It is also a challenge to give combinatorial valuations as an input since the size can become exponential.

- ***Model Architecture.*** The model architecture is a very crucial part which not only models the complexity of the mechanism learnt. It is also possible to hard-wire certain game theoretic constraints like DSIC into the architecture like in [69, 89, 136]. Although such hard-wiring restricts the application of the network only to a specific domain for a specific setting. Considering that the input is the valuation which is defined over a specific number of agents and specific number of items, a different network is required when the number of agents or the items change if using fully-connected network. In [185], the authors explore Convolutional Neural Network (CNN) so that the network is independent of the input size. The network outputs, the allocation and payment for a certain input valuation.

- ***Loss Function.*** The loss function is the primary part where the game-theoretic goals are captured. If the social planner wishes to maximize welfare, or revenue, it can be calculated based on the network output and the loss would then be defined appropriately. In [69], the authors have also introduced constraints like DSIC into the loss by using Lagrangian multipliers. In [143], the authors introduce fairness and efficiency constraints. The optimization of the loss is then performed using Stochastic Gradient Descent (SGD) or one of its suitable variants as common in general deep learning algorithms.

- ***Evaluation Metrics.*** The evaluation is very specific to the mechanism that is being designed. The optimal revenue or social welfare finally reflects how well the network has learnt. Along similar lines, it is also common to introduce ratio of obtained result to the best (if known) when the mechanism is approximately optimal.

Equipped with basic framework, we describe relevant work that apply neural networks to learn specific mechanisms which are analytically difficult to design.

2.6.3 Existing Literature

We list the following papers we see how we can leverage neural networks for analytically challenging problems in mechanism design. We also emphasize the basic modelling of the network and key ideas used in each of the papers to achieve the desired results.

Optimal Auctions through DL ([69])

Optimal auctions are used to ensure high revenue across industries. Typical to an auction setting, the bidders are strategic and have private valuations which they do not reveal. The auctioneer wants to implement an SCF that maximizes revenue and is DSIC. We have Myerson optimal auction for single item case under the assumption of quasilinear utilities. The problem is still analytically challenging even for two bidders and two items. In this paper, the authors provide DL-based models *RochetNet* and *RegretNet* for solving the multi-item optimal auction design.

The problem set-up has n bidders and $m \in M$ items. Each bidder is assumed to have a valuation function which gives the value the bidder has for any subset of items, $v_i : 2^M \to \mathbb{R}_{\geq 0}$. The valuation profiles $v \in V$ is drawn independently from the known distributions $F = (F_1, \ldots, F_n)$. The valuations of the bidders are assumed to be additive i.e., $v_i(S) = \sum_{j \in S} v_i(\{j\})$ or unit demand i.e., $v_i(S) = \max_{j \in S}(\{j\})$. The bidders report their valuations untruthfully, their bidding profile denoted by $b = (b_1, \ldots b_n) \in V$. The auction mechanism in the quasilinear environment is considered to have an allocation function $g(b) : V \to 2^M$ and payments made by the bidders denoted by $p(b) : V \to \mathbb{R}_{\geq 0}$. The utility received by each agent of type v_i when bidding b is given by $u_i(v_i, b) = v_i(g_i(b) - p_i(b))$. In this paper, the aim is to learn (g, p) using through neural networks such that they satisfy the following desirable properties.

- Revenue Optimal, i.e., minimize the negative expected revenue

$$-\mathbb{E} \sum_{i \in N} p_i(v)$$

- DSIC to ensure the utility of each agent is maximized on truthful reporting

$$u_i(v_i; (v_i, b_{-i})) \geq u_i(v_i; (b_i, b_{-i})), \forall v_i, \forall b, \forall i$$

- Ex-post Individual Rationality (IR) to ensure that the each agent receives non-zero utility,

$$u_i(v_i; (v_i, b_{-i})) \geq 0, \forall v_i, \forall b_{-i}, \forall i$$

RochetNet. This network is designed for single-bidder multi-item setting. From [170], it is already known that non-decreasing convex utility functions give DSIC mechanisms. Hence the authors design a specific architecture of the network described below to ensure

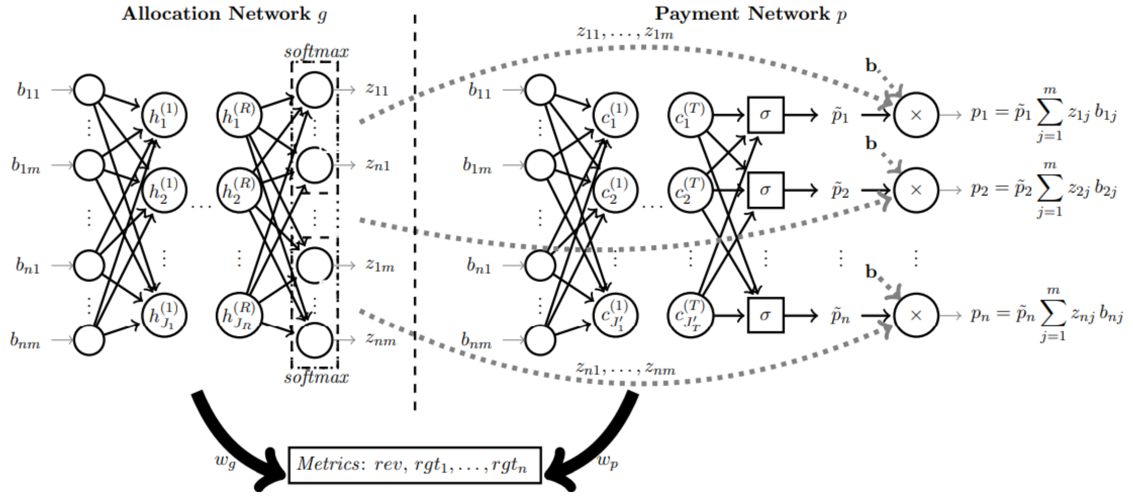


Figure 2.4: RegretNet for additive bidders [69]

the utility is non-decreasing and convex. Given b as the input the network with weights $w = (\alpha, \beta)$ computes the utility,

$$u^{\alpha,\beta} = \max\left\{\max_j\{\alpha_j.b + \beta_j\}, 0\right\}$$

Since by the characterization from [170], the above network already ensures DSIC, it is left to train the network such that the revenue is maximized. For that, the following loss is used. We know that $u(b) = \sum_{j=1}^m g_j(b) - p(b)$, hence $g^w(b) = \nabla u^{\alpha,\beta}(b) = \alpha_{j^*(b)}$, $p^w(b) = \nabla u^{\alpha,\beta}(b).b - u^{\alpha,\beta}(b) = \beta_{j*(b)}$ where $j^*(b) \in \text{argmax}_j \alpha_j b + \beta_j$. Replacing the argmax with softmax following loss is obtained,

$$\mathcal{L}(\alpha,\beta) = -\mathbb{E}_{v\in F}[\sum_j \beta_j \tilde{\nabla}_j(v)], \quad \tilde{\nabla}_j(v) = \text{softmax}_j(\{\alpha_j.b + \beta_j\})$$

The above network is trained using stochastic gradient descent (SGD) for the above loss.

RegretNet. In the above setting DSIC was ensured due to the architectural design. In this network, the authors learn optimal auctions for more general settings with more than one bidder and combinatorial valuations. RegretNet has two fully connected networks, i) allocation network $g^w : \mathbb{R}^{nm} \to [0,1]^{nm}$ and ii) payment network $p^w : \mathbb{R}^{nm} \to \mathbb{R}^n \geq 0$. Proper constraints are imposed on the networks to ensure items are not over allocated (introduce softmax in the last layer) and the payments are non-negative as given in Figure 2.4. There is no DISC constraint in architecture therefore the notion of regret is introduced which is then used in the loss. The empirical form of regret is defined as follows,

$$r\hat{g}t_i(w) = \frac{1}{L}\sum_{l=1}^{L}\left[\max_{v_i' \in V_i} u_i(v_i^{(l)}; (v_i', v_{-i}^l)) - u_i^w(v_i^{(l)}; v^{(l)})\right]$$

Hence the loss function will be constrained with DISC, i.e. expected regret for each bidder should be 0.

$$\begin{aligned} \max_{w} \quad & \mathbf{E}_{v\sim\mathcal{F}}\Big[-\sum_{i\in[n]} p_i\Big] \\ \text{s.t.} \quad & rgt_i(w) = 0, \forall i \in [n] \end{aligned}$$

So the neural network will be optimize over the following Lagrangian function,

$$C_p(w,\lambda) = \mathbf{E}_{v\sim\mathcal{F}}\Big[-\sum_{i\in[n]} p_i\Big] + \sum_{i\in[n]} \lambda_i \, \widehat{rgt_i(w)} + \frac{\rho}{2}\Big(\sum_{i\in[n]} \widehat{rgt_i(w)}\Big)^2$$

where $\lambda \in \mathbf{R}^n$ is a vector of Lagrange multipliers, and $\rho > 0$ is a fixed parameter that controls the weight on the quadratic penalty. The loss function defined is not convex hence, the global convergence is not guaranteed but the empirical results show effective performance.

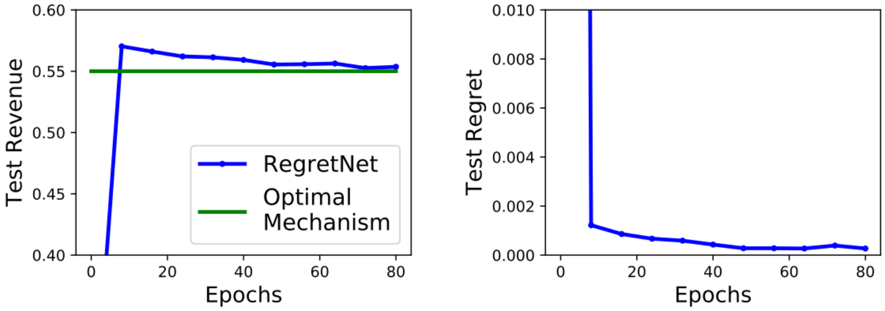
Results. The authors reproduce known optimal results for the following two settings given by Manelli-Vincent and Pavlov auctions [134, 163].

(A) $n = 1$ and $m = 2$ for additive valuations, the valuations are drawn independently from $U[0,1]$

(B) $n = 1$ with unit-demand valuations over $m = 2$, the items are drawn independently from $U[2,3]$

Distribution	Opt	RegretNet		RochetNet
	rev	*rev*	*rgt*	*rev*
Setting A	0.550	0.554	< 0.001	0.550
Setting B	2.137	2.137	< 0.001	2.136

(a)



(b)

Figure 2.5: (a) Test revenue and regret for RegretNet and revenue for RochetNet for A and B. (b) Test revenue and regret w.r.t. training epochs for A with RegretNet [69]

From Figure 2.5, it can be seen the both the networks recover the optimal revenue with $< 1\%$ error.

When $n = 1$ and arbitrary number of items, Straight Jacket Auction (SJA) is proposed in [84]. It is shown that *RochetNet* gives the optimal revenue for $m \leq 6$ and revenue that matches SJA for $m = 7, 8, 9$ and 10. The authors then go on to find optimal revenue for settings upto 5 additive bidders and 10 items with ease although there are no known results for comparison.

Automated Mechanism Design via Neural Networks [176]

In the paper, as discussed in the previous section [69], the authors follow two approaches to ensure DSIC for designing revenue optimal auctions. The first approach of hard-wiring the DSIC constraint into the network (*RochetNet*) requires a lot of domain knowledge and the network architecture will not be generalizable. In the second approach they introduce IC as a soft-constraint which would mean that the network can still produce mechanisms that are not IC. In order to overcome this issue, the authors in [176], represent the mechanism as a menu i.e., a list of (valuation, outcome) tuples in the single buyer case. According to the taxation principle [192], by simply letting the buyer do the selection the mechanism is IC. There are no known exact mechanisms for the following two scenarios with single buyer and two items, i) Revenue optimal mechanisms when the menu size is restricted to a constant. ii) Revenue optimal mechanisms when the valuations for the two items are correlated.

A naive mechanism is defined by a set of actions also known as the menu items are represented by a pair of $[x, p]$ where x is the allocation vector $x \in \{0, 1\}^m$ and p is the payment $p \in \mathbb{R}_+$ from the buyer to seller. The utilities are assumed to be quasi-linear and valuations of the buyers are additive. Simply letting the buyer do a selection from the menu-items is enough to ensure IC [192]. It is challenging to design the menu items such that the revenue of the seller is maximized. The authors use the following network to design optimal revenue mechanism.

Network Architecture. The architecture as given in Figure 2.6, consists of two networks, the mechanism network and the buyer network. The mechanism architecture is a fully-connected network that outputs a set of menu items with a constant vector as input. The output has two parts, first is an allocation matrix X with m rows (number of items) and k columns (number of menu items). The second is a payment vector of length k representing the price for each of the menu items. The second network is the buyers network which maps a mechanism (in this case a set of menu items obtained as output from the mechanism network) to a buyers strategy $s(v)$, i.e., the menu item the buyer chooses based on his valuations. The output of the buyers network is $m + 1$-dimensional with the first m dimensions representing the value the buyer has for each of the m items and the last dimension consists of the probability vector over the k menu items. In the buyers network

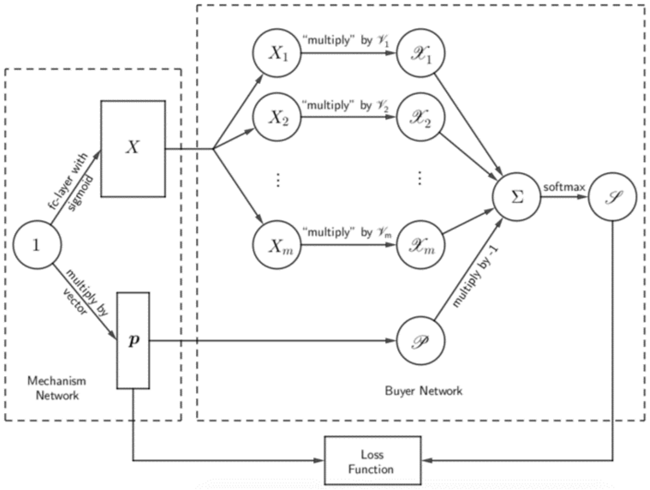


Figure 2.6: Network Architecture: Mechanism and Buyer network [176]

there are no weights learnt, it is simply multiplying each X_i with the valuation v_i i.e the value obtained from the i^{th} item for an allocation X and is represented by $\mathcal{X}$. Similarly the payments are also determined and represented by $\mathcal{P}$, the final utility is obtained by $\sum_{i \in m} \mathcal{X}_i - \mathcal{P}$. Finally applying softmax over the utility values, helps decide the menu item that is chosen based on the one that gives highest utility denoted by $s(v)$. In order to train the network for maximizing revenue the following loss is used, $Loss = -\sum_{v \in V} Pr[v] p^T s(v)$ Given that the distributions from which the valuations are drawn is predefined, it is easy to compute $Pr[v]$. The authors then proceed to find optimal deterministic mechanisms for various settings and provide theoretical proofs for the mechanisms found using neural networks.

DL for Multi-Facility Location Mechanism Design [89]

In certain settings, it is unethical to charge money to the agents involved. For example, in this paper the authors consider the setting where given a space $\Omega \subseteq \mathbb{R}^d$, the problem of locating the space to construct public facilities. In order to overcome the Gibbard-Sattherwaite impossibility, the agents' preferences over the locations are assumed to be single-peaked. When $d = 1$ and only one facility to be constructed i.e., $k = 1$, [148] proves that choosing the median of the agents' peaks is the strategy-proof (DSIC) mechanism that also minimizes the sum of the agents' distances to the outcomes (social cost). Yet there are many worst-case approximation results for $k > 2$. In this paper, the authors use deep learning to design a strategy-proof mechanism to predict locations that also minimizes the social cost.

The problem more formally is set up considering N agents $\{1, 2, ...N\}$, set of location $\Omega = [0, 1]$, K facilities. Each agent has single peaked preference over Ω. For each agent i, utility will be the based on the facility that was placed closet to its preferred location. i.e. $u_i(o) = \max_{k \in K} u_i(o_k)$ .We assume $u(x) = -|x - a|$ where $a = \tau(u)$ is the peak. The paper purposes MoulinNet for single facility problem and RegretNet-nm for general mechanisms.

MoulinNet. For single facility location Moulin in [148] provides the following result. Hence, the architecture of the MoulinNet uses the following to ensure strategy-proofness.

Theorem 2.10. *A unanimous mechanism $f : U \to \Omega$ is strategy-proof iff it is generalized median rule, i.e, for each $S \subseteq \{1, \ldots, n\}$ there exists some $a_S \in \Omega$ for all $(u_1, \ldots, u_n) \in U$,*

$$f(u) = \underset{S\ \in\{1,\ldots,n\}}{min} \max \left\{ \underset{i \in S}{max}\ \tau(u_i), a_S \right\}$$

By exploiting the above theorem, the following is the network with weights (w, b) ,

$$f^{w,b}(u) = \underset{S\ \in\{1,\ldots,n\}}{min} \left\{ \underset{i \in S}{max}\ \{\tau(u_i), h^{w,b}(v(S))\} \right\}$$

In the above, $a_S = h^{w,b}(v(S))$, the input to the network is $v(S)$ where S is represented using binary vector $x \in \{-1, 1\}^n$, with $x_i = 1$ iff $i \in S$. The network $h^{w,b}$ is designed to be a monotonic function. The parameters are optimized with the social cost as the loss function, which is estimated from the sample S.

RegretNet-nm. In this network, the strategy-proofness is not encoded into the network architecture but added as a loss. Hence this network can be generalized to more than one facility location. For n agents and K facilities, the network $f^w(u)$ is fully-connected with input as the agents peaks $\tau(u_1), \ldots, \tau(u_n)$ and the output is the K facilities. The empirical loss for social cost that the network is trained to minimize is given as below,

$$\mathcal{L}(w) = \frac{1}{Rn} \sum_{j=1}^{R} \sum_{i=1}^{n} u_i^j(f^w(u^j))$$

In order to ensure DSIC, the following notion of regret is introduced,

$$rgt_i(f) = \mathbb{E}_{u \sim D} \left[\underset{u_i' \in U_i}{max}\ u_i(f(u_i', u_{-i})) - u_i(f(u_i, u_{-i})) \right]$$

The final objective is to introduce the empirical version of the constraint in the loss by using Lagrangian multiplier. The loss is then minimized using SGD.

The authors compare their approach with standard mechanisms from literature. Both the networks yield similar performance as the best percentile rule [182]. There are no known results when the agents' peaks are drawn from non-product distributions but the flexibility of RegretNet allows it to perform well even in such scenarios.

Chapter 3

Redistribution Mechanism

Consider a social setting where public resources/objects are to be allocated among competing and strategic agents so as to maximize social welfare (the objects should be allocated to those who value them the most). This is called allocative efficiency (AE). We need the agents to report their valuations for obtaining these resources, truthfully referred to as dominant strategy incentive compatibility (DSIC). Typically, we use auction-based mechanisms to achieve AE and DSIC. However, due to Green-Laffont Impossibility Theorem, we cannot ensure budget balance in the system while ensuring AE and DSIC. That is, the net transfer of money cannot be zero. This problem has been addressed by designing a redistribution mechanism so as to ensure minimum surplus of money as well as AE and DSIC. Designing redistribution mechanisms which perform well in expectation becomes analytically challenging for heterogeneous settings.

We train a neural network to determine an optimal redistribution mechanism. We also propose a loss function to train a neural network to optimize the worst case. We design neural networks with the underlying rebate functions being linear as well as nonlinear in terms of bids of the agents. We observe that a neural network based redistribution mechanism for homo-

geneous settings which uses nonlinear rebate functions outperforms linear rebate functions when the objective is optimal in expectation. Our approach also yields a redistribution mechanism that is optimal in expectation for heterogeneous settings.

3.1 Introduction

We address the problem of allocating public resources/objects among multiple agents who desire them. These strategic agents have their private values for obtaining resources. The allocation of the objects should be such that society, as a whole, gets the maximum benefit. That is, the agents who value these resources the most should get them. This condition is referred to as *Allocative Efficient (AE)*. To achieve this, we need the true valuations of the agents, which the strategic agents may misreport for personal benefit. Thus, there is a need for an auction-based mechanism. A mechanism ensuring truthful reporting is called *Dominant Strategy Incentive Compatible* (DSIC). The classical Groves mechanisms [94] satisfy both of these properties.

Groves mechanisms achieve DSIC by charging each agent an appropriate amount of money known as Groves' payment rule. The most popular among Groves mechanism is VCG mechanism [54, 94, 190]. Use of VCG mechanism results in the collection of money from the agents. It should be noted that our primary motive in charging the agents is to elicit their true valuations and not to make money from them as the objects are public. Hence, we need to look for the other Groves mechanisms. Moreover, the mechanism cannot fund the agents. Thus, we desire a mechanism that incurs neither deficit nor surplus of funds; it must be *Strictly Budget Balanced* (SBB). However, due to Green-Laffont Impossibility Theorem ([93]), no mechanism can satisfy AE, DSIC, and SBB simultaneously. Thus, any Groves payment rule always results in either a surplus or deficit of funds.

To deal with such a situation, Maskin et al. 1979 [138] suggested that we first execute VCG mechanism and then redistribute the surplus among the agents in a manner that does not violate DSIC. This mechanism is referred to as a *Groves' redistribution mechanism* or simply *redistribution mechanism* (RM) [95, 99] and the money returned to an agent is called its *rebate*. The rebates are determined through *rebate functions*. Thus, designing an RM is the same as designing an underlying rebate function.

In the last decade, a lot of research focused on dealing with the Green-Laffont Impossibility theorem and on designing an optimal redistribution mechanism (RM) that ensures the maximum possible total rebate [48, 55, 75, 95, 96, 98, 99, 102, 100]. An optimal RM could be optimal in expectation or optimal in the worst case. The authors of [102, 103, 149] address the problem of finding the optimal RM when all the objects are identical (homogeneous). Guo and Conitzer [101, 103] model an optimization problem and solve for an optimal linear rebate function which guarantees maximum rebate in the worst-case (WCO) and optimal in expectation for a homogeneous setting. The authors of [95, 99] extend WCO to a heterogeneous setting where the objects are different and propose a nonlinear rebate function called as *HETERO*. Despite HETERO being proved to be optimal for unit demand in heterogeneous settings, in general, it is challenging to come up with a nonlinear RM analytically. Analytical solutions for optimal in expectation RMs in heterogeneous settings are elusive. Moreover, the possibility of a nonlinear rebate function which is optimal in expectation for a homogeneous setting has not been explored yet. Thus, there is a need for a new approach towards designing RMs. In this work, we propose to use neural networks and validate its usefulness.

Our Contributions. To the best of our knowledge, this is the first attempt towards learning optimal redistribution mechanisms (RM) using neural networks. This work has been further extended by [97, 185, 195].

- To begin with, we train neural networks for the settings where researchers have designed RMs analytically. In particular, we train networks, OE-HO-L and OW-HO-L for optimal in expectation for homogeneous settings with linear rebate functions and optimal in the worst case for homogeneous settings with linear rebate functions respectively. Both neural networks match the performance of theoretically optimal RMs for their respective settings.

- Next, we train a network, OW-HE-NL to model the nonlinear rebate function for optimal in worst case RM in heterogeneous settings, discarding the need to solve it analytically. Note that, traditionally, neural networks have been mostly used for stochastic approximation of an expectation, but our model is also able to learn a worst-case optimal RM as well.
- Motivated by the network performance above, we train OE-HO-NL, an optimal in expectation RM with nonlinear rebate function for the homogeneous setting. We find that this model ensures a greater expected rebate than the optimal in expectation RM with linear functions, proposed by Guo et al. 2008 [101].
- We also train OE-HE-NL, an optimal in expectation for heterogeneous settings with nonlinear functions and we experimentally observe that its performance is reasonable.

Related Work

To obtain the required private information from strategic agents truthfully, mechanism design theory is developed [62, 63, 160]. The key idea is to charge the agent appropriately to make mechanisms truthful or DSIC. The most popular auction-based mechanisms are VCG and Groves mechanisms [54, 94, 190], which satisfy the desirable properties, namely, allocative efficiency (AE) and dominant strategy incentive compatibility (DSIC). Another desirable property is the net transfer of the money in the system should be zero, i.e., it should be *strictly budget balanced* (SBB). Green et al. 1979 [93] showed no mechanism can satisfy AE, DSIC, and SBB simultaneously. As we cannot compromise on DSIC, we must compromise on one of AE or SBB.

Faltings 2005 [75] and Guo et al. 2008 [100] achieved budget balance by compromising on AE. Hartline et al. 2008 [106] proposed a mechanism that maximizes the sum of the agents' utilities in expectation. Clippel et al. 2014 [55] used the idea of destroying some of the items to maximize the agents' utilities, leading to approximately AE and approximately SBB. A completely orthogonal approach was proposed by Parkes et al. 2001 [161], where the authors propose an optimization problem which is approximately AE, SBB and though not DSIC, it is not easy to manipulate the mechanism. However, an aggressively researched approach is to retain AE and DSIC and design a mechanism that is as close to SBB as possible. These are called *redistribution mechanisms* (RM).

Maskin et al. 1979 [138] first proposed the idea of redistribution of the surplus as far as possible after preserving DSIC and AE. Bailey 1997 [21], Cavallo 2006 [48], Moulin 2009 [149], and Guo et al. 2007 [102] considered a setting of allocating p homogeneous objects among n competing agents with unit demand. Guo et al. 2009 [103] generalized their work in [102] to multi-unit demand to obtain worst-case optimal (WCO) RM. In [101], the authors designed RM that is optimal in expectation for homogeneous settings.

Gujar et al. 2011 [95] proved that no linear RM can assure non-zero rebates in worst case and then generalized WCO mechanism mentioned above to heterogeneous items, namely HETERO. Their conjecture that HETERO was feasible and worst-case optimal was proved by Guo 2012 [99] for heterogeneous settings with unit-demand.

3.2 Preliminaries

Let us consider a setting comprising p public resources/objects and n competing agents who assign a certain valuation to these objects. Each agent desires at most one out of these p objects. These objects could be homogeneous, in which case, agent i has valuation $v_i = \theta_i$ for obtaining any of these p resources. It could also be the case that the objects are distinct or heterogeneous and each agent derives different valuation for obtaining different objects ($v_i = (\theta_{i1}, \theta_{i2}, \ldots, \theta_{ip})$). These objects are to be assigned to those who value it the most, that is, it should be allocatively efficient (AE). The true values $\mathbf{v} = (v_1, v_2 \ldots, v_n)$ that the agents have for the objects are based on their private information $\theta = (\theta_1, \theta_2, \ldots, \theta_n) = (\theta_i, \theta_{-i})$ and the strategic agents may report them as $(\theta'_1, \ldots, \theta'_n)$. In the absence of appropriate payments, the agents may boast their valuations. Hence, we charge agent i a payment $m_i(\theta')$ based on the reported valuations.

We need to design a mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{P})$, an allocation rule $\mathcal{A}$ and a payment rule $\mathcal{P}$. $\mathcal{A}$ selects an allocation $k(\theta') \in K$ where K is the set of all feasible allocation and $\mathcal{P}$ determines the payments. With this notation, we now explain the desirable properties of a mechanism.

3.2.1 Desirable Properties

One of our primary goals is to ensure allocative efficiency.

Definition 3.1 (*Allocative efficiency* (AE))**.** - *A mechanism $\mathcal{M}$ is allocatively efficient (AE) if it chooses in every given type profile, an allocation of objects among the agents such that sum of the valuations of the allocated agents is maximized. That is, for each $\theta \in \Theta$,*

$$k^*(\theta) \;\in\; \underset{k \in K}{\operatorname{argmax}} \sum_{i=1}^{n} v_i(k, \theta_i).$$

The most desirable property is that the agents should report their valuations truthfully to the mechanism. Formally, it is called *dominant strategy incentive compatibility* (DSIC).

Definition 3.2 (*Dominant Strategy Incentive Compatibility* (DSIC):)**.** *We say a mechanism $\mathcal{M}$ to be dominant strategy incentive compatible (DSIC), if it is the best response for each agent to report their type truthfully, irrespective of the types reported by the other agents. That is,*

$$v_i(k(\theta_i, \theta_{-i}), \theta_i) - m_i(\theta_i, \theta_{-i}) \geq v_i(k(\theta'_i, \theta_{-i}), \theta_i) - m_i(\theta'_i, \theta_{-i})$$

$$\forall \theta'_i \in \Theta_i, \forall \theta_i \in \Theta_i, \forall \theta_{-i} \in \Theta_{-i}, \forall i \in N.$$

Given an AE allocation rule, Groves proposed a class of mechanisms known as *Groves' mechanisms* that ensure DSIC. Groves' payment rule is $m_i(\theta) = -\sum_{j \neq i} v_j(k_*(\theta), \theta_j) + h_i(\theta_{-i})$, where h_i is an arbitrary function of reported valuations of the agents other than i. Clarke's payment rule is a special case of Groves' payment rule where $h_i(\theta_{-i}) = \sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j)$ where k^*_{-i} is an AE allocation when the agent i is not part of the system. Thus, Clarke's payment rule is given by,

$$t_i(\theta) = \sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j) - \sum_{j \neq i} v_j(k^*(\theta), \theta_j) \ \forall i = 1, \ldots, n \tag{3.1}$$

And, $m_i = t_i$. This payment scheme is referred to as *VCG payment*. The total payment by all the agents is, $t(\theta) = \sum_{i \in N} t_i(\theta)$

Another property we desire is budget balance condition.

Definition 3.3 (*Budget Balance* or *Strictly Budget balance* (SBB))**.** *We say that a mechanism $\mathcal{M}$ is strictly budget balanced (SBB) if for each $\theta \in \Theta, m_1(), m_2(), \ldots, m_n()$ satisfy the condition, $\sum_{i \in N} m_i(\theta) = 0$. It is weakly budget balanced if $\sum_{i \in N} m_i(\theta) \geq 0$.*

One can implement AE allocation rule and charge the agents VCG payments. In the case of auction settings, the seller collects the payments. In our setting, the goal is not to make money as these objects are public resources. However, in general, due to Green-Laffont Impossibility Theorem [93], no AE and DSIC mechanism can be strictly budget balanced. That is, the total transfer of money in the system may not be zero. So, the system will be either left with a surplus or incur a deficit. Using Clarke's mechanism, we can ensure under fairly weak conditions, that there is no deficit of money (that is, the mechanism is weakly budget balanced) [54]. The idea proposed by [138] is to design a payment rule to first collect VCG payments and then redistribute this surplus (rebate) among the agents while ensuring DSIC. This leads to *Groves' Redistribution Mechanism*, in which the rebate is given by a rebate function.

Example 7: Redistribution Mechanism Homogeneous Unit Demand

Consider n agents and p units of a resource, each agent requires one unit of the resource. Hence consider allocation in the non-trivial case when $p < n$. According to the linear rebate function, the rebate

$$r_i = c_0 + c_1 v_1 + \ldots + c_{i-1} v_{i-1} + c_i v_{i+1} + \ldots + c_{n-1} v_n$$

where $v_1 \geq v_2 \geq \ldots \geq v_n$. We consider the following mechanism,

Bailey-Cavallo Mechanism. In this mechanism $c_{p+1} = p/n$, $c_i = 0$ for all other i. The rebate is thus given by,

$$r_i = \frac{m}{n} v_{m+2} \quad i \leq m+1, \quad r_i = \frac{m}{n} v_{m+1} \quad i > m+1$$

The total money redistributed is $(m+1)\frac{m}{n}v_{m+2} + (n-m-1)\frac{m}{n}v_{m+1}$. Since $r_i \geq 0$ for all agents, the mechanism satisfies IR. Further since $(m+1)\frac{m}{n}v_{m+2} + (n-m-1)\frac{m}{n}v_{m+1} \leq n\frac{m}{n}v_{m+1} = mv_{m+1}$, hence it is also Feasible.

- Best Case Rebate - When $v_{m+1} = v_{m+2}$, the above mechanism returns mv_{m+1} which is the total VCG payment collected, i.e., 100% redistribution in the best case.

- Worst Case Rebate - When $v_{m+2} = 0$, the money redistributed back is $\frac{n-m-1}{n}$ of the total VCG payment collected.

In general, the interest is to find mechanisms which maximize the worst-case refund or the expected refund as described further in the sections below.

3.2.2 Existing Approaches

Groves' Redistribution Mechanism: Since SBB cannot coexist with DSIC and AE, we would like to redistribute the surplus to the participants as much as possible, preserving

DSIC and AE. Such a mechanism is referred to as *Groves redistribution mechanism* or simply *redistribution mechanism*. Designing a redistribution mechanism involves designing an appropriate rebate function. We desire a rebate function which ensures maximum rebate (which is equivalent to minimum budget imbalance). In addition to DSIC, we want the redistribution mechanism to have the following properties:

1. **Feasibility** (F). The total payment to the agents should be less than or equal to the total received payment.

2. **Individual Rationality** (IR). Each agent's utility by participating in the mechanism should be non-negative.

3. **Anonymity**. Rebate function is same for all the agents, $r_i() = r_j() = r()$. This may still result is different redistribution payments as the input to the function may be very different.

While designing redistribution mechanism for either homogeneous or heterogeneous objects, we may have linear or nonlinear rebate function of the following form,

Theorem 3.1 (Gujar et al. 2011 [95]). *In the Groves redistribution mechanism, any deterministic, anonymous rebate function f is DSIC iff,*

$$r_i = f(v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n) \; \forall i \in N$$

where, $v_1 \geq v_2 \geq \ldots \geq v_n$.

Definition 3.4 (*Linear Rebate Function*). *The rebates to an agent follow a* linear rebate *function if the rebate is a linear combination of bid vectors of all the remaining agents. Thus, $r_i(\theta, i) = c_0 + c_1 v_{-i,1} + \ldots + c_{n-1} v_{-i,n-1}$.*

There may exist a family of redistribution mechanisms which satisfy the above constraints, but the aim is to identify the one mechanism that redistributes the greatest fraction of the total VCG payment. To measure the performance of redistribution mechanism [103], defines redistribution index,

Definition 3.5 (*Redistribution Index*). *The redistribution index of a redistribution mechanism is defined to be the worst case fraction of VCG surplus that gets redistributed among the agents. That is,*

$$e^{ow} = inf_{\theta:t(\theta)\neq 0}\frac{\sum r_i(\theta_{-i})}{t(\theta)}$$

With the notation defined above and Green-Laffont Impossibility theorem in the backdrop, we first explain a redistribution mechanism in Example 3.2.1

Optimal Redistribution Mechanisms

It may happen that, one mechanism might redistribute higher rebate at θ_1 and another mechanism at θ_2. Hence, we use the two kinds of evaluation metrics defined to select a mechanism. One metric compares the rebate functions based on maximum expected total redistribution, to find the mechanism which is optimal in expectation. The other metric finds the optimal in worst-case redistribution mechanism, based on the lowest redistribution index it guarantees

Optimal in Expectation (OE)

If the prior distributions over agents' valuations are available we can compare the mechanisms based on total expected redistribution. In [101] the authors derive the mechanism and prove its optimality for homogeneous settings with linear rebate function. We define a redistribution index for OE setting as follows:

$$e^{oe} = \frac{\mathbb{E}\sum_i r_i(\theta_{-i})}{\mathbb{E}\sum_\theta t(\theta)}$$

Maximizing e^{oe} is equivalent to maximizing the expected total rebate. The authors formulated the problem as given in Theorem 3.1.

The OE objective for heterogeneous objects as well as with nonlinear rebate function has not been addressed yet.

Table 3.1: Optimization problem formulation

	OE	OW
Variables :	$c_0,\ c_1,....,c_{n-1}$	$e^{ow}, c_0,\ c_1,....,c_{n-1}$
Maximize :	$\mathbb{E}\sum_{i=1}^{n} r_i$	e^{ow}
Feasibility :	$\sum_{i=1}^{n} r_i$ ¡= t	$\sum_{i=1}^{n} r_i \leq t$
Other constraints		
For worst-case :		$\sum_{i=1}^{n} r_i \geq e^{ow}t$
IR :		$r_i \geq 0$

Optimal in Worst-case (OW)

The redistribution mechanism is better if it ensures higher rebate to the agents on average. In the absence of distributional information, we would evaluate a mechanism by considering the worst redistribution index that it guarantees. In [103] the authors gave the following model and analytically solved it for homogeneous setting with linear rebate functions. They also claim the worst-case optimal mechanism is optimal among all redistribution mechanisms that are deterministic, anonymous and satisfy DSIC, AE and F. The optimization problem is formulated as given in Table 3.1. For heterogeneous settings, Gujar et al. 2011 [95] defines a nonlinear redistribution mechanism which is called HETERO and Guo 2012 [99] proves the optimality of HETERO. There is no optimal mechanism with linear rebate function for heterogeneous settings as established by the following theorem,

Theorem 3.2 (Gujar et al. 2011 [95]). *If a redistribution mechanism is feasible and individually rational, then there cannot exist a linear rebate function which is simultaneously DSIC, deterministic, anonymous and has a non-zero redistribution index.*

Equipped with the knowledge of the existing approaches and the above theorem, we describe our approach for designing the optimal rebate function using neural networks.

Our Approach. As mentioned, for a homogeneous setting the linear rebate functions that are OE and OW can be analytically found by formulating a redistribution mechanism as a linear program. However, for heterogeneous settings, a linear redistribution mechanism need not be a good choice (Theorem 3.2). Even though Guo 2012 [99] has solved for redistribution mechanisms in heterogeneous by proving HETERO to be OW, an OE mechanism for heterogeneous settings has not been formulated yet. Moreover, OW mechanism (HETERO) is not simple to describe. In addition, for homogeneous settings, it is not known whether nonlinear redistribution mechanisms can do better than linear for OE objective.

We address these issues, using a novel data-driven approach to approximate a rebate function for a given setting, without analytically solving for it. That is, we generate a large number of bid profiles randomly and train a neural network to determine the rebates for the agents so as to achieve the given objective, either OE or OW. We consider both the rebate functions, linear and nonlinear for homogeneous objects as well as heterogeneous objects. The choice of neural networks is largely motivated by the universal approximation theorem, which states that a feed-forward network with a single hidden layer containing a finite number of neurons can approximate continuous functions on compact subsets.

3.3 Proposed Model

Neural Networks are biologically inspired paradigms which learn optimal functions from data. The main components that customize such a network for a specific task are its architecture and the objective function which guides its training. To design a rebate function which is OE or OW, we define an appropriate neural network in Section 3.3.1. We begin by describing an artificial neuron which is the fundamental processing unit of a neural network.

Basic Structure

An artificial neuron receives inputs $x_0 \; to \; x_m$. If necessary, we apply the nonlinear activation function ϕ to obtain the output (y) from the neuron. The activation function is for thresholding the output to introduce nonlinearity in the network. Thus the output of a neuron is, $y = \phi(\sum_{i=0}^{m} w_i x_i)$, where w_i is the weight for i^{th} input.

In any general network, we connect the neurons together in a specific and useful manner. The neurons are grouped into primarily three layers, the input layer, hidden layer, and output layer. The weights are randomly initialized before training. Given a set of training input and output pairs, the model compares its own output with the desired output and tries to learn the optimal set of weights by back-propagating the error through the network. In our case, we do not have a desired output, but we have an objective function that is to be maximized. That is, we need to determine optimal weights such that the rebate function is OE or OW. In addition, our mechanism should be Feasible and Individual Rational. The total VCG payment by the agents is t, and the neural network parameters are (w, b) then,

- Feasibility : $g(w,b) : t - \displaystyle\sum_{i=1}^{n} r_i(w,b) \geq 0$
- Individual Rationality : $g'(w,b) : r_i(w,b) \geq 0, \; \forall i \in N$

The above inequality constraints are added to the loss function during training. Having defined a general network and the constraints, we define the specific design of the network that we use.

3.3.1 Neural Network Architecture

Linear rebate function

To model the linear rebate function as given by Definition 3.4, we use a network consisting of neurons with n input and n output nodes without any activation function. The input nodes represent agents' valuations and the output nodes represent their re-

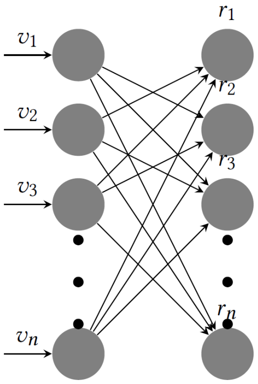


Figure 3.1: Linear Network

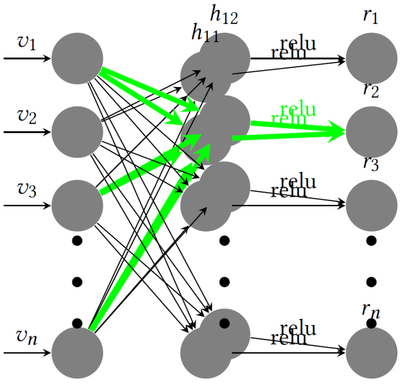


Figure 3.2: Nonlinear Network

bate. As required by Theorem 3.1, a rebate function for an agent i should depend only on the valuations of the remaining agents. Hence, we connect the i^{th} output node to all the input nodes except i^{th} input node as shown in Figure 3.1. We used a total of $n-1$ weights and 1 bias. Since the weights and the bias used are the same for calculating the rebate of each agent (represented by each node in the output layer) we ensure that the $r()$ is anonymous. In addition to the weights (w) which model the c_1 *to* c_n in $r()$, there is a same bias added (b), which models the c_0, to each output.

$$r_i = \sum_{j=1}^{n-1} v_i w_j + b, \forall i = 1 \rightarrow n$$

Nonlinear rebate function

The network consists of neurons with n input and n output nodes and one hidden layer. The input nodes represent agent valuations and output nodes represent the rebate. The neurons in the input and hidden layer use ReLU activation which returns 0 if the output of that neuron is negative else returns the output itself. Gujar et al. 2011 [95] defines the optimal nonlinear rebate function as the combination of marginal payments. We believe that the rebate functions though nonlinear, should only contain first degree terms for bid values as payments do not have higher order terms, making the function piece-wise linear. Hence, we use ReLU as our activation function.

As required by Theorem 3.1, a rebate function for an agent i should depend only on the valuations of the remaining agents. Hence, we connect the i^{th} output node to the i^{th} layer of hidden nodes which are connected to all the input nodes except the i^{th} input node as shown in Figure 3.2. The redistribution function being anonymous, the weights of the connections entering each hidden layer and each output are the same. The green thick lines in Figure 3.2 represent a unique set of weights which are connected to the second agents' output. The same weights are used for calculating the rebate of the other agents as well. The first set of weights (w) connect the input nodes to the hidden nodes and bias (b) is added to the hidden nodes. The second set of weights (w') connects the hidden nodes to the output nodes and the same bias is added (b') to each output node.

$$r_i = \sum_{k=1}^{h} relu(\sum_{j=1}^{n-1} v_i w_{jk} + b)w'_k + b', \forall i = 1 \; to \; n,$$

h: Number of hidden neurons, $relu(x) = max(0, x)$

The defined network architectures can model different functions depending on the weights. Hence, the training of the network guides the network to learn appropriate weights. Prior to training, we must recall Theorem 3.1 which necessitates the ordering of the input valuations. We further require the evaluation of t the VCG Payment. In the following section, we mention the details of the same.

3.3.2 Ordering of Inputs and Payments

The ordering and calculation of VCG payment in both homogeneous and heterogeneous cases are independent of the neural network.

Homogeneous Objects. All the given p objects are similar and each agent desires at most one object. The bids submitted are θ where, $\theta \in \mathbb{R}^n$. We order the bids such that

$v_1 \geq v_2 \geq \ldots \geq v_n$. Payment by agent i,

$$t_i = \begin{cases} v_{p+1} & i \leq p \\ 0 & i > p \end{cases}$$

Hence, $t = pv_{p+1}$.

Heterogeneous Objects. All the p objects are different, each agent will submit his valuation for each of the objects. The bids submitted are θ where $\theta \in \mathbb{R}^{p \times n}$. We define a particular ordering among these vectors based on the overall utility of each agent and the marginal valuations they have for each item. The allocation of the goods is similar to a weighted graph matching problem and is solved using the Hungarian Algorithm. Once we get the allocation say, k^*, we proceed to calculate the payments using the VCG payment $t = \sum_{i \in N} t_i$, where each t_i is given by Equation 3.1. The ordering of bids for the winning p agents is determined based on their utilities. The utility u_i of agent i is given by,

$$u_i = \sum_{j \in N} v_j(k_*(\theta), \theta_j) - \sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j) \ , \ \forall i = 1, \ldots, n.$$

If two agents have the same utility, their ordering is determined by their marginal values for the first item, and if it is same, then by the second item and so on. Once their ordering is determined, we remove the p agents and then run the VCG mechanism to get the next p winning agents and calculate the ordering using the same procedure as above. If the remaining agents are less than p we can still find the allocation and hence order the remaining agents till none are left or one is left. The time complexity of this ordering is polynomial in np. With the given ordering of the inputs and payments, we use the specified network models in both homogeneous and heterogeneous settings. For each setting, we model either OE or OW mechanism. For both OE and OW the network architecture remains same whereas the objective changes as defined in the following section.

3.3.3 Objective Function

During the forward pass the input valuations are multiplied by the network weights and the corresponding rebate for each agent is calculated. The initial weights being random, the rebate calculated will not be optimal. In order to adjust the weights to obtain the optimal rebate function, we add an objective at the end of the network which maximizes the rebate in both OW and OE. The loss function essentially is the negative total rebate of all the agents. The objective also takes care of the Feasibility and Individual Rationality condition.

Optimal in Expectation (OE)

- Given that we need to maximize the total expected rebate, the loss is defined as:

$$l(w, b) : \frac{1}{T} \sum_{j=1}^{T} \sum_{i=1}^{n} -r_i^j,$$

  T =total number of training samples

- Given Inequality constraint for Feasibility we modify it to equality as:

$$G^j(w, b) = max(-g(w, b), 0), \ \forall j = 1, 2..., T$$

- The overall loss function:

$$L(w, b) = l(w, b) + \frac{\rho}{2} \sum_{j=1}^{T} G^j(w, b)^2 \tag{3.2}$$

Worst case Optimal (OW)

- Given that in OW we are trying to maximize the worst possible redistribution index, as in Definition 3.5, the loss is given by:

$$l(k) : -k$$

$$such\ that\ g_3 : \sum_{i=1}^{n} r_i - kt \geq 0$$

T = total number of training samples

- Given inequality constraint for Feasibility we modify it to equality as $\forall j = 1, 2..., T$:

$$G_1^j(w,b) = max(-g(w,b), 0)$$

Given inequality constraint for IR we modify it to equality as:

$$G_2^j(w,b) = max(-g'(w,b), 0)$$

The inequality condition for finding the worst case optimal is modified as follows

$$G_3^j(w,b) = max(-g_3(w,b), 0)$$

- The overall loss function for the worst case:

$$L(w,b,k) = l(k) + \frac{\rho}{2}\sum_{j=1}^{T}[G_1^j(w,b)^2 + G_2^j(w,b)^2 + G_3^j(w,b)^2] \tag{3.3}$$

The network and objective together can be used to model any mechanism which is either OW or OE. We conduct a few experiments in order to learn an optimal mechanism which was analytically solved in theory for homogeneous settings. Further, our experiments for heterogeneous settings with objective OW, try to model HETERO [95] and also OE nonlinear mechanisms for homogeneous settings.

3.4 Implementation Details and Experimental Analysis

The proper training of neural networks is very crucial for its convergence. Xavier initialization and Adam optimization guide us in choosing the appropriate initialization and optimizer which are crucial for stabilizing the network. Given that we have two different networks and two different objectives, we experimented on various combinations of these to validate the data-driven approach with existing results. In the following subsections, we specify the implementation details.

Initialization and Optimizer

Xavier initialization [86]. The right way of initialization of a neural network is to have weights that are able to produce outputs that follow a similar distribution across all neurons. This will greatly help convergence during training and we will be able to train faster and effectively. Xavier initialization tries to scale the random normal initialized weights with a factor α, such that there is unit variance in the output. $\alpha = \frac{1}{\sqrt{n}}$, where n is the number of input connections entering that particular node. $\alpha = \sqrt{\frac{2}{n}}$ for ReLU.
Adam optimizer [127]. Adam is a first-order gradient-based optimization of stochastic objective functions. The method computes individual adaptive learning rates for different parameters from estimates of the first and second moments of the gradients. The default values provided in the TensorFlow library are used for $beta1 = 0.9$, $beta2 = 0.999\ and\ epsilon = 1e - 08.$. The learning rates are different for different cases as mentioned in their respective sections.

3.4.1 Different Settings for Training

Optimal in Expectation for Homogeneous Objects (OE-HO)

The inputs form a matrix of $(S \times n)$, where S is the batch size, the values are sampled from a uniform random distribution $U[0, 1]$. The batch size is set to be as large as possible, for $n < 10$, $S = 10000$ and $n = 10$, $S = 50000$. After that we apply the ordering and calculate payments for the given input valuations as defined in Section 3.3.2. Next, we feed it to the linear network model (Figure 3.1) whose parameters are initialized using Xavier initialization. The objective function (Equation 3.2) is applied to the output of the

n, p	Homogeneous			Heterogeneous	
	OE-HO Theoretical	OE-HO Linear NN	OE-HO Nonlinear NN	OE-HE Linear NN	OE-HE Nonlinear NN
3,1	0.667	0.668	0.835	0.667	0.835
4,1	0.833	0.836	0.916	0.834	0.920
5,1	0.899	0.901	0.961	0.900	0.969
6,1	0.933	0.933	0.973	0.934	0.970
3,2	0.667	0.665	0.839	0.458	0.774
4,2	0.625	0.626	0.862	0.637	0.855
5,2	0.800	0.802	0.897	0.727	0.930
6,2	0.875	0.875	0.935	0.756	0.954
10,1	0.995	0.996	0.995	0.995	0.995
10,3	0.943	0.945	0.976	0.779	0.923
10,5	0.880	0.880	0.947	0.791	0.897
10,7	0.943	0.944	0.976	0.781	0.857
10,9	0.995	0.997	0.996	0.681	0.720

Table 3.2: e^{oe} for homogeneous and heterogeneous setting.

network and parameters are updated using Adam optimizer, learning rate set to 0.0001. The nonlinear model (Figure 3.2) is also trained in a similar manner. We used 1000 nodes in the hidden layer and the network was trained with a learning rate of $10e-4$.

Optimal in Worst-case for Homogeneous Objects (OW-HO)

As in the OE case, a linear network model is used and a similar procedure is followed. The input is sorted as given in Section 3.3.2 and along with the calculated payments is fed to the linear network. The objective is optimized with the learning rate set to 0.0001 and the training is carried out till the loss decreases and saturates which happens when the redistribution index is optimal. As discussed in Section 3.2.2 for homogeneous setting linear rebate functions are optimal among all possible deterministic functions which are DSIC and AE, hence we did not use a nonlinear model for this case.

Optimal in Expectation for Heterogeneous Objects (OE-HE)

The inputs are again randomly sampled from a uniform distribution $U[0,1]$, the input matrix is of the form, $(S \times n \times p)$. Then the inputs are ordered as defined in Section 3.3.2. Both the networks (Figures 3.1, 3.2) are used for finding the optimal in expectation mechanism. Just like in the homogeneous case, network parameters are initialized using Xavier initialization. For the payment calculation, we use the scipy library for linear sum assignment. This library assigns objects such that the cost is minimized, whereas we want the valuation to be maximized as per AE, hence we negate the bids before passing it to the function. Besides, the Hungarian algorithm for assignment works only when the number of objects to be assigned is the same as the agents, hence we introduce dummy agents or dummy objects with zero valuation so that the input matrix is a square matrix. The objective function is optimized using Adam optimizer with a learning rate $10e-4$ for both the linear and nonlinear models. In the nonlinear network, the number of nodes in the hidden layer was set to 1000.

Optimal in Worst-case for Heterogeneous Objects (OW-HE)

For designing this particular mechanism, we use the same inputs that we used in the OE setting for heterogeneous items. The networks used and their initialization is also same. The only difference is the objective function which is given by Equation 3.3 and the optimizer used is Adam. The linear network learning rate is $10e-4$. In the nonlinear network, the number of hidden nodes used was 100 and a learning rate of $10e-4$ for all

n, p	Homogeneous		Heterogeneous	
	OW-HO theoretical	OW-HO Linear NN	OW-HE [1] Linear NN	OW-HE Nonlinear NN
3,1	0.333	0.336	0.332	0.333
4,1	0.571	0.575	0.571	0.571
4,2	0.250	0.250	0.0	0.250
5,1	0.733	0.739	0.733	0.732
5,2	0.454	0.460	0.0	0.454
5,3	0.200	0.200	0.0	0.199
6,1	0.839	0.847	0.839	0.838
6,2	0.615	0.620	0.0	0.614
6,3	0.375	0.378	0.0	0.375
7,1	0.905	0.910	0.905	0.904
7,2	0.737	0.746	0.0	0.736
7,3	0.524	0.538	0.0	0.523
8,1	0.945	0.949	0.945	0.943
8,2	0.825	0.834	0.0	0.825
9,1	0.969	0.972	0.968	0.968
9,2	0.887	0.894	0.0	0.886
10,1	0.982	0.985	0.982	0.982
10,2	0.928	0.936	0.0	0.927

Table 3.3: e^{ow} for Homogeneous and Heterogeneous setting.

values of n, with $p = 1$. When the values of $p > 1$, the number of hidden nodes was increased to 1000 and a learning rate of $10e-5$ was used.

[1] All values below 10e-3 are considered to be 0.0

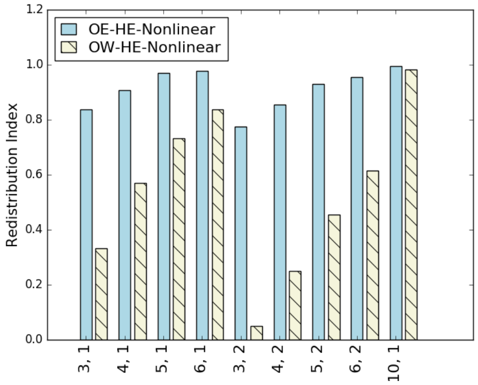


Figure 3.3: OE-HE-Nonlinear Vs OW-HE-Nonlinear

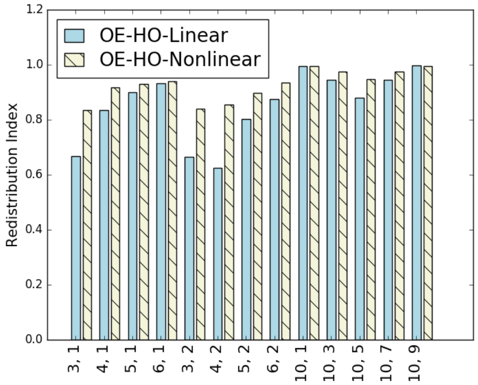


Figure 3.4: OE-HO-Linear vs OE-HO-Nonlinear

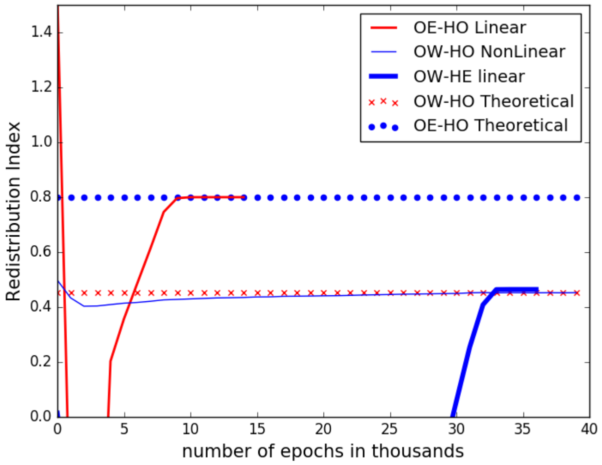


Figure 3.5: RI values with change in epoch for $n = 5$, $p = 2$

Specific parameters used for training

- We used TensorFlow library for all the implementations. We used GPUs: Tesla K40c and GeForce GTX Titan X. The network training time varies from a few minutes to a whole day, for the different n, p values.

- The number of input samples, T for binary settings ideally should be 2^{np} and way more than this for the real value cases. So we use 10000 samples where $np \leq 13$, 70000 for $np \leq 16$ and 100000 for the rest of the cases.
- All the experiments are run for a maximum of 400000 epochs and the constant ρ as given in the overall loss functions (Equations 3.2, 3.3) is set to 1000. The number of nodes in the hidden layer for the network given in Figure 3.2 is 1000 ideally.

3.4.2 Results and Discussion

We now describe the results obtained from the networks trained as discussed above. in Table 3.2, we compare the values of redistribution index (e^{oe}) for all the experiments with OE objective for different (n,p) values under homogeneous and heterogeneous settings. The first column indicates theoretical bounds on OE RMs with linear rebate functions. In the column heterogeneous, we compare the redistribution indexes obtained by our OE-HE networks with linear and nonlinear networks. Similarly, in Table 3.3, we compare the OW redistribution index (e^{ow}) for all the networks under consideration and theoretical values for different (n,p) values. With these tables, the following are our observations:

Achieving the analytically solved bounds. For the OE objective in homogeneous setting, our network OE-HO-Linear achieves the theoretical values of e^{oe} proposed in [101]. Similarly, the theoretical e^{ow} values are achieved by the network OW-HO Linear.

OW nonlinear rebate function for heterogeneous setting. The values from OE-HE-Linear NN illustrate the impossibility theorem stated in [95] that there cannot be a linear rebate function with non-zero redistribution index for heterogeneous settings. For $p = 1$, there being no difference in homogeneous or heterogeneous, the values remain the same but are zero for $p > 1$. The network OW-HE-Nonlinear achieves the theoretical values given in OW-HO theoretical (Table 3.3).

OE nonlinear rebate function for homogeneous setting. Guo et al. 2008 [101] have only tried to find the OE linear rebate function. OE-HO Nonlinear NN outperforms the linear counterpart. Figure 3.3 illustrates the comparison. This indicates the existence of a nonlinear rebate function which guarantees higher e^{oe} than the linear rebate function for a homogeneous setting.

OE nonlinear rebate function for heterogeneous setting. The values from OE-HE-Linear NN is shows the same results as OE-HO Linear for $p = 1$ and different otherwise. OE-HE-Nonlinear NN outperforms the linear network. Figure 3.4 compares between the OW and OE performance of the nonlinear network for heterogeneous settings.

Figure 3.3 illustrates the significantly better performance of Optimal in Expectation RM wrt Optimal in Worst Case for heterogeneous settings for different (n, p) values. For reference, it also has OE performance for homogeneous settings. We also observe that the performance of OE mechanisms with nonlinear rebates is significantly better as compared to OE mechanism with linear rebates (Figure 3.4). The convergence of neural networks for $n = 5, p = 2$ with the number of epochs while training can be found in Figure 3.5. Typically, most of the networks studied here converge in less than 80000 epochs for $n \leq 10$. (Whenever the objective value of a neural network drops below zero, we skip them in the plot).

3.5 Conclusion

To summarize, we show that neural networks can learn optimal redistribution mechanisms with proper initialization and a suitably defined ordering over valuation profiles. Our analysis shows that one can design nonlinear rebate functions for homogeneous settings that perform better than optimal in expectation linear rebate functions. We could design optimal in expectation rebate functions for heterogeneous objects which is not solved analytically. There are many challenges to be handled here. For example, can we design a vanilla neural network that can learn linear, nonlinear rebate functions without explicitly designing such architectures, based on the problem specific needs? Can we come up with training strategies independent of (n, p)?

Chapter 4

Expertsourcing

In many practical applications such as expertsourcing and online advertisement, the use of mechanism design (auction based mechanisms) depends upon inherent stochastic parameters. These parameters typically being unknown, learning is inevitable. The inherent stochasticity in such settings has been addressed using multi-armed bandit (MAB) algorithms. The mechanisms which incorporate MAB within them are referred to as Multi-Armed-Bandit Mechanisms. While most of the MAB mechanisms focus on frequentist approaches like upper confidence bound algorithms, recent work has shown that using Bayesian approaches like Thompson sampling results in mechanisms with better regret bounds; although lower regret is obtained at the cost of the mechanism ending up with a weaker game theoretic property i.e. Within-Period Dominant Strategy Incentive Compatibility (WP-DSIC). The existing payment rules used in the Thompson sampling based mechanisms may cause negative utility to the auctioneer. In addition, if we wish to minimize the cost (or maximize revenue) to the auctioneer, it is very challenging to design payment rules that satisfy WP-DSIC while learning through Thompson sampling.

In our work, we propose to use a data-driven approach for designing MAB-mechanisms. Specifically, we use neural networks for designing the payment rule which is WP-DSIC, while the allocation rule is modelled using Thompson sampling. Our results, in the setting of crowd-sourcing for recruiting quality workers, indicate that the learned payment rule guarantees better revenue while maximizing social welfare and also reduces variance in the utilities to the agents.

4.1 Introduction

In the real world, we often encounter situations where we have to choose among competing and strategic agents to achieve a specific goal. These agents hold private information which is crucial to the decision. Misreporting of private information may lead to a sub-optimal outcome. Hence, we need to design a mechanism that ensures truthful reporting of private information, [155, Chapter 9].

Designing an appropriate mechanism to ensure productive result involves designing an *allocation rule* and a *payment rule*. Auction mechanisms dealt with are usually deterministic, in the sense that once all the bids are known, all the agents and the auctioneer are sure about the course of the auction. There are many settings like crowd-sourcing, online advertisement etc, where auction design relies on environmental parameters which neither the agent nor the hiring agency is sure about. For example, in crowd-sourcing, the actual quality of the agent is known to neither the agent nor the auctioneer or the probability of clicks that an advertisement will receive in online advertising are external parameters. Such parameters are not deterministic but are subject to various environmental conditions, hence are stochastic or could even be adversarial. In this work, we restrict to stochastic settings. In such a setting, it becomes necessary to figure out the average values of these parameters (qualities in crowd-sourcing and click-through rates in online advertisement) through exploration and at the same time ensure that the agents do not misreport their cost. However, in the presence of such learning algorithms, the strategic agents have more freedom to manipulate. Hence it is required to design novel mechanisms that also learn the environmental parameters. Such mechanisms are referred to as *Multi-Armed-Bandit (MAB) Mechanisms* [19, 16, 33, 80, 82, 117, 116, 175].

In MAB, we consider each of the agents or the advertisements as an arm. The auctioneer repeatedly selects an arm in order to observe its performance and gets an estimate of the expected reward from that arm. The performance of a MAB algorithm is captured through the notion of *regret*, which is the difference between the expected reward from the optimal arm and the expected reward from the algorithm. There are two popular algorithms in MAB; one algorithm is based on the frequentist approach called as Upper Confidence Bound (UCB) algorithm [13]. The other technique follows the Bayesian approach and is called Thompson Sampling [187]. Thompson sampling has state-of-the-art performance in solving MAB problems. In practice, it is known to achieve lower regret than the other algorithms. When designing a MAB based mechanism, we impose restrictions on the allocation rule to ensure the truthfulness of the payment rule. This affects the regret of the algorithm. The payment rule could be 1) Deterministic, which leads to high regret in social welfare [19, 66], or 2) Randomized, which achieves low regret but a higher variance in agent utilities [18, 32]. Previously there has been work related to UCB based mechanism design [18, 19] and Thompson sampling based mechanism design [83].

In this work our goal is to design MAB based mechanisms which ensure truthful reporting of the strategic values and achieve *Allocative Efficiency (AE)*. We consider the problem of selecting high quality service providers (agents) such that the welfare obtained by the hiring agency (auctioneer) is maximized at a minimal cost. The welfare depends on the Quality of Service (QoS) the agent provides and is a stochastic quantity. This is a reverse auction setting where the auctioneer pays the selected agent for its service. The auctioneer would want to minimize the payments to the cost optimal agents at each round (AE is satisfied). Note that, this is different from Myerson's optimal auction design, which in our setting would be the same as minimizing the payments to the agents without guaranteeing AE. In order to evaluate the payments made by our mechanism, we introduce the notion of *Cost Index (CI)*. It is the expected value of the ratio of payments made by the mechanism to the optimal payments. In our setting, we desire CI should to be as low as possible ideally near one.

Ghalme et al. 2017 [83] propose two *Thompson sampling based MAB mechanisms*, *TSM-D* and *TSM-R* for solving the above problem of crowd-sourcing. The primary aim in their paper is to achieve low regret for the auctioneer while ensuring reduced variance in the utilities of the agents. The lower the regret achieved by the learning algorithm, the more likely it is for the mechanism to achieve AE. As discussed by the authors, ensuring ex-post dominant strategy incentive compatibility (DSIC) requires the agents to have full knowledge of future events, hence is difficult to achieve. Instead, their mechanism ensures a weaker notion of truthfulness, called *Within-Period DSIC* (WP-DSIC). The payment rules in TSM-D and TSM-R are designed just to ensure WP-DSIC, but the auctioneer's payments to the agents need not be the minimum possible. The mechanisms also ignore the possibility of the payment exceeding the welfare of the auctioneer. Our analysis shows TSM-D pays very high as compared to welfare and there is a non-zero probability of the payments being higher than welfare in TSM-R. Analytically coming up with payment rules in Thompson sampling based MAB settings is challenging. With these shortcomings of TSM-D and TSM-R in sight, we propose a data-driven mechanism which learns the optimal payment rule to minimize the payments while ensuring high social welfare. We also ensure this mechanism is WP-DSIC and *Ex-post Individual Rationality* (EPIR).

Recently, researchers are exploring mechanism design using neural networks, e.g., [69] as often, it is not known how to design mechanisms analytically. In this approach given the appropriate data, the network learns a payment rule such that certain game theoretic properties are satisfied and in some cases, there is a network which also learns the allocation rule. The network acts like a function approximator and learns the complex mapping needed to satisfy the constraints. We propose to have a *neural network and multi-armed bandit based mechanism design*. To the best of our knowledge, it is the first work to use neural network based approach to design payments when learning is happening through MAB techniques.

Our Contributions. i) Data-driven approach for learning the payment rule in a stochastic setting. ii) The payment rule is learned to minimize the total payment while maximizing welfare. iii) The payment rule enjoys the desirable properties of within-period DSIC and ex-post IR. iv) The payment is ensured not to exceed the welfare. v) The variance in the utility to the agents decreases with time.

Related Work

Leonid Hurwicz first introduced the notion of mechanisms with his work in 1960 [114]. Vickrey 1961 [190] introduced the celebrated Vickrey auction (second price auction). Hurwicz 1972 [113] introduced the key notion of incentive compatibility in 1972. This notion allowed mechanism design to incorporate the incentives of rational players. Clarke 1971 [54] and Groves 1973 [94] came up with a generalization of Vickrey mechanisms and helped define a broad class of DSIC mechanisms called Vickrey-Clarke-Groves or VCG mechanisms. The field of Algorithmic Mechanism Design is for designing mechanisms in computational settings. Designing mechanisms in deterministic setting has seen a lot of research, whereas stochastic settings started getting attention recently. The stochastic MAB problem is a classic problem described by Robbins 1952 [169]. In many settings like online advertisement and crowd-sourcing, the role of arms is played by the strategic agents who may hold some private information which is of interest to the learner, while the welfare obtained from these arms is stochastic. Since the agents maximize their own profit, they may misreport their valuations which call for MAB based mechanism design.

Multi-arm based Mechanism Design

Babaioff et al. 2009 [19] and Devanur et al. 2009 [66] characterized truthful mechanisms for MAB motivated by the pay-per-clicks auction for internet advertising. [175, 2] are the other works that explore this domain. Ganesh et al., Jain et al. 2016, 2014 [80, 116] pose the crowd-sourcing as MAB based mechanism design problem. All the existing mechanisms are upper confidence bound (UCB) based [18, 19, 33]. Recent work [83] has shown that the Thompson sampling algorithm has shown slightly better performance guarantees than others [6, 49, 124]. The authors in [83] have designed a mechanism using Thompson Sampling which motivates our work.

To the best of our knowledge, these approaches have not been used in designing MAB mechanisms. We make efforts towards using neural networks for designing a Thompson sampling based mechanism.

4.2 Preliminaries

We have an auctioneer in need of a certain service repeatedly. There is a pool $K = \{1, 2, \ldots, k\}$ of the service providing agents. Each agent i' QoS is stochastic and the average QoS is represented by $\mu_i \in [0, 1]$ (higher the better). The cost c_i is private information held by the strategic agent i. Note that, μ_i denotes the probability with which the auctioneer is satisfied with the service provided by the agent i. If the auctioneer is satisfied, he obtains a welfare of W and zero otherwise. We use Bernoulli rewards as it is very common in the literature and valid in most of real-world situations. For e.g., the service could be document classification, image identification etc, where μ_i indicates the accuracy with which agent i performs this task. c_i denotes the cost incurred by the agent i for providing the service for one round. Thus, the auctioneer obtains a welfare $r_i = W - c_i$ with probability μ_i and $-c_i$ with probability $1 - \mu_i$ if an agent i is selected. The auctioneer's goal is to select an agent that maximizes the expected welfare $w_i = W\mu_i - c_i$.

Note that, it is also possible to consider reward to the auctioneer in round t as $R_t = W\mu_{I_t} - p_{I_t,t}$ where I_t denotes the agent selected in round t. However, as our first goal is to maximize social welfare and as also it is common in literature [19, 83] to consider welfare as a reward, we use welfare as a reward. That is $R_t = W\mu_{I_t} - c_{I_t,t}$.

The expected welfare from agent i (w_i), which is also the reward to the auctioneer as has two components, 1) $W\mu_i$ which is unknown and stochastic, and 2) $-c_i$, which is private to the agent i and is strategic. Let the history of allocations and observations till round $t-1$ be denoted by h_t which is common knowledge. Let $b_{i,t}$ denote the bid or cost reported by the agent i. b_t is the bid vector for all the agents in round t. Let $b_{-i,t}$ be the bid vector in t, of all the agents other than i. Now, we define Δ_i as the difference between the expected reward of any sub-optimal agent and the optimal agent. Given that the parameters for optimal agent are c_{opt}, μ_{opt}, $\Delta_i = (W\mu_i - c_i) - (W\mu_{opt} - c_{opt})$ and $\Delta = max_i \Delta_i$. Typically the performance of such MAB algorithm is captured using *regret*: the difference between the performance of the algorithm and the performance of an optimal arm. We have not provided explicit regret analysis, as we follow Thompson sampling for allocation and the regret for the same will hold true irrespective of the payment rule. Hence we refer to [83, Theorem] for the regret analysis.

Let $I_t \in K$ be the service providing agent selected at round t. The auctioneer pays $p_{i,t}(b_t; h_t)$ to the agent i if the agent is selected in round t. The utility of an agent i in round t is given by $u_{i,t}(b_t; h_t; c_i) = \mathbb{1}\{I_t(b_t; h_t) = i\}(p_{i,t}(b_t; h_t) - c_i)$. When we use Thompson sampling to select the agent at each round, there is an inherent randomness caused by it and is denoted by w_t.

Given that the agents are strategic, we need to design an appropriate mechanism to elicit the true costs i.e. c_i. The mechanism denoted by $\mathcal{M} := (\mathcal{A}, \mathcal{P})$ has two components, first is the *Allocation Rule* ($\mathcal{A}$) which takes a bid vector b_t and history h_t as inputs and outputs the index I_t of the selected agent. The second component is the *Payment Rule*

($\mathcal{P}$) which determines the payment at each round. A few properties which we want our mechanism to have are as follows,

4.2.1 Desirable properties

Definition 4.1 (*Allocative Efficiency* (AE))**.** *We say a mechanism $\mathcal{M}$ is allocatively efficient if every round t it selects agent I_t such that,*

$$I_t(b_t) \in argmax \; W\mu_i - b_{i,t}$$

If μ_i is known, we can focus on a single round auction. (We drop t from relevant terms for time being). We prefer the mechanism to be DSIC, that is, reporting the truth is a dominant strategy for all the agents. We can use Groves' payment to achieve this if the allocation rule is AE.

Definition 4.2 (*Groves Payment*)**.** *An AE mechanism is DSIC if it satisfies the following payment structure*

$$p_i(b_i, b_{-i}) = \sum_{j \neq i} R_j(I(b)) + g_i(b_{-i}) \;\; \forall i = 1, \ldots k$$

where g_i is any arbitrary function mapping to $\mathbb{R}$.

The above theorem provides the sufficient condition under which a mechanism that is AE is also DSIC. The First Characterization Theorem of Green-Laffont proves Groves' theorem as also necessary. Any mechanism $\mathcal{M}$ that satisfies properties in Definitions 4.1 and 4.2 is called *Groves mechanism*. A special case of Groves mechanism is VCG mechanism where $g_i(b_{-i}) = -\sum_{j \neq i} R_j(I(b_{-i}))$. Note that, VCG mechanisms may end up paying very high amounts to the agents. Thus, we explore designing Groves' mechanism. However, designing such payment rules analytically in the presence of MAB algorithms is challenging. Hence, we propose to use a neural network which learns the function g_i.

In the case of repeated auctions, one can generalize DSIC to ex-post DSIC, that is, even when the agents have access to an oracle predicting all the future random events, reporting truth is still a dominant strategy. Such assumption is impractical and typically agents will be myopic[1] as pointed out in [83], we use the following practical notion of Incentive Compatibility; namely Within Period DSIC (WP-DSIC). In WP-DSIC, it is the dominant strategy to report truthfully in that particular round if agents do not consider future gains; albeit agents can manipulate if they consider future rounds.

Definition 4.3 (*Within Period Dominant Strategy Incentive Compatible* (WP-DSIC))**.** *We say a mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{P})$ is WP-DSIC if for all agents and for all rounds, the utility of an agent from truthful bidding is at least as much as the utility from any non-truthful bidding irrespective of the bids of other agents, i.e.* $\forall i, \forall c_i, \forall t, \forall w_t, \forall h_t$ *and* $\forall b_{-i,t}$,

$$u_{i,t}(c_i, b_{-i,t}; h_t; c_i|w_t) \geq u_{i,t}(b_{i,t}, b_{-i,t}; h_t; c_i|w_t) \quad \forall b_{i,t}$$

One can achieve WP-DSIC by using the Groves mechanism in each round. Another important desirable property is individual rationality. That is, no agent should incur losses by participating in the mechanism.

Definition 4.4 (*Ex-Post Individual Rationality* (EPIR))**.** *We say a mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{P})$ is EPIR if every agent has a non-negative utility with truthful bidding irrespective of the bids of other agents i.e.,* $\forall i, \forall c_i, \forall t, \forall h_t, \forall w_t$,

$$u_{i,t}(c_i, b_{-i,t}; h_t; c_i|w_t) \geq 0 \ \forall b_{-i,t}$$

Apart from the properties mentioned above, considering the setting of reverse auction we also want the mechanism to incur minimum cost to the auctioneer. In mechanism design the efficiency of a mechanism in terms of cost effectiveness is captured through *frugality*. Frugality captures how much the mechanism overpays as compared to the optimal

[1] i.e the agent always maximizes the expected reward w.r.t the current round and does not take into account any future rounds

mechanism. However, in the presence of Thompson sampling based learning, it is not clear how to define the cheapest Nash cost. Hence to capture how much a mechanism is overpaying for truthful reporting, we introduce the notion of *cost index* (CI). We use CI to evaluate different mechanisms for their cost efficiency.

Definition 4.5 (*Cost Index* (CI))**.** *Cost Index of a mechanism $\mathcal{M}$ at round t is the expected ratio of payments by the mechanism to the actual cost incurred by the selected agent,*

$$CI_t^{\mathcal{M}} = \mathbb{E}\left[\frac{p_{I_t,t}}{c_{I_t,t}}\right]$$

The CI as defined above always has to be higher than one.

4.2.2 Existing Approaches

The authors in [83] propose a Thompson sampling based allocation along with two different payment rules, which form the basis for our data-driven approach. We describe these in the following subsections.

Allocation Rule

Thompson sampling algorithm maintains a prior over expected welfare from each agent based on the observed history. At each round, the algorithm selects an agent based on the samples of the expected welfare from the prior distribution. Then the priors are updated after observing the welfare corresponding to the selected agent at a particular round.

Let $X_{i,t}$ be the Bernoulli welfare or reward which takes the value 1 when the agent i provides satisfactory service else takes the value 0. The probability that $X_{i,t} = 1$ is μ_i. The actual reward at round t is $WX_{i,t} - c_i$ and the expected reward is $W\mu_i - c_i$. In Thompson sampling, we maintain Beta priors on the stochastic rewards of agent i with parameters $\alpha_{i,t}$ and $\beta_{i,t}$. Here, $\alpha_{i,t}$ denotes the number of times agent i has provided satisfactory service till round t and $\beta_{i,t}$ denotes the number times the agent fails. Let $\theta_{i,t}$ be the sample from this Beta distribution. The allocation rule in round t is give by the following,

$$I_t = max_i \; \{W\theta_{i,t} - b_{i,t}\} \tag{4.1}$$

Maintaining beta priors for the Bernoulli rewards turns out to be a convenient choice as the posterior distribution is again a Beta distribution with a simple update in its parameters as described below. Given that the beta priors for round t are $(\alpha_{i,t}, \beta_{i,t})$. If $X_{I_t,t} = 1$ then $\alpha_{I_t,t+1} = \alpha_{I_t,t} + 1$ and $\beta_{I_t,t+1} = \beta_{I_t,t}$ else, $\beta_{I_t,t+1} = \beta_{I_t,t} + 1$ and $\alpha_{I_t,t+1} = \alpha_{I_t,t}$. In the first round, the Thompson sampling algorithm assumes to have prior Beta$(1, 1)$ on μ_i which is the uniform distribution on $(0, 1)$. This allocation mechanism strives to maximize social welfare given that the bids from the service providers are a true reflection of their actual costs. In order for this to be true, the authors have proposed the following two payment rules.

TSM-D

It is an estimate based payment rule. Given that $j_t^* = argmax_{i \neq I_t}\{W\theta_{i,t} - b_{i,t}\}$ be the second best agent at round t based on sample values. Let $\hat{\mu}_{i,t} = \frac{\alpha_{i,t}}{\alpha_{i,t}+\beta_{i,t}}$ and $N_{i,t} = \alpha_{i,t} + \beta_{i,t}$. We define $e_{i,t}(\gamma) = \sqrt{\frac{4\gamma \ln(t)}{N_{i,t}}}$ to be the exploration term for agent i at round t with parameter $\gamma \geq 1$. Then the payment at round t is given by,

$$p_{I_t,t} = W\hat{\mu}_{I_t,t} - W\hat{\mu}_{j_t^*,t} + b_{j_t^*,t} + 2W(e_{I_t,t}(\gamma) + e_{j_t^*,t}(\gamma)) \tag{4.2}$$

TSM-D is deterministic given the history of past allocations. Being deterministic it achieves low variance in utilities when the events are fixed. As mentioned in [83], the exploration terms help ensure the game theoretic properties. Moreover, the game theoretic properties satisfied by this rule are very weak. TSM-D is EPIR with a high probability and WP-DSIC with a high probability.

As the payment at every round depends on the entire history and given that the game theoretic properties satisfied are not so attractive, we explore the following formulation of payment which overcomes these issues.

TSM-R

The payment at round t is randomized and is given by the following equation,

$$p_{I_t,t} = W\theta_{I_t,t} - W\theta_{j_t^*,t} + b_{j_t^*,t} \tag{4.3}$$

TSM-R satisfies both EPIR and WP-DSIC, hence is more desirable than TSM-D. Given the payment at each round depends on the random values θ_t, the variance in the utility values may become worse. We get the following Lemma from [83] to bound the variance,

Lemma 4.1. *Variance in utility of optimal agent i^* satisfies,*

$$\lim_{t\to\infty} var(u_{i^*,t}(.)) \leq \lim_{t\to\infty} \frac{W^2}{2} max\left\{\frac{1}{N_{i^*,t}+3}, \frac{1}{N_{j_t^*,t}+3}\right\}$$

For any other agent $i \neq i^*$, variance in utility asymptotically goes to 0.

This payment rule is not dependent on the entire history and also satisfies the desirable game theoretic properties, hence our neural network based payment rule is loosely based on this. In the next section, we specify the details of the neural network architecture used for designing the mechanism.

4.3 Proposed Model: TSM-NN

Our model is a Thompson sampling based Neural Network (TSM-NN). At every round, our model uses Thompson sampling for allocation. Once the allocation is done, we get the rewards corresponding to the allocated agent. Then we invoke the neural network model which is designed to implement the Groves payment rule.

Allocation rule

For implementing the allocation rule, we directly conduct Thompson sampling as described in Algorithm 7.

Algorithm 7 Mechanism TSM-NN

```
Input: Number of rounds T, Number of agents k, bids $\{b_{i,t}\}_{i=1}^{k}$ in each round $t \in \{1, 2, \ldots, T\}$
Output: Allocations $\mathcal{A} = \{I_i\}_{t=1}^{T}$ and payments $\mathcal{P} = \{p_{I_t,t}\}_{t=1}^{T}$
Initialize: $\alpha_{i,1} = 1, \beta_{i,1} = 1 \;\; \forall i \in \{1, 2, \ldots, k\}$
for t = $1, 2, \ldots$,T do
    Sample: $\theta_{i,t} \sim Beta(\alpha_{i,t}, \beta_{i,t}) \; \forall i \in K$
    Allocate:
        $I_t = \underset{i}{argmax}\{W\theta_{i,t} - b_{i,t}\}$
    Payment: $NN_{input} : [b_{-i,t}, \theta_t]$,
        $NN_{payment} = \text{TSM-NN}(NN_{input})$
    Observe: The Bernoulli reward of an agent $I_t$ for round $t$
        i.e. $X_{I_t,t} = \begin{cases} 1, & \text{w.p } \; \mu_{I_t} \\ 0, & \text{w.p } \; 1 - \mu_{I_t} \end{cases}$
    Update:
        $\alpha_{I_t,t+1} = \alpha_{I_t,t} + \mathbb{1}\{X_{I_t,t} = 1\}$
        $\beta_{I_t,t+1} = \beta_{I_t,t} + \mathbb{1}\{X_{I_t,t} = 0\}$
        $\alpha_{i,t+1} = \alpha_{i,t}, \; \beta_{i,t+1} = \beta_{i,t} \; \forall i \neq I_t$
end for
```

Ideally for a mechanism to be completely data-driven, the allocation rule must also be implemented using a neural network. That is, we must implement Thompson sampling using neural network. On careful consideration, we figured that implementing Thompson sampling requires the network to learn a distribution at every round. There have been quite a few generative networks that learn data distribution like GANs [91] and VAE's [126]. In our case, we have new parameters of the Beta distribution i.e. $\alpha \text{and} \beta$ at every round as the priors are updated. Hence we need to train a different network for every possible combination of α, β that can arise. This of course is one possible approach and it indeed will significantly increase the training time. Another major challenge arises from the fact that we deal with Bernoulli rewards which are discrete. Discrete distributions are challenging for neural networks to handle considering that they use back-propagation for optimization. With a view to all these issues, we have left the implementation of the allocation rule using neural networks to future work.

Payment rule

After the allocation is done according to the above rule, we then get the rewards corresponding to the selected agent. To ensure Groves' payment rule, the payment by an agent is dependent on bids by all the other agents except itself. Hence the input to our neural network is a list of bids by other agents and $W\theta_t$, where θ_t are samples from the beta distribution maintained by the Thompson sampling at round t (motivated by the payment rule TSM-R). This input is passed to one hidden-layer neural network with ReLU as activation. The final output layer of the neural network has one node which gives the payment corresponding to the agent selected $I_t = i$ in round t. The payments corresponding to the other agents are 0, hence the total payment by the auctioneer at every round is given by the output of the network. Considering, the network follows Groves' payment rule, it has to be WP-DSIC. The other properties are ensured with the help of proper loss functions as described further in the following section. To summarize and compare with the existing approaches we provide the different properties satisfied by them in Table 4.1

	TSM-D	TSM-R	TSM-NN
Payment	Deterministic	Randomized	Randomized
WP-DSIC	No	Yes	Yes
EPIR	No	Yes	Yes
Variance in utility of the optimal agent[2]	(highest)	(medium)	(lowest)
Cost Index	(highest)	(medium)	(lowest)

Table 4.1: Summarizing properties satisfied by the three mechanisms

4.4 Implementation Details and Experimental Analysis

At each round t, we receive the bids from the agents and sample the rewards from the Beta distribution to form our input. This input is then passed on to the network described in Section 4.3. The network gives us the payment corresponding to the agent selected in round t. Now, we want our mechanism to have a few properties and we mention the corresponding component required to enforce it in the network's loss function. It is to be noted, as discussed in Section 4.3 WP-DSIC is already established in the network as the input for the agent i does not include its own bid.

4.4.1 Components of the Loss Function

Before moving on to the overall loss function, we break it up into components, where each component is responsible for enforcing certain properties that we desire.

Minimum Payment. The Allocation rule defined in Section 4.3 ensures that the welfare is maximized. Given the setting of a reverse auction, the auctioneer also wants to minimize the amount of money it has to pay while recruiting the agents. Hence we want the network to output payments such that they are minimized, which is enforced by l_{mp} for

$t = \{1, 2, \ldots T\}$ and where I_t is given by Equation 4.1

$$l_{mp} = -p_{I_t,t} \tag{4.4}$$

Cost Index is a parameter we use, to estimate the excess payment made by the mechanism.

EPIR. As given in Definition 4.4, in this component we try to minimize payments only to the extent that the agent is able to obtain a positive utility. Hence our l_{epir} is given by,

$$l_{epir} = max\{-(p_{I_t,t} - b_{I_t,t}), 0\} \tag{4.5}$$

In this equation, we replace the $c_{I_t,t}$ with $b_{I_t,t}$ from the utility definition as we assume the bids are truthful given that the mechanism is WP-DSIC.

Rationality of the auctioneer. Both the payment rules proposed previously TSM-R and TSM-D, do not consider the possibility where the payment at any round t, $p_{I_t,t}$ might exceed the welfare to the auctioneer W. To incorporate this in our loss we define $l_{rc} = max\{-(W - p_{I_t,t}), 0\}$.

Given the main components, we describe the overall loss function to the network as follows,

$$L = l_{mp} + \rho_1 \; l_{epir}^2 + \rho_2 \; l_{rc}^2 \tag{4.6}$$

In the above, equation ρ_1 and ρ_2 are tunable hyper-parameters which are set to appropriate values such that payments learned by the network are optimal.

Optimizer and training details

Optimizing the loss defined in Equation 4.6, we use *Adam Optimizer* and *Xavier Initialization* for initializing the parameters in the neural network. While training, we sample 6000 bids from the uniform distribution $U[0, 50]$ for two agents. For each set of bids, we train the network for 10^5 trials. The value of W is set to 50. Given the experiments are for 2 agents, it is sufficient for the network to have 10 hidden neurons with ReLU as activation, for reasonable results. Since updating the network at every trial increases the time for training, we collect the bids and rewards for a batch of trials which in our case is 10^3 and update the network weights for this batch of samples. The value of ρ_1 is set to 100 and ρ_2 is set to 1 for the claimed results.

4.4.2 Experiments and Results

In this section, we discuss the different experiments conducted for comparison with the existing approaches, TSM-R and TSM-D. The primary point of comparison is the payments made by the mechanism. Then we consider the variance in the utilities to the agents. It might put off agents if they face high uncertainties in payments for the exact same quality of service that they offer. Hence it is crucial to monitor, the variance in the utilities caused by the underlying mechanism. It is important to note that, the payment rule given in TSM-D achieves low variance only when the Bernoulli events or rewards are kept constant. This assumption is not practical, as the rewards may vary with different episodes (Algorithm 7, line 8).

Average Payments. In the experiment, we fix the bid values to $\{30.0, 35.0\}$ such that $\Delta = 3.5$. Fixing the bids, we run 10^5 trials, for 1000 iterations. Then we take the average of the payments predicted by the network across the 1000 iterations and compare it with the predictions by the existing approaches. From Figure 4.1 we can see that the average payments by the NN are consistently low, although higher than 30 to maintain IR leaving the first 10^4 trials. The average payments corresponding to NN is 30.6 which is just above the optimal 30.0 in this particular example. The plots show that the payments by TSM-D are way higher than W which violates the condition of rationality to the auctioneer.

Variance in utilities. Similar to the previous case, for calculating the variance, test run the network for 1000 iterations each having 10^5 trials. The bids are again fixed to have $Delta = 3.5$. We calculate variance in utility to the optimal agent across the 1000 iterations and plot it for all the three mechanisms Figure 4.2.

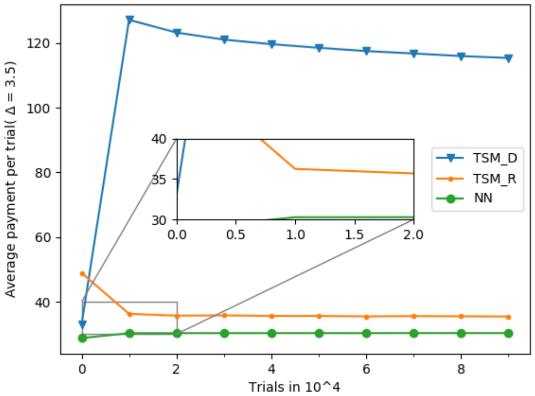


Figure 4.1: Average payments Vs Trials

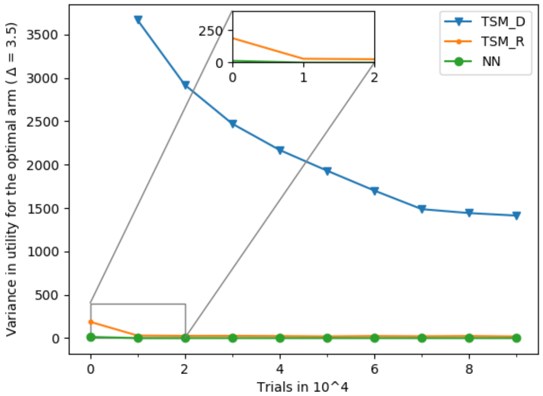


Figure 4.2: Variance in Utility Vs Trials

Effect of Δ. In this experiment, we chose different values for the bids, which in turn correspond to the different values of Δ. The main aim of this experiment is to check the effect of increasing Δ values on the variance of the utilities obtained by the agents. TSM-D, as claimed in, [83] ideally should not have much change in variance w.r.t Δ only if the events are fixed. In this case, we have conducted experiments without fixing the events which leads to high variance in the utilities from TSM-D primarily due to the exploration terms $e_{i,t}$ in its payment rule. We can analyze from Figure 4.3, that both TSM-R and NN have similar performance, although NN manages to maintain lower utilities as compared to TSM-R at higher values of Δ

Cost Index. In this experiment, we compare the different mechanisms on the cost index. It is to get an estimate on how much extra the auctioneer is paying than he ideally should in order to ensure truthfulness. From the plot in Figure 4.4, it is clearly indicative that NN has the least CI in all the rounds t. TSM-D has the highest CI whereas TSM-R has considerable value although higher than NN.

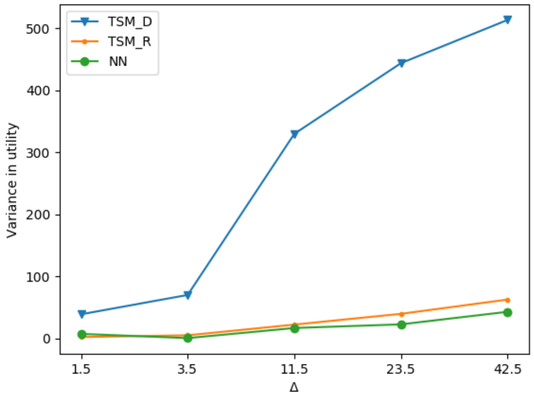


Figure 4.3: Variance in Utility Vs Delta values

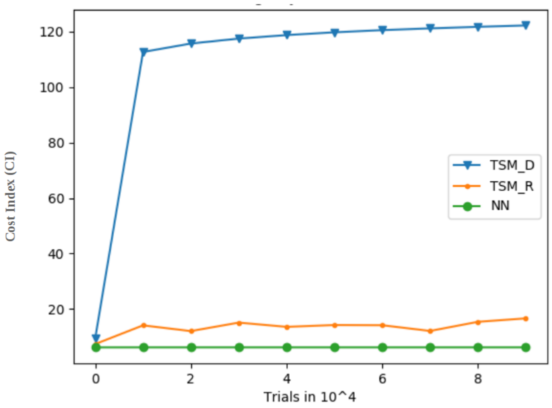


Figure 4.4: Cost Index Vs Trials

4.5 Conclusion

We explored the setting of designing mechanisms with inherent stochasticity. The underlying stochastic parameters are learned using MAB algorithms. We as our primary objective have automated the process of designing the payment rule for a Thompson sampling based mechanism. To the best of our knowledge, this is the first time, where data-driven approach has been used to model a MAB mechanism. We used a neural network model to design the payment rule. Our network as the payment rule, is able to ensure WP-DSIC and EPIR mechanism while making sure that the utilities of the agents do not vary significantly. Additionally, unlike the previous approaches, this approach also ensures that payments are minimized and also do not exceed the welfare value, which would violate the auctioneer's rationality. We have confirmed the above claims, by comparing the Cost Index between the approaches in our experiments.

Chapter 5

Fair Division

Fairness is well studied in the context of resource allocation. Researchers have proposed various fairness notions like envy-freeness (EF), and its relaxations, proportionality and max-min share (MMS). There is a vast literature on the existential and computational aspects of such notions. While computing fair allocations, any algorithm assumes agents' truthful reporting of their valuations towards the resources. Whereas in real-world web-based applications for fair division, the agents involved are strategic and may manipulate for individual utility gain. In this work, we study strategy-proof mechanisms without monetary transfer, which satisfies the various fairness criteria.

We know that for additive valuations, designing truthful mechanisms for EF, MMS and proportionality is impossible. Here we show that there cannot be a truthful mechanism for EFX and the existing algorithms for EF1 are manipulable. We then study the particular case of single-minded agents. For this case, we provide a Serial Dictatorship Mechanism that is strategy-proof and satisfies all the fairness criteria except EF.

5.1 Introduction

Fair division of resources is critical in various situations like division of inheritance and land, allocation of rooms among housemates, jobs to workers, and time slots to courses. In a typical scenario, the agents involved report their valuations for the resources available. The central aggregator or the underlying software aggregates these reported valuations to output a fair allocation. Various web-based applications like Spliddit [1], Fair Proposals System [2], Coursematch [3], Divide Your Rent Fairly [4], etc offer such solutions readily. Often the participants are strategic and misreport their valuations to improve their utility. The party that strictly adheres to the protocol and reveals its true valuation (while it can misreport and achieve more utility) may find it unfair if others misreport for their benefit even though the underlying algorithm is fair for the reported types. In auction settings, one prevents such strategic manipulations through monetary transfers. Whereas in resource allocation, no monetary transfers are allowed. Hence, it is essential to look for truthful mechanisms that ensure fairness without payments.

In this work, we focus on indivisible resources and the fairness notions of *envy freeness* (EF), *proportionality* and *maxi-min share* (MMS). Proportionality [180] is the first concept of fairness ever proposed. It ensures that each agent receives a fair share of its utility. Another popular notion is envy-freeness (EF). An allocation is EF when no pair of agents exist such that one of the agents increases its utility by exchanging their allocated goods [78]. For divisible goods, EF allocations always exist [181], and complete allocation may not exist for indivisible goods. It is also NP-hard to compute an approximation to EF [130]. When the valuations are sub-additive, EF implies proportionality [41]. Although proportionality is a weaker notion, its existence is still not guaranteed for indivisible goods.

[1] www.spliddit.org

[2] www.fairproposals.com

[3] www.coursematch.io

[4] https://www.nytimes.com/interactive/2014/science/rent-division-calculator.html

Property	Single-Minded	Identical	Additive	
		Additive	($n = 2$)	($n \geq m$)
EF			✗[130]	
Proportionality	✓(SD)[5]		✗[10] (alternative proof)	
EFX	✓(SD)		✗(Theorem 5.2)	
			(even for $m = 4$)	
EF1	✓(SD)	✓(RSD)	✗[9]	✓(RSD) ($m \geq 5$)
MMS	✓(SD)		✗[10]	

Table 5.1: Existence of SPF Mechanisms for various types of Valuations

Given the above results, in [43, 130], the authors relax EF and introduce EF up to the most-valued good or EF1. An EF1 allocation is always guaranteed to exist even for indivisible goods and can be computed in polynomial time by the cycle-elimination algorithm [130]. It is interesting to consider a stronger property of EF1, which is EF up to the least-valued good, known as EFX [47]. EFX always exists for upto three agents [50]. For indivisible goods, another fairness criterion considered is MMS [43], where each agent's utility is at least its MMS guarantee. The MMS guarantee is the worst-case value an agent receives when partitioning the goods and others choose before it. MMS allocation is guaranteed to exist for up to two agents [167].

The above existential and complexity results assume that each agent's preferences (determined using their valuations for each bundle) are known. In this work, we are interested in preference elicitation to prevent manipulations. Hence we study the existence of truthful or SP (Strategy-Proof) mechanisms that ensure fairness or *Strategy-Proof Fair* (SPF). A direct-revelation mechanism takes all the input valuation functions and returns an allocation. A direct-revelation mechanism is SPF if it ensures fair allocation when no agent can gain higher utility by misreporting. In mechanism design literature, it is standard to introduce payments to design truthful mechanisms, especially in auction settings [56, 88, 186]. In this work, we focus on the basic model of fair mechanism design without money. When the goods are divisible, [28, 142] prove that no deterministic SP mechanism (without monetary transfers) is proportional or even approximately proportional for complete allocation. Since EF is a stronger property, having SP mechanism for EF is also impossible. It is known that there exist randomized SP mechanisms which ensure EF when the goods are divisible [145]. There are other works [42, 52, 57] which give SPF mechanisms without money for divisible goods. Bouveret et al. 2011 [40] show that sequential allocation is strategy proof when agents have identical rankings. This way of allocation is referred to as *Picking Sequences*.

SPF Mechanism for Indivisible goods Lipton et al. 2004 [130] prove that it is impossible to design a truthful mechanism that achieves minimum envy or EF by providing a counterexample. When there are two agents ($n = 2$) and the number of goods (m) is greater than 5, there cannot be a deterministic SP mechanism with complete allocation for EF1 even for additive valuations [9]. There are impossibility results for MMS in [10]; the authors prove that for two agents, there is no truthful mechanism that ensures better than $\frac{1}{m/2}$ -MMS allocation.

Our Contribution

1. We study the EFX property for two agents, where it is guaranteed to exist. From Amanatidis et al. 2017 [9], having an SP mechanism for EF1 with two agents and more than 5 goods is impossible. EFX being a stronger property, also follows the same result for the given setting. We provide an example that proves that designing an SP mechanism for EFX is impossible even when the number of goods is 4.
2. Aligning with the results of Amanatidis et al. 2017 [9], we provide examples to show that the greedy round-robin algorithm and cycle-elimination algorithm for finding EF1 are manipulable. When agents have identical additive allocations, greedy round robin provides allocations that is EF1 as well as strategy proof.

3. Given that the valuations can be very complex to represent in general, we restrict ourselves to the simpler case of (SM) *single-minded* agents. SM bidders is very common in the auction literature multi-item setting [140, 168]. In such a setting we provide (SD) *(Serial Dictatorship Mechanism)* that again extends greedy to obtain SP mechanism for EFX, EF1, MMS. SD also provides proportional allocations when they exist.

In Table 5.1, we summarize all the results for the existence of an SP mechanism for various fairness criteria. When the agents are single-minded SD is a direct SP mechanism which also ensures EF1, EFX, MMS and proportionality when it exists. When the valuations are (additive) identical, RSD is an SP mechanism that ensures EF1. Additive valuations are the most well-studied in literature. For EF, proportionality, and EFX, there are counterexamples when there are 2 agents to prove that there cannot exist an SP mechanism when valuations are additive. Even for MMS and EF1 under additive valuations, the results are for 2 agents.

5.2 Preliminaries

Consider the problem of division of indivisible resources. We represent each instance by $\langle N, M, V\rangle$ which are formally defined below,

- Finite set of agents $N = \{1, \dots, n\}$
- Finite set of indivisible goods $M = \{1, \dots, m\}$.
- Valuation functions V where $v \in V$ denotes a particular profile and $\forall i \in N$, $v_i : 2^M \rightarrow \mathbb{R}_+$. Let v_{-i} be the valuation profile of all agents, excluding i.
- We assume v_i is monotonic, $\forall i \in N, \forall S \subseteq T \subseteq M, v_i(S) \leq v_i(T)$
- Additive valuations imply for any $S \subseteq M, v_i(S) = \sum_{j \in S} v_i(\{j\})$

- Identical valuations imply $\forall i, j \in N, \forall S \in M, v_i(S) = v_j(S)$. Identical additive valuations imply v_i is both identical and additive.

- Single minded agents with desirable bundles $D = (D_1, \ldots, D_m)$. The valuation of an agent $i \in N$ is given by, for a $c \in \mathbb{R}_+$

$$\forall S \in M,\ v_i(S) = \begin{cases} c, & \text{if } S \supseteq D_i \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

- The set of all possible complete allocations, $\mathcal{A}$. Given $A \in \mathcal{A}$ denotes a specific allocation, and A_i is allocation per agent. By complete allocation, we mean if there are m goods then $\forall A, \sum_i |A_i| = m$, assuming each resource can be allocated only to a single agent.

We define the relevant fairness notions below with examples,

Definition 5.1 (*Proportionality*). *Given an instance $\langle N, M, V \rangle$, the allocation A is proportional* iff $\forall i \in N$,

$$v_i(A_i) \geq \frac{1}{n} v_i(M)$$

Example 8: Proportional Division

Consider two agents 1 and 2 and three goods a, b, c. v_i is given below where $(x, y) \in \{(a, b), (b, c), (c, a)\}$.

	$v(a)$	$v(b)$	$v(c)$	$v(x, y)$	$v(a, b, c)$
1	10	20	15	30	30
2	10	20	15	30	30

Possible proportional allocations are when 1 receives item b and 2 receives goods $\{a, c\}$ or vice versa. Also when agent 1 receives c and agent 2 receives goods $\{a, b\}$ or vice versa.

Definition 5.2 (*Envy-freeness* (EF))**.** *For $\langle N, M, V\rangle$ an allocation A is envy-free* iff*,*

$$\forall i, j \in N \quad v_i(A_i) \geq v_i(A_j)$$

Consider 2 agents 1 and 2, two goods a, b. For agent 1, $v_1(a) = 20, v_1(b) = 10$ and for agent 2, $v_2(a) = 10, v_2(b) = 20$. It is envy-free to allocate a to agent 1 and b to agent 2.

Both the notions of proportionality and EF are too strong in the case of indivisible goods and are not guaranteed to exist. Consider the case when there are two agents and only one item, it is impossible to have any allocation that is either EF or even proportional. When the valuations are sub-additive, every EF allocation is proportional as shown in Figure 5.1. In [130], the authors define the following notion weaker than EF.

Definition 5.3 (*EF1*)**.** *For $\langle N, M, V\rangle$ an allocation A is EF1* iff $\forall i, j \in N$, $\exists a \in A_j$ *s.t.,*

$$v_i(A_i) \geq v_i(A_j \backslash \{a\})$$

EF1 allocation always exists for general monotone valuations. Another relaxation of EF stronger than EF1 is defined below,

Definition 5.4 (*EFX*)**.** *For $\langle N, M, V\rangle$ an allocation A is EFX* iff, $\forall i, j \in N$, $\forall a \in A_j$ *such that,*

$$v_i(A_i) \geq v_i(A_j \backslash \{a\})$$

We saw before that EF allocation is not possible when there are two agents and only one item. But an allocation where the item is assigned to either 1 or 2 is both EF1 and EFX.

Unlike EF1, EFX is guaranteed to exist only for three agents or when agents have identical valuations. The relations between EF, EFX and EF1 is represented in Figure 5.1 for any general monotonic valuations. Another relaxation is defined by Budish 2011 [43],

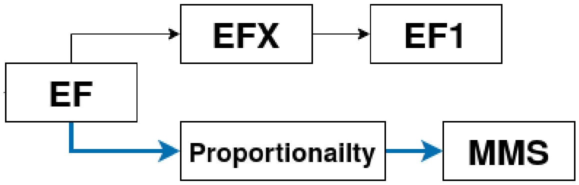


Figure 5.1: Relation between various fairness criteria [41]

Definition 5.5 (*Maximin Share* (MMS))**.** *For* $\langle N, M, V\rangle$ *an allocation* A *is MMS* iff $\forall i \in N$, $v_i(S_i) \geq \mu_i$ *where* $\mu_i = \max_{A\in\Pi_n(M)} \min_{A_j\in A} v_i(A_j)$

Example 9: MMS Division

Consider there are two agents 1 and 2, three goods a, b, c. We consider additive valuations for both agents. Let the valuation of each item be given as follows,

	$v(a)$	$v(b)$	$v(c)$
1	10	20	40
2	10	40	20

In this example $\mu_1 = \mu_2 = 30$. Agent 1 gets c and 2 gets a, b would be an MMS allocation.

There is no relation between EF1/EFX and MMS allocations; for two agents MMS implies EFX. Although any allocation which is proportional is also MMS when the valuations are sub-additive, Figure 5.1. There exists an MMS allocation for 2 agents but it may not exist for more than two agents.

5.2.1 Strategy-Proof Mechanisms

In mechanism design, we assume the agents are self-interested and strategic. The agents have private information (valuation over goods) that is indispensable for the desired outcome. The agents may or may not reveal their private information based on their individual utility. Mechanism design deals with the two-fold problem of i) *Preference Elicitation* and

ii) *Preference Aggregation*. In the former, one explores the specific mechanism in which the agents' best interest lies in revealing their true valuations. The latter, nonetheless challenging, is the problem of obtaining the desired outcome, once the true valuations are known. In our case, this would be finding the fair allocation. There are two kinds of approaches to solving the problem of preference elicitation. 1) Direct Mechanism 2) Indirect Mechanism. We focus on a direct mechanism which is defined as follows,

Definition 5.6 (*Direct Mechanism*)**.** *The direct mechanism ($\mathbb{M}$) maps the true valuations of the agents to the desired outcome. It is a mapping from the valuations of the agents to the space of allocations $\mathbb{M} : V \to \mathcal{A}$.*

Definition 5.7 (*Deterministic SP Mechanism*)**.** *A deterministic mechanism $\mathbb{M}$ is strategy-proof (SP), if $\forall v$, $\forall i \in N$,*

$$v_i(\mathbb{M}(v_i, v_{-i})) \geq v_i(\mathbb{M}(v'_i, v_{-i})), \ \forall v'_i \ \forall v_{-i}$$

where v'_i is a misreported valuation.

We also look for little weaker mechanisms, in the context of identical valuations.

Definition 5.8 (*Deterministic NSP Mechanism*)**.** *A deterministic mechanism $\mathbb{M}$ is Nash strategy-proof (NSP), if $\forall v$, $\forall i \in N$ when other agents report truthfully,*

$$v_i(\mathbb{M}(v_i, v_{-i})) \geq v_i(\mathbb{M}(v'_i, v_{-i})), \ \forall v'_i$$

where v'_i is a misreported valuation.

Note that, in NSP, it is the best response to each agent to report truthfully if others are reporting truthfully. SP is a stronger notion of truthfulness – no matter what others are reporting, it is the best response for each agent to report truthfully.

Definition 5.9 (*Strategy Proof Fair Mechanism* (SPF))**.** *A mechanism $\mathbb{M}$ is SPF* iff *$\mathbb{M}$ is SP or NSP and fair (for the given fairness condition).*

Before we discuss the existence of SPF mechanisms for various fairness criteria, we would like to state an observation that makes our search easier. The observation is based on the relationship between the various fairness criteria given in Figure 5.1,

Observation 5.1. *For any two fairness criteria X and Y, if $X \implies Y$ from Figure 5.1, i.e., every allocation that satisfies X also satisfies Y. We can conclude that if there does not exist an SP mechanism for Y, then there will not exist an SP mechanism for X.*

With this background, we first present the impossibilities of SP and fair mechanisms for additive valuations.

5.3 Impossibility of SP and Fair Mechanisms

As we have discussed before, fair allocations may still cause unrest among agents if some agents choose to lie and agents adhering to the rules and revealing their true valuations forgo the benefits they could have received. In this work, we are concerned about the existence of truthful mechanisms that can implement the fairness definitions defined above.

EF

Lipton et al. 2004 [130] raises the question of the existence of truthful mechanisms that implement EF.

Theorem 5.1 (Lipton et al. 2004 [130])**.** *Any mechanism that returns an allocation with minimum possible envy cannot be truthful. The same is true for any mechanism that returns an envy-free allocation whenever there exists one.*

As proof, the authors provide an example with two agents with additive valuation functions, where every possible envy-free allocation can be manipulated by either of the agents. An SP mechanism for EF, ($\mathbb{M}^{EF}$) would select from $\mathcal{A}^{EF}$, i.e., a set of all EF allocations. If there exists a valuation profile v, where $\mathcal{A}^{EF}_v$ be all possible EF allocations for v, $\forall A^{EF} \in \mathcal{A}^{EF}_v$ and with strict inequality for atleast one A^{EF}.

$$\exists i,\ \exists v_i'\ s.t.,\quad v_i(\tilde{A}_i^{EF}) \geq v_i(A_i^{EF}),\quad \forall \tilde{A}^{EF} \in \mathcal{A}_{v'}^{EF} \tag{5.2}$$

We know that for any deterministic SP mechanism that ensures EF, $\mathbb{M}^{EF}(v) \subseteq \mathcal{A}_v^{EF}$, hence the Equation 5.2 holds for $\mathbb{M}^{EF}(v)$ which implies that no matter the mechanism, it is always manipulable by certain agent i under the valuation profile, v.

Proportionality

For sub-additive valuations, proportionality is a stronger property than MMS (Figure 5.1). [10] prove that for 2 agents, there is no SP mechanism that ensures better than $\frac{1}{m/2}$-MMS allocation. Hence, it is impossible to have SP mechanism which ensures MMS and hence proportionality for 2 agents. We prove the same by constructing an example guided by Equation 5.2.

Example. Consider $n = 2$, $m = 3$, agents $\{1, 2\}$ and goods, $\{a, b, c\}$. The true valuations v are given by Table 5.2a. For truthful reporting, there are 2 possible proportional allocations $\mathcal{A}_v^{prop}$ given by Table 5.2b. For the first allocation A^I, agent 1 obtains a value of 20. If 1 reports v_i' as given in Table 5.2c, then the only possible proportional allocation is given in Table 5.2d, the value for which is 30, is strictly better than what she was offered. Similarly, for the next allocation A^{II}, agent 2 has the incentive to misreport.

EFX

In [9], they prove that it is impossible to design SP mechanism for EF1 for $n = 2$ and $m \geq 5$. Since EFX (Definition 5.4) is a stronger property the same result holds when $m >= 5$. We prove that it is also impossible to have an SP mechanism for EFX when $m = 4$.

Theorem 5.2. *Any mechanism that returns an allocation that is EFX cannot be truthful even in the case of additive valuations.*

	$v(a)$	$v(b)$	$v(c)$
1	20	10	5
2	5	10	20

(a) The true values

	1	2
A^I	a	bc
A^{II}	ab	c

(b) $\mathcal{A}_v^{prop}$

	$v(a)$	$v(b)$	$v(c)$
1	10	10	10
2	5	10	20

(c) Agent 1 misreports

	1	2
$\tilde{A}^I$	ab	c

(d) $\mathcal{A}_{v'}^{prop}$

Table 5.2: Counter example for proportionality

Proof. Consider an example where $n = 2$, we have agents $\{1, 2\}$ and $m = 4$, $\{a, b, c, d\}$. For truthful reporting v as given in Table 5.3a, there are 4 possible EFX allocations $\mathcal{A}_v^{EF}$ given by Table 5.3b. For the first two allocations A^I, A^{II}, agent 1 receives b which it values at 100 or $\{b, c\}$ which it values 120. If agent 1 reports v_i' as given in Table 5.3c, where the total valuation is the same i.e., 200. Under this misreport, the only possible EFX allocations are given in Table 5.3d in which agent 1 receives at least 140 which is at least as good as the value it received for truthful reporting. Similarly for the next two allocations A^{III}, A^{IV} agent 2 has an incentive to misreport. Hence there are only four possible EFX allocations and for each allocation at least one agent has an incentive to misreport. □

Thus, for additive valuations, there is an instance where no SP mechanism can be EFX.

EF1

In this subsection, we explore the existing algorithms that find EF1 allocations and prove that these are manipulable. We provide an instance with $n = 2$ for each case.

	$v(a)$	$v(b)$	$v(c)$	$v(d)$
1	40	100	20	40
2	100	40	20	40

(a) The true values

	1	2
A^I	b	acd
A^{II}	bc	ad
A^{III}	bd	ac
A^{IV}	bdc	a

(b) $\mathcal{A}_v^{EFX}$

	$v(a)$	$v(b)$	$v(c)$	$v(d)$
1	90	70	15	25
2	100	40	20	40

(c) Agent 1 misreports

	1	2
$\tilde{A}^I$	bd	ac
$\tilde{A}^I$	bdc	a

(d) $\mathcal{A}_{v'}^{EFX}$

Table 5.3: Counter example for EFX

Greedy round-robin Algorithm

In [47], the authors provide a simple algorithm for obtaining EF1 allocations when the valuations are additive. It involves the following steps,

- Fix an arbitrary order on the agents
- Allocate the first agent its most valuable good
- The next agent is allocated it's most valuable among the remaining goods
- The algorithm terminates when all the goods are allocated

The following example shows that the above algorithm can be manipulated by the agents.

Proposition 5.1. *Greedy round-robin algorithm is manipulable for additive valuations.*

Proof. Consider $n = 2$, $\{1, 2\}$, $m = 5$, $\{a, b, c, d, e\}$ where the valuations are additive and given by Table 5.4a. When $n = 2$, there is only two possible orders among the agents. When applying greedy algorithm with 1 followed by 2 or 1 → 2, agent 1 gets $\{a, c, d\}$

	$v(a)$	$v(b)$	$v(c)$	$v(d)$	$v(e)$
1	12	10	8	6	1
2	1	10	8	6	9

(a) The true values

	1	2
1 → 2	acd	be
2 → 1	ac	bde

(b) EF1 allocations

	$v(a)$	$v(b)$	$v(c)$	$v(d)$	$v(e)$
1	10	12	8	6	1
2	1	10	8	6	9

(c) Agent 1 misreports

	1	2
1 → 2	abd	ce

(d) EF1 allocation

	$v(a)$	$v(b)$	$v(c)$	$v(d)$	$v(e)$
1	12	10	8	6	1
2	1	10	8	8	5

(e) Agent 2 misreports

	1	2
2 → 1	ad	bce

(f) EF1 allocation

Table 5.4: Greedy round-robin is manipulable

(Table 5.4b) with a value of 26. If the agent misreports its value as given in Table 5.4c, the allocation that agent 1 gets is $\{a, b, d\}$ which it values at 28 that is strictly more than when it was truthful. Similarly, agent 2 can misreport to an advantage when the order is $2 \rightarrow 1$. It improves its allocation from $\{b, d, e\}$ (Table 5.4b) that it values at 25 to $\{b, c, e\}$ whose value is 27. □

Cycle-elimination Algorithm

The greedy method fails for general valuations, instead the cycle-elimination algorithm [130] provides EF1 solution in polynomial time in general. The algorithm is as follows,

- Goods are allocated in arbitrary order

- An envy graph is maintained where the agents are the vertices and a directed edge $i \rightarrow j$ represents that agent i envies agent j under the current allocation.
- The next item is allocated to the agent with no incoming edge. If there is a cycle, it can be eliminated by exchanging the goods of the agents that form the cycle, with the ones they envy.

We show that the above algorithm is manipulable by the agents.

Proposition 5.2. *Cycle-elimination algorithm is manipulable even for identical valuations.*

1	2	graph
d		$1 \leftarrow 2$
d	a	$1 \leftarrow 2$
d	ab	$1 \rightarrow 2$
dc	ab	$1 \rightarrow 2$

(a) Cycle-elimination on v

1	2	graph
d		$1 \leftarrow 2$
d	a	$1 \rightleftharpoons 2$
a	d	no envy
ab	d	$1 \leftarrow 2$
ab	dc	$1 \leftarrow 2$

(b) Cycle-elimination on v'

1	2	graph
d		$1 \leftarrow 2$
d	a	$1 \rightleftharpoons 2$
a	d	no envy
a	bd	$1 \rightarrow 2$
ac	bd	$1 \leftarrow 2$

(c) Cycle-elimination on v'

Table 5.5: Cycle-elimination is manipulable

Consider $n = 2$, $\{1, 2\}$, and $m = 4$, $\{a, b, c, d\}$. Let $(x, y) \in \{(a, b)(b, c), (a, c)\}$ where the valuations v of the agents are identical and given by $v(a) = v(b) = v(c) = 5, v(d) = 10$ and $v(x, y) = 16$ for $x, y \in \{a, b, c, d\}$. The value of other subsets not mentioned are additive. When we run the cycle-elimination algorithm, the steps are as given in the Table 5.5a. It is easy to see that the agent who gets the item d (w.l.o.g we assume agent 1 gets d) always ends up with a value 15 and can try to increase the utility by gaining the other bundle whose value is 16. Now consider the following misreported valuation by the agent who gets item d. $v'(a) = v'(b) = v'(c) = 5, v'(d) = 4$ and $v'(x, y) = 16$, for $x, y \in \{a, b, c, d\}$. With the misreported valuation, we again run the cycle-elimination algorithm and there can be two possible outcomes as presented in Table 5.5b, 5.5c. We see that in these cases the agent 1 receives $\{a, b\}$ or $\{a, c\}$ which it values at 16 i.e., strictly more than the previous value for $\{d, c\}$ that is 15. In this example, we prove that there is always an agent that can manipulate to increase its utility.

In the above case, we considered an example of (general) identical valuations. In fact it is also possible to manipulate cycle-elimination for (additive) identical valuations.

Example. Consider $n = 2$ and $m = 3$ where the agents have the following (additive) identical valuation v and the agent that does not receive the good c misreports the valuation to v' also given below. For many orderings over m that the algorithm chooses, the value obtained for v' is as good as v. But when the ordering is chosen to be a then b then c or (b, a, c), the agent manipulating ensures a value of 17 as opposed to just receiving 5.

$v(a)$	$v(b)$	$v(c)$	$v'(a)$	$v'(b)$	$v'(c)$
5	5	12	4	5	12

5.4 Identical Additive Valuations

When valuations are identically additive, we know that picking sequences is strategy proof [40]. Based on picking sequences, we provide an algorithm, (RSD) *(Repeated Serial Dictatorship)* to obtain truthfulness while ensuring EF1.

Repeated Serial Dictatorship is EF1 and (i) SP when $m \leq n$ and (ii) NSP when the valuations are identical and additive. Given that RSD implements a greedy round-robin algorithm under additive valuations, the output allocation A is EF1.

(i) Case $m \leq n$: under this case, the while loop in Algorithm 8 runs for m-iterations, given $m \leq n$, each agent only gets one chance to participate and select the item x. The ordering chosen by the algorithm is independent of the agent valuations and hence cannot be manipulated. From the algorithm, we know that given the remaining goods R,

$$A_i = x \in \underset{j \in R}{argmax}\ v_i(j)$$

If the agent misreports s.t. $y \in \underset{j \in R}{argmax}\ v_i'(j)$ and $x \neq y$. The agent receives y s.t. under true valuations, $v_i(y) \leq v_i(x)$ and hence cannot strictly increase its utility.

Algorithm 8 Repeated Serial Dictatorship Mechanism (RSD)

```
Input: $\langle N, M, V \rangle$, $V$ is identical additive
Output: $(A_1, A_2, \ldots, A_n) \in \mathcal{A}^{EF1}$
Set an arbitrary but fixed order on the agents, w.l.o.g, $(1, 2, \ldots, n)$
$A_i = \phi, \quad \forall i$
$R = M$ (goods remaining after each iteration)
$i = 0$ (agent number)
while $R \neq \phi$ do
    $x \in \underset{j \in R}{argmax}\ v_i(j)$
    $A_i = A_i \bigcup x$
    $R = R \setminus x$
    $i = (i + 1) \mod n$
end while
```

(ii) Case v is Identical (Additive): Let v be the truthful report, we assume $v^1 \geq v^2 \geq \ldots \geq v^m$ be the value all the agents have for the m goods in decreasing order. The Algorithm 8 will continue for m rounds and assign the goods in this order itself. The goods remaining at round j is given by $R_j = \{v^j, \ldots, v^m\}$. Let us assume an agent i gets allocated k items before the algorithm terminates, then it selects from the following subsets and receives the items it values the most in each of these,

$$\{R_i, R_{i+n}, \ldots, R_{i+kn}\}$$

Hence $A_i = \{v^i, v^{i+n}, \ldots, v^{i+kn}\}$. If the agent i misreports and the remaining agents report truthfully, in any of the rounds w.l.o.g, i^{th} round s.t., the relative ordering between the items changes, then the agent might face the two possible sets in the next round $(i + n)$,

- Misreport s.t. agent i gets item p instead of i, $p \leq n+i-1$, then the set it faces in the next rounds is $\{R_{n+i}, \ldots, R_{i+kn}\}$. Hence the items allocated are $A'_i = \{v^p, v^{i+n}, \ldots, v^{i+kn}\}$. It can be clearly verified that, $v_i(A_i) \geq v_i(A'_i)$, hence no incentive to misreport.

- Misreport s.t agent i gets item p where, $k'n + i > p \geq (k'-1)n + i$, $k' \geq 2$ then the sets i faces are

  $$\{R_{n+i-1} \setminus \{p\}, \ldots, R_{i+(k'-1)n-1} \setminus \{p\}, R_{i+k'n}, \ldots, R_{i+kn}\}$$

  Hence the items allocated are

  $$A''_i = \{v^p, v^{n+i-1}, \ldots, v^{i+(k'-1)n-1}, v^{i+k'n}, \ldots, v^{i+kn}\}$$

  Using the fact that $p \geq i + (k'-1)n$, we know that $v^{i+(k'-1)n} \geq v^p$. Hence we compare the sets A_i and A''_i ($\succeq$ represents element-wise comparison) as follows to obtain $v_i(A_i) \geq v_i(A''_i)$,

  $$\{v^i, v^{i+n}, \ldots, v^{i+(k'-2)n}, v^{i+(k'-1)n}, v^{i+k'n}, \ldots, v^{i+kn}\} \succeq \\ \{v^{i+n-1}, v^{i+2n-1}, \ldots, v^{i+(k'-1)n}, v^p, v^{i+k'n}, \ldots, v^{i+kn}\}$$

This completes the proof for truthfulness for RSD under additive and identical valuations.

5.5 Single-Minded Agents

In this section, we restrict to a simpler valuation profile. We assume the agents are (SM) single minded. SM agents are only interested in a single bundle of goods D. Upon receiving the specific bundle or any super-set they get a positive utility and zero value for any other bundle (Formally given by Equation 5.1). The problem instance is $\langle N, M, D\rangle$.

Algorithm 9 Serial Dictatorship Mechanism (SD)

```
Input: $\langle N, M, D \rangle$, $D = (D_1, D_2, \ldots, D_n)$
Output: $(A_1, A_2, \ldots, A_n)$
Order the agents s.t. $|D_1| \leq |D_2| \leq \ldots \leq |D_n|$ (Ties broken arbitrarily)
$i = 0$ (agent number)
$R = M$ (goods remaining after each iteration)
while $R \neq \phi$ do
    Let $D_i$ be the preferred set for the current agent $i$
    if $D_i \subseteq R$ then
        $A_i = D_i$
        $R = R \setminus D_i$, $i = i + 1$
    else
        if $i < n$ then
            $i = i + 1$
        else
            $A_i = R$
            $R = \phi$
        end if
    end if
end while
```

Observation 5.2. *When all the agents are SM, any allocation is MMS and EF1. The μ_i in Definition 5.5 is 0 in this setting when $n > 1$. Hence, allocating all the goods to one agent is also MMS. Similarly, all possible allocations satisfy EF1. If an agent i receives its desired bundle or super-set then it doesn't envy any agent. If an agent j receives D_i, then* *removing any item $x \in D_i$ would remove envy. If no agent receives D_i as a whole, there is no envy.*

Based on the observation, we extend greedy round-robin algorithm to design a SD *(Serial Dictatorship)* mechanism is SP since it is also a picking sequence. SD trivially satisfies EF1 and MMS and we prove that it also satisfies EFX.

Theorem 5.3. *The Serial Dictatorship Mechanism is strategy-proof (SP) and also satisfies EF1, MMS and EFX when the agents are single-minded.*

Proof. In the Algorithm 9, the while loop can run for a maximum of n rounds. This means each agent i has only one round in which it can be allocated the preferred bundle D_i. The ordering is according to the increasing cardinality of D_i. An agent can manipulate the ordering by reporting its desired bundle as D'_i s.t., $|D'_i| < |D_i|$. This means the agent will be allocated if at all a bundle that it does not desire. If $|D'_i| > |D_i|$ then the probability that the agent gets any allocation is strictly less than when it reports truthfully. Hence the agent does not have any incentive to manipulate the ordering.

Given that the agent cannot manipulate the ordering. At any round, it is optimal for the agent to report truthfully the desired set D_i. Now we prove that the allocation A obtained from SD satisfy the following fairness criteria,

- (EF1 and MMS). This is trivially true due to Observation 5.2 which states that any allocation is EF1 and MMS when we have SM agents.
- (EFX). Let us assume k agents, denoted by L (lucky), are allocated their desired sets and hence do not have any envy. If $k \neq n$, then $n - k$ agents, denoted by U (unlucky), did not receive their desired subset. From the algorithm, we know that for any agent $i \in U$, $D_i \nsubseteq R_i$ where R_i is the set of goods at the beginning of i^{th} round.
    - $\forall i, j \in U$, i does not envy j, because j is allocated empty bundle unless j is the agent appearing at the last n and receives the items remaining. In this case, since agent i is given the chance to choose before j which clearly shows it cannot envy j.

- $\forall i \in U$, $\forall \bar{i} \in L$, if $\bar{i} < i$, then $|D_{\bar{i}}| < |D_i|$, hence agent i cannot envy $\bar{i}$. If $|D_{\bar{i}}| = |D_i|$ then removing any item from the bundle of $\bar{i}$ will remove envy. Hence it still satisfies EFX.
- $\forall i \in U$, $\forall \bar{i} \in L$, if $\bar{i} > i$, then $D_{\bar{i}} \subseteq R_i$ hence $D_{\bar{i}} \neq D_i$, hence the agent i does not envy $\bar{i}$

Hence the allocation is EFX. This concludes the proof for the theorem. Hence SD is SP and provides allocations that satisfy EF1, MMS and EFX. □

Note on Proportionality. When the agents are SM, proportional allocation exists if, $v_i(A_i) \geq \frac{1}{n}v_i(D_i) > 0, \ \forall i \in N$ The above is true only when all the agents get their desired bundle. If such a solution exists then it is easily found by the SD.

5.6 Conclusion

In the literature, there are many algorithms for finding a fair division of resources. Yet such algorithms may not be really fair if one agent can manipulate it by misreporting its value to obtain higher utility. We show that greedy round-robin and cycle-elimination algorithms are manipulable. In general, we study the possibility of having strategy-proof, deterministic mechanisms without money which ensure various criteria of fairness like EF, proportionality, EFX, EF1, and MMS. It is known that such a mechanism does not exist for EF, proportionality and MMS under additive valuations. It also does not exist for EF1 under additive valuations when the number of items is more than 5. We prove that it does not exist for EFX even when the number of items is 4. Given these impossibility results, we look into settings where agents have simpler valuation types like single minded bidders. Under this assumption, we provide a strategy proof algorithm SD. SD satisfies all fairness criteria except EF. RSD satisfies EF1.

Part B – Fair Decisions for Groups

Chapter 6

Part B - Preliminaries and Related Work

AI systems are ubiquitous in the current times, facilitating numerous real-world, even real-time, decision-making applications. The existing models achieve near-optimal results for specific performance measures, often obtained at the cost of certain ethical constraints like fairness and privacy. More recently, researchers have uncovered the prejudiced predictions of such models towards certain demographic groups, especially in machine learning predictions. Due to existing bias against a certain race, gender or age, the data available is often biased. The prejudices in the data, amplified by the algorithms trained only for higher accuracy, lead to unfair decisions for specific groups. Moreover, such algorithms made public on various online platforms potentially leak private information of the individual data used in training. Researchers have studied different notions of fairness and privacy and ways to ensure them in the machine learning framework.

6.1 Machine Learning

Machine Learning (ML) is a subset of AI wherein we use existing data to train mathematical models. The trained model is then used for predictions on a novel, unseen data points. ML models are in other words a functional mapping from the input to output learnt

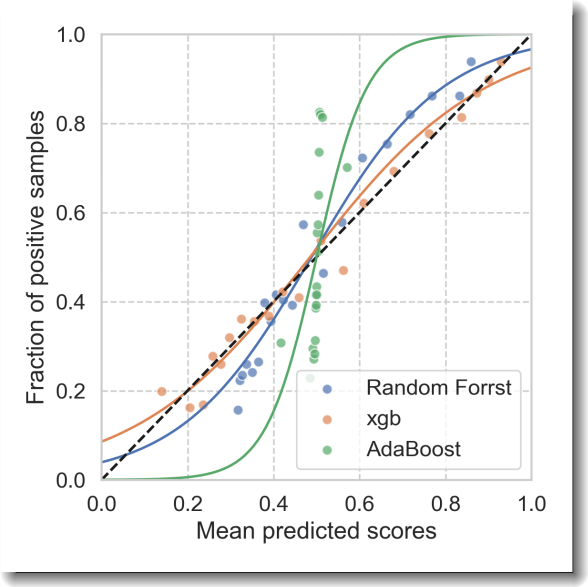


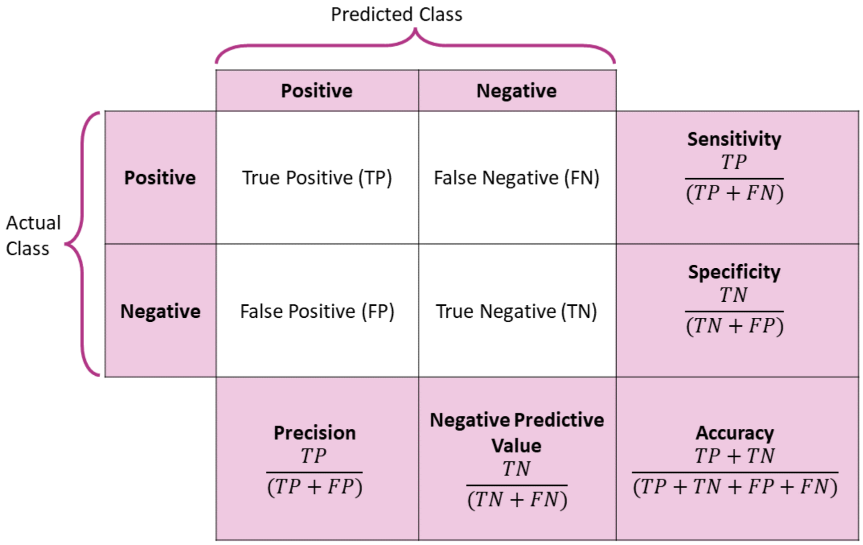
Figure 6.1: Platt scaling for probability calibration [51]

based on specific training methods. *Classification* is an important task within the field of ML that involves identifying the category (or categories) to which an observed data point belongs. For classification, the models require supervision i.e., the classes an existing data point belong to which serves as a ground truth while training.

Mathematically, let's say we have the observations: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where $x_i \in$ Feature space X , and $y_i \in$ Label space Y. Our classifier is a function h that maps the feature space to the label space, $h : X \rightarrow Y$ Note that Y consists of all the possible class labels. In the case of a binary classification problem, we typically see two classes, $Y = \{0, 1\}$. We desire to have a classifier that is *calibrated*. Informally, this means that the "score" (between 0 and 1) produced by our classifier actually corresponds to the real probability of the predicted outcome. Formally,

Definition 6.1 (*Calibration*)**.** *A classifier h is said to be calibrated if*

$$\Pr(Y = 1|h(X) = r) = r \tag{6.1}$$



		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN)	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP)	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Figure 6.2: Confusion Matrix for a binary classifier [173]

Platt's Scaling. One way of achieving this is through *Platt's scaling*. Platt's scaling is a technique to transform classifier model outputs into a probability distribution over classes. This will give us a calibrated classifier. According to Barocas et al. 2019 [24], Platt's scaling treats a non-calibrated score as a single feature and attempts to fit a regression model against the target variable based on this feature. We try to fit Y to a sigmoid-like curve S, where $S = \frac{1}{1+e^{-h(x)}}$. And then try to minimize the log-loss given by $-\mathbb{E}[Y \cdot log(S) + (1-Y) \cdot log(1-S)]$

Confusion Matrix. In order to evaluate the classifier, the widely used metric is accuracy. Informally it represents the total number of samples which have been correctly classified. In general, any performance measure is based on the components of a confusion matrix. As shown in Figure 6.2, FN is False Negative, TP is True Positive, TN is True Negative, and FP is False Positive. We define these as:

$$Accuracy = \frac{TN+TP}{P+N} = \frac{TN+TP}{TN+FN+TP+FP}$$

$$\begin{aligned}
\textit{Precision} &= PPV = P(Y = 1 \mid \hat{Y} = 1) = \frac{TP}{FP + TP} \\
\textit{Recall} &= TPR = P(\hat{Y} = 1 \mid Y = 1) = \frac{TP}{FN + TP} \\
\textit{Negative Predictive Value (NPV)} &= \frac{TN}{TN + FN} \\
\textit{False Positive Rate (FPR)} &= P(\hat{Y} = 1 \mid Y = 0) = \frac{FP}{TN + FP} \\
\textit{False Negative Rate (FNR)} &= P(\hat{Y} = 0 \mid Y = 1) = \frac{FN}{FN + TP} \\
\textit{True Negative Rate (TNR)} &= P(\hat{Y} = 0 \mid Y = 0) = \frac{TN}{TN + FP} \\
\textit{True Positive Rate (TPR)} &= P(\hat{Y} = 1 \mid Y = 1) = \frac{TP}{TP + FN}
\end{aligned}$$

Precision is also called *Positive Predictive Value (PPV)*. Recall is also referred as *True Positive Rate (TPR)* or *Sensitivity*. Further note that, $TPR + FNR = 1$ and $TNR + FPR = 1$.

Receiver Operating Characteristics (ROC). ROC is a property of a distribution (X, Y): It gives the optimal TPR for a given FPR on the distribution. It is an estimate of how predictive the score is of the target variable. A quantitative metric derived from the ROC is the Area Under the Curve (AUC), which is a measure of predictiveness. An area of $\frac{1}{2}$ corresponds to random guessing, and an area of 1 corresponds to perfect classification. The ROC curve is shown in Figure 6.3.

Given the basic components in an ML framework, now we look at ML models through a fairness perspective. We first identify the biases that arise in an ML pipeline. Then we discuss the quantitative fairness measures proposed in the literature and some of their properties. Finally, we also discuss some existing approaches for ensuring fairness in ML.

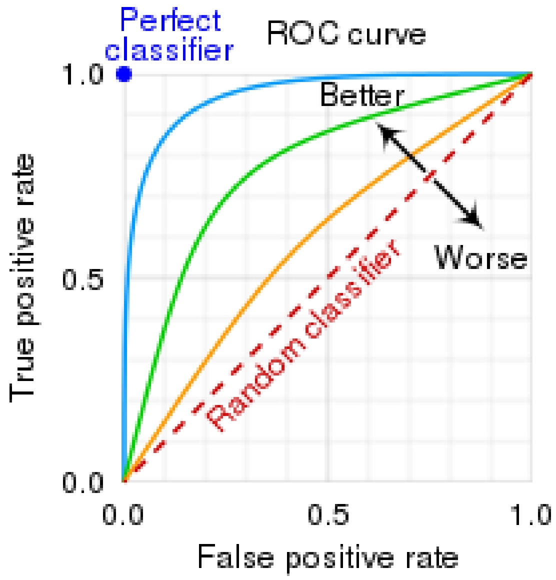


Figure 6.3: Receiver Operating Characteristics (ROC) [source: Wikipedia]

6.2 Fairness in Machine Learning

There are two potential sources of unfairness in machine learning outcomes - (i) humans and data, and (ii) machine - algorithms and associations. Biases exist in many forms and shapes [141], and here, we discuss some come types of biases briefly.

1. ***Historical Bias.*** This arises even if the data is perfectly sampled and selected if there already exists bias and socio-technical issues in the world. Although the data generation process reflects the world accurately, the world as it is often led to a model that could inflict harm on a population.

   As an example, historical bias can be found in a 2018 image search result, while searching for women CEOs image the search results were more biased towards male CEOs. There were only 5% Fortune 500 women CEOs and although the search results were reflecting the reality, it is worth pondering whether or not the search algorithms should reflect it. Another example is word embeddings, the learned vector representations of words

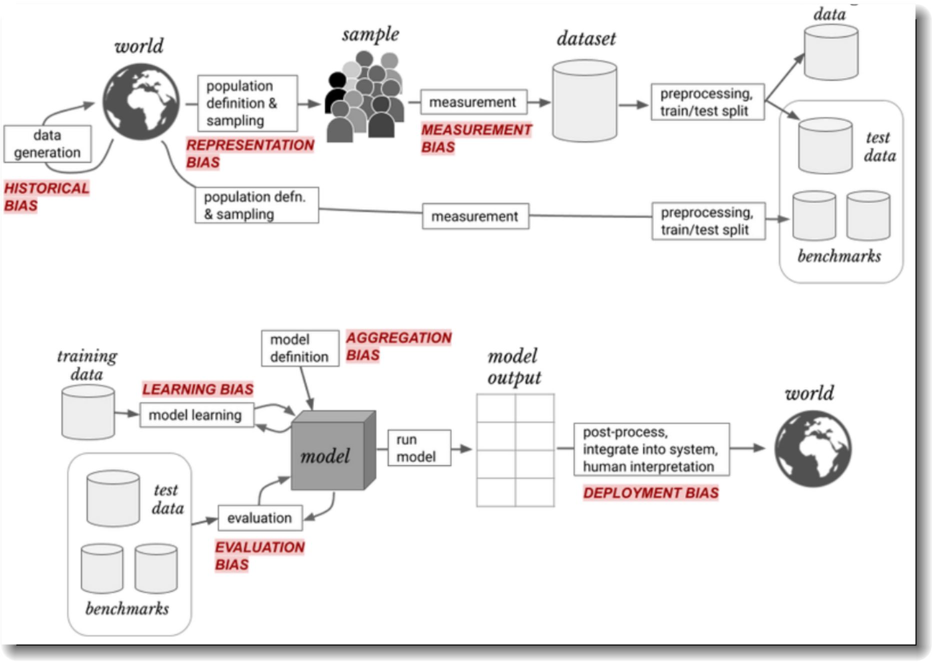


Figure 6.4: Types of Biases in ML framework [183]

popular in natural language processing, that are learned from a large corpus of text reflecting human biases.

2. ***Representation Bias.*** It occurs when some part of the population is under-represented in the development sample and that can subsequently fail while generalizing to the use population. ImageNet, a widely-used image dataset consisting of 1.2 million labelled images, lack geographical diversity and demonstrates a bias towards Western countries.
3. ***Measurement Bias.*** It happens from the way we choose, utilize, and measure a particular feature and label to use in a prediction problem. For example, in COMPAS, a recidivism risk prediction tool, prior arrests and close circle arrests were used as a proxy to measure the level of riskiness. This proxy can be viewed as differentially measured since the minority community is more highly controlled and policed.
4. ***Evaluation Bias.*** This occurs during a model evaluation when the benchmark data used for a particular task does not represent the use population and ultimately arises

because of a desire to quantitatively compare models against each other. As an example, commercial facial analysis tools benchmarked on Adience and IJB-A benchmark datasets were biased toward skin color and gender. It can also be exacerbated by the choice of metrics that are used to report performance. For example, aggregate measures can hide subgroups under performance.

5. ***Aggregation Bias.*** When one model is used for data in which there are underlying groups or types of examples that should be considered differently, aggregation bias happens. The assumption made while aggregation is that the mapping from inputs to labels is consistent across subsets of the data. When a dataset represents groups with different backgrounds, cultures or norms, a specific variable can mean different across them. This can lead to a model that fits well to the majority population or one that is not optimal for any group.

   Consider the following example of analyzing Twitter posts of a youth gang in Chicago and the shortcomings of general, non-context specific NLP tools were evident. As an example, specific hashtags and emojis used by this gang convey different meanings that a non-specific model trained on the general Twitter data would fail to identify. Also, some phrases that were predicted to be aggressive were actually lyrics from a local rapper. These findings tell us that ignoring the group-specific context in favour of a general model for all social media might lead to misclassifications of the tweets from this population.

6. ***Population Bias.*** When demographics, statistics, representatives, and user characteristics are different in the end user population represented in the dataset or platform from the original target population, population bias arises. As an example, consider the different use demographics on different social media platforms. Women are more likely to spend more time on Pinterest, Facebook, and Instagram while men are more active in online forums like Reddit and Twitter.

7. ***Simpson's Paradox.*** A trend, association or characteristic observed in underlying subgroups may be quite different from the association or characteristic observed when these subgroups are aggregated. Simpson's paradox can arise during the analysis of heterogeneous data that is composed of individuals with different behaviours or subgroups. As an example, we can consider the gender bias lawsuit in university admissions against UC Berkeley. Looking at the graduate school admissions data, it seemed like there was a bias against women. Only a smaller fraction of women candidates were admitted to different programs than their male counterparts. However, if we separate the data and analyze the departments, women had equality and in some cases had an advantage over men. One reason for the emergence of bias in this example is that women tended to apply to departments that had lower admission rates for both genders. In short,
8. ***Longitudinal Data Fallacy.*** Treating cross-sectional data from observational studies as if they were longitudinal can create biases due to Simpson's paradox. For example, upon analyzing bulk Reddit data, it has been observed that comment length decreased over time on average. When we look into it carefully, the data is representative of a cross-sectional view of the users who have joined the platform in different years. Once we disaggregated cohorts by considering the period they joined the platform, it was observed that the comment length increased over time within these cohorts.
9. ***Sampling Bias.*** It happens due to non-random sampling of subgroups. This leads to either systematic over-representation or under-representation of certain subgroups and the trends estimated for one population may not generalize well to another population.
10. ***Behavioral Bias.*** This happens due to different user behaviour across contexts, platforms or different datasets. As an example, it is observed that people react and behave differently on different platforms due to the differences in emoji representations among platforms, and this can even lead to communication errors.
11. ***Temporal Bias.*** It arises from the differences in populations and behaviours over time. For example, to capture attention people start using a particular hashtag about

a particular topic start but as the discussion gets continued, hashtags may not be used thereafter.

12. ***Popularity Bias.*** It arises as things that are more popular tend to be exposed more. In search engines and recommendation systems, popular objects would be presented to a wide public. It is also worth noting that popularity metrics are subject to manipulation such as fake reviews or social bots.
13. ***Algorithmic Bias.*** When the bias is not present in the input data, and is added purely by the algorithm, we term it as algorithmic bias. If the input data is indeed biased, the output of the algorithm might also reflect the bias. However, even if all the possible biases are detected, defining how an algorithm should proceed is difficult and it may require experts' help to detect to if the output has any bias at all.
14. ***User Interaction Bias.*** This can arise from two sources - the user interface and through the user itself and can be influenced by other types such as:

    - ***Presentation Bias.*** This arises from how information is presented. As an example, users click only the content that they can see and cannot see all the information on the Web.
    - ***Ranking Bias.*** Top-ranked results will result in more clicks than others and get more importance. This is common in search engines and crowd-sourcing applications.

15. ***Social Bias.*** When others' actions or content coming from other people affects our judgement, social bias arises. As an example, when we rate an object with a low score but are influenced by other people's high ratings, we may think that we are being too harsh and change our rating.
16. ***Emergent Bias.*** Due to changes in population, cultural values or societal knowledge in general, usually sometime after the completion of design, emergent bias arises as a result of use and interaction with real users. As an example, consider user interfaces in devices which tend to reflect the habits of users by design.

17. ***Self-Selection Bias.*** When the subjects of the research select themselves, self-selection bias happens. It is a subtype of sampling bias. Consider a survey about smart or successful students, some less successful students might think they are successful to be a part of the survey and this would bias the outcomes of the analysis.
18. ***Omitted Variable Bias.*** When one or more important variables are left out of the model, omitted variable bias occurs. As an example, let us consider a model developed to predict the annual number of customers who will stop subscribing to a particular service. If the company finds that a lot more people are unsubscribing from their service than what the model estimated, it might be the case that the model did not account for a new competitor in the market which offers the same service but at half the price of the company.
19. ***Cause-Effect Bias.*** As a result of the fallacy correlation implies causation, cause-effect bias arises. Let's consider the example of a data analyst trying to find out how successful their new program is. They observe that customers who are part of their new program are spending more money than the ones who are not and concluded that their new program is successful. It might be the case that only their committed customers who might have planned to spend more money have joined the new program in the first place.
20. ***Observer Bias.*** When a research investigator subconsciously projects their expectations into the project, observer bias could happen. Observer bias can happen when researchers cherry-pick participants or statistics that will favour their research and researchers unintentionally influence participants during surveys and interviews.

Given all possible biases that can occur in an ML pipeline, we state some group fairness notions that quantify the bias present in the prediction of ML models.

6.2.1 Group Fairness Notions

In *Group Fairness* (GF), we aim to build a classifier that is fair for protected attributes. For instance, a classifier should not be partial/unfair on the basis of one's sex or race or religion, etc. An example could be college admissions, a candidate should not be rejected just because of their gender. In GF, we aim to quantify fairness in models with respect to such protected attributes. Let A be the set of protected attributes, Y be the labels for the classifier, and $\hat{Y}$ be the predictions made by the classifier. We consider that the sensitive attribute is binary where a represents the samples belonging to one group and $1-a$ belongs to the other group.

Independence

Definition 6.2 (*Independence*)**.** *We say that the random variables $(A, Y, \hat{Y})$ satisfy Independence if the sensitive attributes A are statistically independent of the prediction $\hat{Y}$, denoted as $\hat{Y} \perp\!\!\!\perp A$.*

Independence is known by many names such as "Group Fairness", "Disparity Parity", "Statistical Parity", "Demographic Parity" and is mathematically defined as:

$$P(\hat{Y} = 1 \mid A = a) = P(\hat{Y} = 1 \mid A = 1 - a) \tag{6.2}$$

Let $x = P(\hat{Y} = 1 \mid A = a)$ and $y = P(\hat{Y} = 1 \mid A = 1 - a)$. Disparate Impact is then given as:

$$min\left(\frac{x}{y}, \frac{y}{x}\right) \geq 1 - \epsilon \tag{6.3}$$

Separation

Definition 6.3 (*Separation*)**.** *We say that the random variables $(A, Y, \hat{Y})$ satisfy Separation if the predictions of the model $\hat{Y}$ are statistically independent of A given true labels, Y, denoted as $\hat{Y} \perp\!\!\!\perp A|Y$.*

Separation is also called *Equalized Odds* and is satisfied when (A) True Positive Rates (TPR) and (B) False Positive Rates (FPR) across the sensitive attributes are equal. We mathematically state (A) as:

$$P(\hat{Y} \mid A = a, Y = 1) = P(\hat{Y} \mid A = 1 - a, Y = 1) \tag{6.4}$$

We also call this value as $\text{TPR}_0 = \text{TPR}_1$. (B) is stated as:

$$P(\hat{Y} \mid A = a, Y = 0) = P(\hat{Y} \mid A = 1 - a, Y = 0) \tag{6.5}$$

This is also called $\text{FPR}_0 = \text{FPR}_1$.

Sufficiency

Definition 6.4 (*Sufficiency*)**.** *We say that the random variables (A, Y, $\hat{Y}$) satisfy Sufficiency if the true labels Y are statistically independent of A given model predictions, $\hat{Y}$, denoted as $Y \perp\!\!\!\perp A | \hat{Y}$.*

Sufficiency is also known as *Calibration by group* and is satisfied when *Positive Predictive Value (PPV)* and *Negative Predictive Value (NPV)* across the sensitive attributes are equal. We mathematically state PPV as:

$$P(Y \mid A = a, \hat{Y} = 1) = P(Y \mid A = 1 - a, \hat{Y} = 1) \tag{6.6}$$

This is also known as $\text{PPV}_0 = \text{PPV}_1$. NPV is stated as:

$$P(Y \mid A = a, \hat{Y} = 0) = P(Y \mid A = 1 - a, \hat{Y} = 0) \tag{6.7}$$

This is also called as $\text{NPV}_0 = \text{NPV}_1$.

Calibration by group is defined as below,

$$P(Y = 1 \mid h(x) = r) = r \quad \forall a \in A \tag{6.8}$$

Calibration by group $\implies$ Sufficiency. Vice-versa is not true, for instance, when all labels are 0.

If $h(x)$ satisfies Sufficiency, then there exists a function $\ell = [0,1] \to [0,1]$ such that $\ell(h(x))$ satisfies calibration by groups. Let, $h(x) = s$ and $\ell(s) = P(Y = 1 \mid h(x) = s, A = a)$. Since $h(x)$ satisfies Sufficiency, this probability is the same for all groups a and hence this map ℓ is the same regardless of what value a we chose. We have,

$$\begin{aligned} s &= P(Y = 1 \mid \ell(h(x)) = s, A = a) \\ &= P(Y = 1 \mid h(x) \in l^{-1}(s), A = a) \\ &= P(Y = 1 \mid h(x) \in l^{-1}(s), A = 1 - a) \\ &= P(Y = 1 \mid \ell(h(x)) = s, A = 1 - a) \end{aligned}$$

6.2.2 Properties of Group Fairness Notions

Property 6.1. *Independence and Sufficiency cannot hold if A and Y are not independent.*

In general, Independence and Sufficiency are mutually exclusive. The only assumption, in this case, is that the A and Y are not independent. Consider, Sufficiency and Independence hold,

$$\begin{aligned} &\hat{Y} \perp\!\!\!\perp A,\ Y \perp\!\!\!\perp A|\hat{Y} \\ &\implies A \perp\!\!\!\perp (\hat{Y}, Y) \\ &\implies A \perp\!\!\!\perp Y \end{aligned}$$

This proves that for both Independence and Sufficiency to hold, A and Y have to be independent. In other words, Independence and Sufficiency cannot hold if A and Y are not independent.

Property 6.2. *If Y is binary, and A is not independent of Y, and $\hat{Y}$ is not independent of Y, then, Independence and Separation cannot both hold.*

Proof. To prove the proposition, it is enough to show that the contra-positive form holds true. I.e., the presence of both Independence and Separation indicates that either A is independent of Y or $\hat{Y}$ is independent of Y.

$$\hat{Y} \perp\!\!\!\perp A \quad \text{and} \quad \hat{Y} \perp\!\!\!\perp A|Y \implies Y \perp\!\!\!\perp A \quad \text{or} \quad Y \perp\!\!\!\perp \hat{Y}$$

From the law of total probability

$$P(A) = \sum_n P(A|B_n) \times P(B_n)$$

$$P(\hat{Y} = \hat{y} \mid A = a) = \sum_y P(\hat{Y} = \hat{Y} \mid A = a, Y = y) \times P(Y = y \mid A = a)$$

Applying, $\hat{Y} \perp\!\!\!\perp A$ and $\hat{Y} \perp\!\!\!\perp A|Y$,

$$P(\hat{Y} = \hat{y}) = \sum_y P(\hat{Y} = \hat{y} \mid Y = y) \times P(Y = y \mid A = a)$$

Also,

$$P(\hat{Y} = \hat{y}) = \sum_y P(\hat{Y} = \hat{y}) \times P(Y = y)$$

$$\implies \sum_y P(\hat{Y} = \hat{y} \mid Y = y) \times P(Y = y \mid A = a) = \sum_y P(\hat{Y} = \hat{y}) \times P(Y = y) \tag{6.9}$$

If,

$$p = P(Y = 0)$$

$$p_a = P(Y = 0 \mid A = a)$$

$$\hat{p_y} = P(\hat{Y} = \hat{y} \mid Y = y)$$

Equation 6.9, reduces to -

$$p_a \times \hat{p_0} + (1 - p_a) \times \hat{p_1} = p \times \hat{p_0} + (1 - p) \times \hat{p_1}$$

$$\implies p(\hat{p_0} - \hat{p_1}) = p_a(\hat{p_0} - \hat{p_1})$$

This equation can only be satisfied, when either $\hat{p_0} = \hat{p_1}$, in which case, $\hat{Y} \perp\!\!\!\perp Y$, or if $p = p_a$, in which case, $Y \perp\!\!\!\perp A$. □

Property 6.3. *If Y is binary and A is not independent of Y, then Separation and Sufficiency cannot both hold true.*

Proof. Since Y is not independent of A, then

$$P(Y = 1 \mid A = 0) \neq P(Y = 1 \mid A = 1)$$

Let's suppose that Separation holds. Since the classifier is not perfect, this means that all groups must have the same non-zero FPR ($FPR > 0$), and the same TPR ($TPR \geq 0$). In the binary case, Sufficiency implies that all groups have the same PPV (Equation 6.6).

$$PPV_a = \frac{TPR \times p_a}{TPR \times p_a + FPR \times (1 - p_a)} \quad \text{and} \quad PPV_b = \frac{TPR \times p_b}{TPR \times p_b + FPR \times (1 - p_b)}$$

From the two equations, it's clear that for $PPV_a = PPV_b$, either $TPR = 0$, or $FPR = 0$. Since, $FPR > 0$, it must be the case that $TPR = 0$. For Sufficiency to hold, NPV must also be the same. Consider,

$$NPV_a = \frac{(1 - FPR) \times (1 - p_a)}{(1 - TPR) \times p_a + (1 - FPR) \times (1 - p_a)}$$

and,

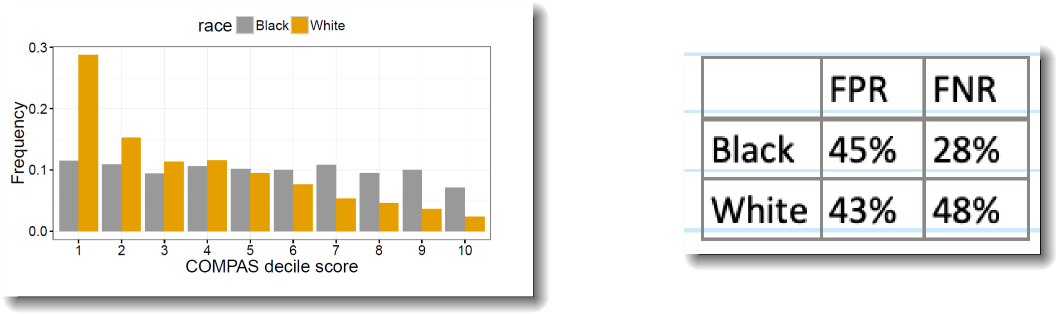
$$NPV_b = \frac{(1 - FPR) \times (1 - p_b)}{(1 - TPR) \times p_a + (1 - FPR) \times (1 - p_a)}$$

Since, $TPR = 0$

$$NPV_a = NPV_b$$
$$\implies \frac{1 - p_a}{p_a + (1 - FPR) \times (1 - p_a)} = \frac{1 - p_b}{p_b + (1 - FPR) \times (1 - p_b)}$$
$$\implies NPV_a \neq NPV_b$$

Since NPV is not equal, Sufficiency does not hold. □

Theorem 6.1 (Chouldechova 2017 [53])**.** *It is impossible to build a classifier that satisfies Separation and Sufficiency unless (1) base rates are equal or (2) the model is a perfect classifier.*



	FPR	FNR
Black	45%	28%
White	43%	48%

Figure 6.5: Violation of Demographic Parity (left) and Equalized Odds (right) [53]

For e.g., the COMPAS system satisfies calibration but FNR is not same for White and Blacks i.e., violates equalized odds and it also violates demographic parity (Figure 6.5).

Proof. Let $p_a = P(Y = 1|A = a)$ and $p_b = P(Y = 1|A = 1 - a)$. Suppose we have h that satisfies Sufficiency and Separation. To satisfy Separation, the two groups must have the same TPR and FRP. TPR is the rate at which the classifier recognizes the positive instances correctly. FPR is the rate at which the classifier mistakenly assigns positive outcomes to actual negative outcomes.

Equalizing TPR:

$$TPR_1 = TPR_0 \tag{6.10}$$

$$P(\hat{Y} = 1 \mid Y = 1, A = 1) = P(\hat{Y} = 1 \mid Y = 1, A = 0)$$

Equalizing FPR:

$$FPR_1 = FPR_0 \tag{6.11}$$

$$P(\hat{Y} = 1 \mid Y = 0, A = 1) = P(\hat{Y} = 1 \mid Y = 0, A = 0)$$

$$PPV_a = \frac{TPR_a \times p_a}{TPR_a \times p_a + FPR_a \times (1 - p_a)} = \frac{TPP_a}{TPP_a \times FPP_a}$$

Similarly,

$$PPV_b = \frac{TPR_b \times p_b}{TPR_b \times p_b + FPR_b \times (1 - p_b)} = \frac{TPP_b}{TPP_b \times FPP_b}$$

Since Separation is satisfied, let us assume the case where Sufficiency is also satisfied. If Sufficiency were to be satisfied, the PPV for the two groups must be the same.

Equalizing PPV,

$$
\begin{aligned}
&PPV_a = PPV_b \\
&\implies \frac{p_a}{TPR \times p_a + FPR \times (1 - p_b)} = \frac{p_b}{TPR \times p_b + FPR \times (1 - p_b)} \\
&\implies \frac{1}{y + x \times \dfrac{1 - p_a}{p_a}} = \frac{1}{y + x \times \dfrac{1 - p_b}{p_b}} \\
&\implies y + x \times \frac{1 - p_a}{p_a} = y + x \times \frac{1 - p_b}{p_b} \\
&\implies x \times (\frac{1 - p_a}{p_a}) = x \times (\frac{1 - p_b}{p_b})
\end{aligned}
$$

This indicates that either the base rate for the two groups is same or the classifier is perfect. Since we don't have a perfect classifier, the bast rate must be the same. This completes the proof that if we build a classifier that satisfies Separation and Sufficiency, either of the two conditions i) base rates are equal or ii) it is a perfect classifier, must be satisfied. □

Given the impossibility results, it is common to design classifiers that satisfy approximate fairness while ensuring high accuracy. In general, the approaches to ensure fairness can be broadly divided into three categories of i) *Pre-processing*, ii) *In-processing* and iii) *Post-processing* approaches. Pre-processing approaches are applied to the input data before the ML model is applied. In-processing approaches are applied within the ML model itself to make the predictions fair irrespective of the input. Post-processing approaches are applied to the model predictions obtained from the ML model. We elaborate on each of these further and discuss some existing approaches.

6.2.3 Pre-processing Approaches

The major source of bias in predictive models is the bias in the input data itself. Many fairness notions like Demographic Parity and Disparate Impact require that the positive outcome is offered equally across different sensitive sub-groups. Although the ground truth label itself may not be equally positive for both the sub-groups or in other words, the base rates are different. In such cases, there are pre-processing approaches which try to equalize the base rates. We discuss two of these approaches, (i) *Massaging Data* [119] and (ii) *Preferential Sampling* [120] below,

Massaging Data. Kamiran et al. 2009 [119] propose this approach to flip the ground truth of the data in a systematic way. For massaging, a ranker is required, a ranker is a classifier that is trained on X (not including sensitive attributes) to predict Y. The authors use the Naive Bayes classifier which provides the class probability i.e., the probability that a sample x belongs to class $y = 1$. In the method, there are two classes of samples identifies, 1) Candidates for promotion (CP) are the samples in the minority sub-group which have negative ground truth or $y = 0$, and 2) Candidates for demotion (CD) are the samples in the majority sub-group which have positive labels. The goal is to equalize the base rates by flipping the labels of samples in CP and CD. In other words, offer positive outcomes to some samples in the minority (CP) and offer negative labels to some samples in CD. The ranker is used to select the best sample whose label must be flipped. Rank the samples of CP in decreasing order of probability of belonging to the positive class and rank the samples of CD in the reverse way. Select the top M from both the ranked lists where M is,

$$M = \frac{(N^0 \times N^{11}) - (N^1 \times N^{01})}{N}$$

where N is the total number of samples, N^0 is the total number of samples belonging to sensitive attribute, $a = 0$ and N^{01} is the number of samples belonging to $a = 0$ having positive label $y = 1$. Similarly, N^1 and N^{11} are defined for $a = 1$. An illustrative example is given in Table 6.1

X	A	Y
x_1	M	$+$
x_2	M	$-$
x_3	M	$-$
x_4	M	$+$
x_5	M	$-$
x_6	F	$+$
x_7	F	$+$
x_8	F	$+$
x_9	F	$+$
x_{10}	F	$-$

X	A	Y	$\mathbb{P}_+$
x_1	M	$+$	62%
x_2	M	$-$	6%
x_3	M	$-$	49%
x_4	M	$+$	67%
x_5	M	$-$	3%
x_6	F	$+$	76%
x_7	F	$+$	89%
x_8	F	$+$	90%
x_9	F	$+$	60%
x_{10}	F	$-$	10%

X	A	Y
x_1	M	$+$
x_2	M	$-$
x_3	M	$+$
x_4	M	$+$
x_5	M	$-$
x_6	F	$+$
x_7	F	$+$
x_8	F	$+$
x_9	F	$-$
x_{10}	F	$-$

Table 6.1: Massaging Data ($\mathbb{P}_+$: probability of belonging to positive class)

Preferential Sampling. Since flipping of labels is too intrusive, Kamiran et al. 2010 [120] propose preferential sampling. The idea is to give importance to certain samples that are more prone to discrimination or favouritism. Such samples often lie at the borderline where the borderline objects are identified using a ranker. We first identify four groups of samples, i) DP is the samples belonging to the sensitive subgroup $a = 1$ with positive outcome $y = 1$, ii) DN samples belonging to the sensitive subgroup with negative outcome $y = 0$. iii) FP is the samples not belonging to the sensitive subgroup $a = 0$ with a positive outcome, and iv) FN is the samples not belonging to the sensitive sub group with a negative outcome. Use a ranker to rank the samples in DP and FP in ascending order and the samples in DN and FN in descending order; both w.r.t the positive class probability. The illustration is provided in Figure 6.6. Increasing the sample size is done by duplicating samples closest to the borderline, once duplicated the sample is moved to the bottom of the ranking and the process is repeated.

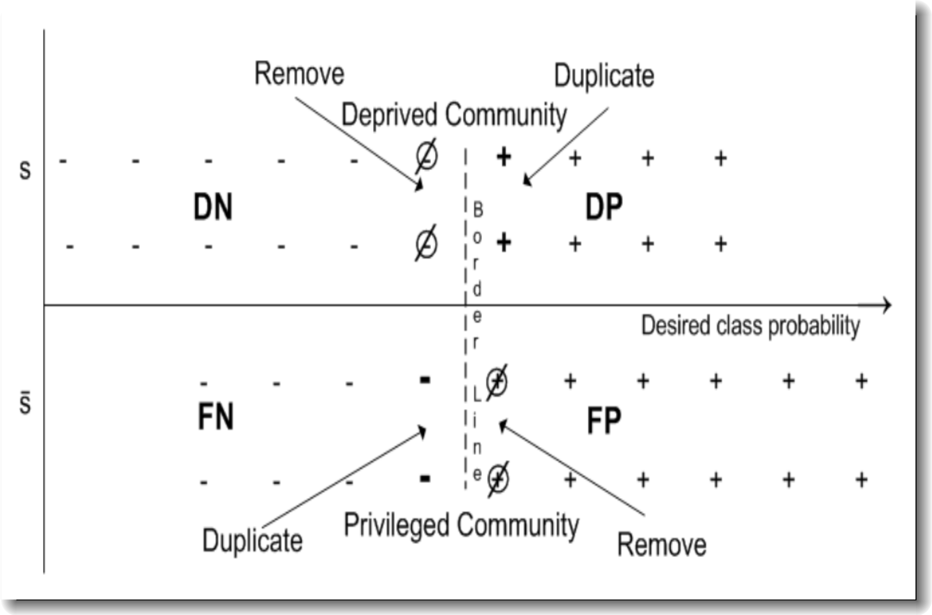


Figure 6.6: Preferential Sampling [120]

We now move on to some advanced approaches for making the input data bias free before applying the classifier. Since we want our classifiers to be independent of the sensitive attributes, hence the goal is to remove the sensitive attribute information from the input. Broadly these approaches look for *transformation* of the input to make it fair.

Removing Disparate Impact. A classifier is said to have a disparate impact (DI) if $\frac{P(Y=1|A=0)}{P(Y=1|A=1)} \leq 0.8$. Feldman et al. 2015 [77] provide a DI certificate i.e., a classifier exhibiting DI must have a low *Balanced Error Rate* (BER) or more leakage of protected attribute information into the predictions. Formally, BER is the error made a classifier that predicts sensitive attributes A from the input features X. Higher BER is more suitable for building a fair classifier. Given a dataset (X, A, Y), A is said to be ϵ-predictable from X if there exists a classifier $f_A : X \to A$ such that $\text{BER}(f_A(X), A) \leq \epsilon$. The final theorem which provides the DI certificate is stated as,

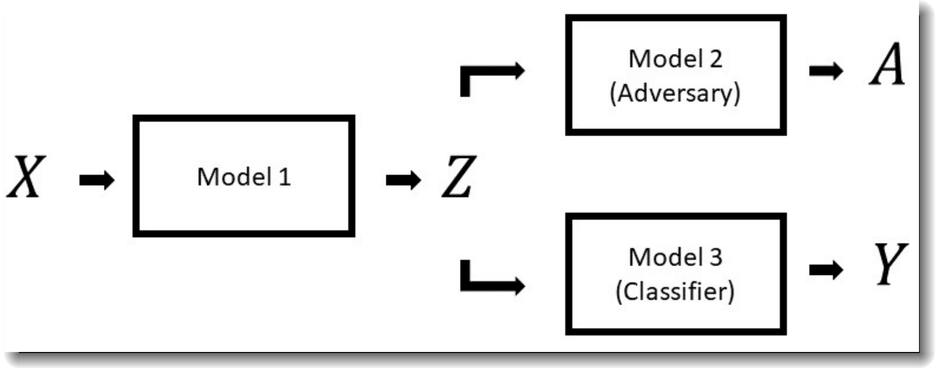


Figure 6.7: Adversarially Learning Fair Representations

Theorem 6.2 (Feldman et al. 2015 [77])**.** *A dataset is* $(1/2 - \beta/8)$*-predictable if and only if it admits disparate impact, where* β *is the fraction of elements in the minority class (A=0) that are selected* $(Y = 1)$*.*

The proof of the above theorem can be found in Feldman et al. 2015 [77], which serves as a good starting point in providing fairness guarantees for a dataset irrespective of the model used for classification. The authors also show how to remove DI from a dataset by transforming the input X. They propose *Combinatorial Repair* [77, Definition 5.1] and *Geometric repair* [77, Definition 5.2] algorithms which transforms the input to i) decrease the predictability of the sensitive attribute A, and ii) preserve the information of X as much as possible.

Adversarial Training. The above approach of removing DI is restrictive to only DI and also assumes certain properties on input features X. Beutel et al. 2017 [31] and Madras et al. 2018 [132] propose a more generic way of removing the predictability of sensitive attributes from X using adversarial training. In [132] the authors assume a model (Figure 6.7), which learns the data representation Z. Z is learnt such that it is capable of reconstructing X, classifying target labels Y and protecting the sensitive attribute A from an adversary. Let W_1, W_2, W_3 denote the parameters of Model 1, 2, 3 as depicted in Figure 6.7. The adversary seeks to maximize the objective L_a which is the loss of predicting A

from Z. L_r is the reconstruction loss between X and Z and L_c is the loss of predicting Y from Z. The overall objective is given by,

$$\min_{W_1,W_3} \max_{W_2} \ \alpha L_r(X, W_1, Z) + \beta L_a(Z, W_2, A) + \gamma L_c(Z, W_3, Y)$$

The network parameters are optimized using alternate gradient descent and ascent steps, further details in [132].

6.2.4 In-processing Approaches

As opposed to pre-processing approaches, the in-processing approaches focus on the training regime. The training of the classifier is such that it is fair irrespective of the bias in the input data. Such approaches are often end-to-end training which learn fair features and fair classification simultaneously. Primarily, these approaches design a loss function for every fairness measure considered. The objective then consists of classification loss and fairness loss, which is minimized during training. The main challenge here is the fairness measures are *complex* and *non-convex*, hence difficult to directly incorporate into the ML framework. To overcome this issue, the following approaches are proposed,

Convexification. The goal here is to design a convex surrogate for the fairness measures. The convex surrogate is used as the loss function during training. The hypothesis is that minimizing the convex surrogate would minimize the violation in the corresponding fairness measure. Bilal Zafar et al., Kamishima et al. 2015, 2011 [36, 122] have proposed such approaches. In [36], the authors design a convex surrogate loss called *Decision Boundary Covariance* for disparate impact (DI).

A classifier is said to be free from DI iff,

$$\min\left\{\frac{\mathbb{P}(\hat{y}_i = 1|a_i = 0)}{\mathbb{P}(\hat{y}_i = 1|a_i = 1)}, \frac{\mathbb{P}(\hat{y}_i = 1|a_i = 1)}{\mathbb{P}(\hat{y}_i = 1|a_i = 0)}\right\} \geq 0.8$$

Since the above loss is complex and non-convex, the authors propose decision boundary covariance. Given a classifier with parameters θ and $d_\theta(x) = \theta^T x$ and μ_a is the mean of the

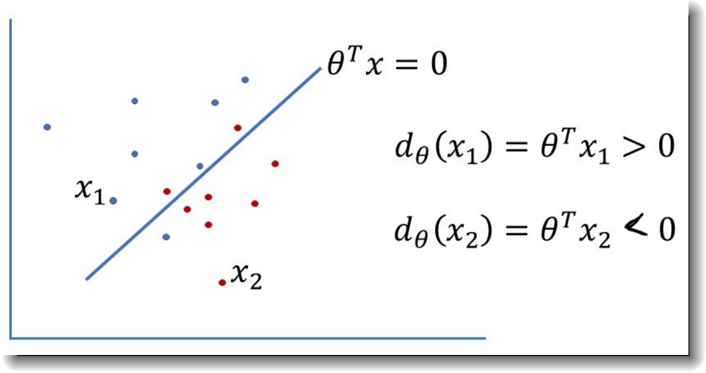

$\theta^T x = 0$
$d_\theta(x_1) = \theta^T x_1 > 0$
x_1
$d_\theta(x_2) = \theta^T x_2 < 0$
x_2

Figure 6.8: Decision Boundary Covariance

sensitive attribute (Figure 6.8). The idea is that the $d_\theta(x) > 0$ for $\hat{y} = 1$ therefore, for a classifier that is free from DI, $\mathbb{P}(d_\theta(x) > 0|a_i = 0) = \mathbb{P}(d_\theta(x_i) > 0|a_i = 1)$ or the covariance between $d_\theta(x)$ and a is 0. The decision boundary covariance is given by,

$$Cov(d_\theta(x), a) = \frac{1}{n}\sum_{i=1}^{n}(a - \mu_a)\theta^T x \tag{6.12}$$

Using the above loss, the authors formulate the following Logistic Regression,

$$\begin{aligned} \min_\theta \quad & -\frac{1}{n}\log(p(y_i = 1|x_i, \theta)) \\ s.t. \; & \frac{1}{n}\sum_{i=1}^{n}(a - \mu_a)\theta^T x \leq c \\ & \frac{1}{n}\sum_{i=1}^{n}(a - \mu_a)\theta^T x \geq -c \end{aligned}$$

The above minimizes classification loss under surrogate fairness constraints. Similarly, the authors also provide the formulation for minimizing violation in the surrogate constraint for accuracy constraints. The authors show that the above formulation leads to fine-grained control on the degree of fairness at a small cost to accuracy. Despite the results, designing such convex surrogates is often challenging and unique to each fairness constraint.

Moreover, minimizing the convex surrogates does not always guarantee that the actual fairness violation is minimized. Hence there are other approaches proposed to learn with complex loss functions and constraints which we discuss below.

Reductionist Approaches. Agarwal et al., Narasimhan 2018, 2018 [5, 152] propose a reductionist approach to solve the constrained optimization problem, i.e., minimizing classification loss under fairness constraints. In the reductionist approach, the problem is reduced to a sequence of cost-sensitive learning tasks. Let us look at the approach proposed in [5].

The main idea is to learn a randomized classifier that provides the best fairness and accuracy trade-off. Given set of classifiers $\mathcal{H} = \{h_1, h_2, \ldots\}$, the probabilities over them is, $\Delta(\mathcal{H}) = \{p_1, p_2, \ldots\}$. Let the classification loss be denoted by $err(Q), Q \in \Delta(\mathcal{H})$. They re-write the Demographic Parity (Definition 6.2) and Equalized odds (Definition 6.3) as linear constraints $M\mu(h) \leq c$.

- *Demographic Parity*: $\mathbb{E}[h(X)|A = a] = \mathbb{E}[h(X)], \forall a$ represented as $\mu_a(h) = \mu_*(h)$

$$\begin{aligned} \mu_a(h) - \mu_*(h) &\leq 0 \\ -\mu_a(h) + \mu_*(h) &\leq 0 \end{aligned}$$

- *Equalized Odds*: $\mathbb{E}[h(X)|A = a, Y = y] = \mathbb{E}[h(X)|Y = y], \forall a$

$$\begin{aligned} \mu_{(a,y)}(h) - \mu_{(*,y)}(h) &\leq 0 \\ -\mu_{(a,y)}(h) + \mu_{(*,y)}(h) &\leq 0 \end{aligned}$$

Final constrained optimization is set up as follows,

$$\min_{Q \in \Delta} \; err(Q) \;\; \text{s.t.} \;\; M\mu(Q) \leq c$$

Final Empirical Loss, where λ is the *Lagrangian multiplier* and the constraint could take specific form for different fairness measures considered. Thus, can be represented as a cost-sensitive classification.

$$L(Q, \lambda) = e\hat{r}r(Q) + \lambda^T(M\hat{\mu}(Q) - \hat{c})$$

$$\min_{Q\in\Delta} \max_{\lambda} \; L(Q, \lambda)$$

The above optimization is solved using [5, Algorithm 1], for which the authors also provide certain convergence guarantees.

6.2.5 Post-processing Approaches

We have already looked at pre-processing and in-processing approaches, where fairness constraints are incorporated before or during training. Although in many applications, there are pre-trained models available. Post-processing approaches try to make the predictions of such trained models fair. These approaches often manipulate the output predictions or the probabilities returned by the models.

Kamiran et al. 2012 [121] change the output labels predicted by the classifier $\hat{y}$ to other labels $\tilde{y}$. The reject option classification method considers those samples S for which predictions are uncertain, i.e., $|\hat{y}_S - 0.5| < \epsilon$, for some margin ϵ, (given the classification threshold in 0.5). For samples within the reject option band defined above, $\tilde{y} = 1$ for the under-represented class $a = 0$ and $\tilde{y} = 0$ for the other class. For samples outside the margin, $\hat{y} = \tilde{y}$. The margin θ is set such that the requirement on Disparate Impact is met. In [105], the authors aim to achieve Equalized Odds through a post-processing approach. From Definition 6.3, we know that the false negative rates and false positive rates must be the same across different groups. Given a classifier with predictions $\hat{y}$, the authors solve an optimization to identify certain probabilities with which to flip the output of every sample. With these probabilities, the output is flipped to ensure equalized odds. This approach is further refined in [46, 165].

In all the above approaches, the output predictions are modified. These output predictions are often obtained after fixing a certain threshold on the output probabilities for e.g., given that $\mathbb{P}(\hat{y} = 1)$ is 0.5 or more then $\hat{y} = 1$ else $\hat{y} = 0$. In [118], the authors propose *Group Specific Threshold Adaptation for faiR classification* (GSTAR). Given a trained model, GSTAR approximates the probability distribution of the model predictions and confusion matrix to quantify accuracy and fairness trade-offs for different thresholds for different sub-groups.

6.3 Privacy Issues

ML models heavily depend on the data they have been trained on. The models often rely on consumer data which also includes private information. The goal is to provide highly efficient solutions while ensuring the confidentiality of the data [178, 64]. We first discuss some non-rigorous methods employed for preserving privacy and show how they are vulnerable.

- ***Re-identification in anonymized database.*** Anonymization of a database is generally not safe, since they're prone to *linkage attacks* as shown in Netflix challenge [153]. Here, anonymization refers to the process of removal of personally identifiable information. Linkage attack involves re-identifying individuals by matching records from other auxiliary sources of data. Knowledge of sensitive data (like medical records Ohm 2009 [156]) can have adverse consequences in the hands of an adversary

- ***Group Queries.*** Individual queries are trivially disallowed on a privacy-preserving database with sensitive information. In the case of group queries, where the user queries are forced to be over a large set of samples, *differencing attack* turns to be a problem. The differencing attack may provide information about the presence of a certain individual or some property about the individual. For example, consider a database containing medical records of people. A user requests two similar queries, one requesting a number of people with a disease except for a specific person, say A, and the other requesting the total number of people with the same disease. The outputs of these queries cumulatively reveal the disease status of A, which is a blatant breach of A's privacy. as depicted by the example. Determining whether such an attack is possible on a database is a computationally undecidable problem

- ***Summary Statistics.*** Differencing attack immediately implies that summary statistics are not privacy-preserving. Besides this, *database reconstruction attacks* even on census data which uses traditional statistics with aggregate values, one can identify data corresponding to a set of individuals [81].

The goal is to define a quantitative privacy measure for the possible privacy loss. Privacy should guarantee plausible deniability and no linkage attacks. It should guarantee that an adversary does not know more about an individual in the data set before and after analyzing the database.

6.3.1 Pure Differential Privacy

Formally, the adversary's posterior view of an individual should be equal to the prior view for complete privacy. More formally, given the set of datasets $\mathcal{X}$ and a query q, any mechanism maps the query to a set of possible outcomes $\mathcal{Y}$. The randomized mechanism is denoted by $M : \mathcal{X} \to \mathcal{Y}$. If two input datasets D, D' are close to each other or differ by a single entry $\parallel D - D' \parallel_1 \leq 1$, then a mechanism M preserves privacy if, $\forall Y \in \mathcal{Y}$, the ratio $\frac{p(M(D)=Y)}{p(M(D')=Y)}$ tends to 1. The notion of *Differential Privacy* (DP) is thus defined in the following way,

Definition 6.5 (*Differentially Private Mechanism*)**.** *A randomized mechanism M is said to be ϵ differentially private if $\forall S \in \mathcal{Y}$ and $\forall D, D'$ s.t. $\parallel D - D' \parallel_1 \leq 1$,*

$$p(M(D) \in S) \leq \exp(\epsilon) p(M(D') \in S)$$

When $\epsilon = 0$, we achieve the highest privacy, but the mechanism returns the same output for every input hence not useful. On the other hand, $\epsilon >> 1$ implies a heavy loss in privacy. Practically ϵ between 0.1 and 0.2 is considered ideal in many scenarios.

An example of ϵ-DP mechanism is the *Laplace Mechanism*. A Laplace mechanism outputs a noisy answer to any query with values in $\mathbb{R}^n$. Firstly, the PDF of the Laplace distribution is denoted by:

$$\mathcal{L}(\mu, b) = \frac{1}{2b} \cdot \exp\left(-\frac{|x-\mu|}{b}\right), \tag{6.13}$$

where μ is the mean and b the scaling factor. When compared to the Gaussian distribution, the Laplace mechanism has a sharper peak and heavier tail. Before defining the Laplace mechanism, we will first define sensitivity of the function $f(\cdot)$, which we wish to make private as follows.

Definition 6.6 (*Sensitivity*)**.** *Let $\mathcal{X} = \{D_1, \ldots, D_m\}$ be the universal set of all adjacent datasets, i.e., $|D - D'| \leq 1$ for $D, D' \in \mathcal{X}$. For $f : \mathcal{X} \to \mathbb{R}^k$, we define its sensitivity Δ_f as:*

$$\Delta_f = \max_{D,D'} ||f(D) - f(D')||. \tag{6.14}$$

When $k > 1$, the maximum usually corresponds to the l_1 or l_2 norm.

We are now ready to define the Laplace mechanism. Recall that $M(f, D)$ is our randomized mechanism which ensures that the input function $f(\cdot)$ becomes (ϵ, δ)-DP such that $\delta \geq 0$.

Theorem 6.3 (*Laplace Mechanism*)**.** *A mechanism $M(f, D, \epsilon)$ is said to be ϵ-DP if it adds noise drawn from $\mathcal{L}\left(0, \frac{\Delta_f}{\epsilon}\right)$ to the output of $f(D)$. Formally,*

$$M(f, D, \epsilon) = f(D) + \mathcal{L}\left(0, \frac{\Delta_f}{\epsilon}\right). \tag{6.15}$$

The proof of the theorem can be found in Wikipedia contributors 2021 [197].

Next, we look at the slightly weaker notion of ϵ-DP which is practical and has various applications.

6.3.2 Approximate Differential Privacy

Approximate differential privacy or (ϵ, δ)-differential privacy, is a relaxation of ϵ-differential privacy where the privacy guaranteed is satisfied with a probability δ. In other words, with ϵ-differential privacy, a mechanism gives the ϵ privacy guarantee all the time, with a probability 1. On the other hand, for (ϵ, δ)-differential privacy, with δ probability the privacy is guaranteed. More formally,

Definition 6.7 (*Differential Privacy* (DP) [72])**.** *For a set of databases $\mathcal{X}$ and the set of noisy outputs $\mathcal{Y}$, a randomized algorithm $M : \mathcal{X} \rightarrow \mathcal{Y}$ is said to be (ϵ, δ)-DP if $\forall D, D' \in \mathcal{X}$, s.t. $|D - D'| \leq 1$, and $\forall S \subseteq \mathcal{Y}$ the following holds,*

$$\Pr[M(D) \in S] \leq \exp(\epsilon) \Pr[M(D') \in S] + \delta. \tag{6.16}$$

Here, ϵ is called the privacy budget and δ as the privacy budget relaxation.

Properties of Differential Privacy

Our first property answers the question: when we run multiple algorithms, each of which have privacy guarantees on their own, what is the privacy guarantee on the union of their outputs? How do the privacy parameters degrade? The cumulative privacy guarantee, i.e., privacy guarantee over a different set of queries, is referred to as composition. The resultant guarantee across k such queries is given by the following property. For the property, consider an adversary $\mathcal{A}$ with a view over k queries as $V^b = (R, Y_{1,b}, \ldots, Y_{k,b})$. Here, Y is the queries output and R $\mathcal{A}$'s internal randomness with $b \in \{0, 1\}$ as a binary parameter.

Property 6.4 (*Composability*)**.** *The class of ϵ-differentially private mechanisms M satisfies $k\epsilon$-DP under k-fold adaptive composition for an adversary $\mathcal{A}$. We assume that each query's randomness is independent of the other.*

Proof. A view of the adversary is the tuple $v = (r, y_1, \ldots, y_k)$. That is,

$$\begin{aligned}\frac{\Pr[V = v]}{\Pr[V' = v]} &= \left(\frac{\Pr[R = r]}{\Pr[R' = r]}\right) \cdot \prod_{i=1}^{i=k} \frac{\Pr[V_i = v_i | V_1 = v_1, \ldots, V_{i-1} = v_{i-1}]}{\Pr[V'_i = v_i | V'_1 = v_1, \ldots, V'_{i-1} = v_{i-1}]} \\ &\leq \prod_{i=1}^{i=k} \exp(\epsilon) \qquad \text{(since } M \in M \text{ is } (\epsilon, \delta)\text{-DP)} \\ &= \exp(k\epsilon).\end{aligned}$$

□

Next, we consider the case when the distance between the adjacent databases is greater than one row. In particular, consider databases differing in c rows. This amounts to the fact that an adversary with arbitrary auxiliary information can know if c particular participants submitted their information. We capture the privacy guarantee for such a case with the next property.

Property 6.5 (*Group Privacy*)**.** *For a set of databases $\mathcal{X}$ and the set of noisy outputs $\mathcal{Y}$, a randomized algorithm $M : \mathcal{X} \to \mathcal{Y}$ is said to be (ϵ, δ)-LDP if $\forall D, D' \in \mathcal{X}$, s.t. $|D - D'| \leq c$, and $\forall S \subseteq \mathcal{Y}$ the following holds,*

$$\Pr[M(D) \in S] \leq \exp(\epsilon \cdot c) \Pr[M(D') \in S] + \delta. \tag{6.17}$$

The proof for this follows directly from Definition 6.7. Crucially, the property highlights that the strength of the privacy guarantee drops linearly with the size of the group.

What is more, a differentially-private mechanism is also immune to post-processing. This implies that irrespective of any operation an adversary performs over the output of a DP mechanism, the privacy guarantees w.r.t. the indistinguishability of the databases does not change.

Property 6.6 (*Closure under Post-processing*)**.** *Let $M : \mathbb{Z}_+^R \to R$ be a randomized mechanism that satisfies (ϵ, δ)-DP. Let $f : \mathbb{R} \to R'$ be an arbitrary function. Then, $foM : \mathbb{Z}_+^R \to R'$ is also (ϵ, δ)-DP.*

Proof. Given two adjacent databases $|D - D'| \leq 1$ and the output space $S \subseteq R'$, consider the following mapping: $T = \{r \in S | f(r) \in S\}$. Now, we have,

$$\begin{aligned}
\Pr(foM(D) \in S) &= \Pr(M(D) \in T) \\
&\leq \exp(\epsilon) \cdot \Pr(M(D') \in T) \qquad \text{(since } M \text{ is } (\epsilon, \delta)\text{-DP)} \\
&\exp(\epsilon) \cdot \Pr(foM(D') \in S).
\end{aligned}$$

That is, foM is also (ϵ, δ)-DP. □

These properties are crucial for designing a DP mechanism. For e.g., consider generating a DP ML model. As these models are generally released for public use, post-processing property implies that the underlying privacy guarantees will not change irrespective of how one used the ML model. We saw that a Laplace mechanism satisfies, ϵ-DP, if we replace Laplace noise with Gaussian noise, it is shown to only satisfy (ϵ, δ)-DP. For instance, the Gaussian mechanism for a function $f : \mathcal{X} \to \mathbb{R}$ and sensitivity S_f

$$M(D) = f(D) + \mathcal{N}\left(0, S_f^2 \cdot \sigma^2\right)$$

where $\mathcal{N}\left(0, S_f^2 \cdot \sigma^2\right)$ is the Gaussian distribution with mean 0 and standard deviation, $S_f\sigma$. The above mechanism is shown to be (ϵ, δ)-DP if $\delta > \frac{4}{5}\exp(-\sigma\epsilon)^2/2$ and $\epsilon < 1$ [72, Theorem 3.22].

Chapter 7

FNNC: Fair Neural Network Classifier

In classification models, fairness can be ensured by solving a constrained optimization problem. We focus on fairness constraints like Disparate Impact, Demographic Parity, and Equalized Odds, which are non-decomposable and non-convex. Researchers define convex surrogates of the constraints and then apply convex optimization frameworks to obtain fair classifiers. Surrogates serve as an upper bound to the actual constraints, and convexifying fairness constraints is challenging.

We propose a neural network-based framework, *FNNC*, to achieve fairness while maintaining high accuracy in classification. The above fairness constraints are included in the loss using Lagrangian multipliers. We prove bounds on generalization errors for the constrained losses which asymptotically go to zero. The network is optimized using two-step mini-batch stochastic gradient descent. Our experiments show that FNNC performs as well as the state-of-the-art, if not better. The experimental evidence supplements our theoretical guarantees. In summary, we have an automated solution to achieve fairness in classification, which is easily extendable to many fairness constraints.

7.1 Introduction

In recent years machine learning models have been popularized as prediction models to supplement the process of decision-making. Such models are used for criminal risk assessment, credit approvals, online advertisements. These machine learning models unknowingly introduce a societal bias through their predictions [25, 30, 53]. E.g., ProPublica conducted its study of the risk assessment tool, which was widely used by the judiciary system in the USA. ProPublica observed that the risk values for recidivism estimated for African-American defendants were on average higher than for Caucasian defendants. Since then, researchers started looking at fairness in machine learning, especially quantifying the notion of fairness and achieving it.

Broadly fairness measures are divided into two categories. *Individual fairness* [71], requires similar decision outcomes for two individuals belonging to two different groups concerning the sensitive feature and yet sharing similar non-sensitive features. The other notion is of *group fairness* [202], which requires different sensitive groups to receive beneficial outcomes in similar proportions. We are concerned with group fairness and specifically: *Demographic Parity* (DP) [71], *Disparate Impact* (DI) [77] and *Equalized odds* (EO) [105]. DP ensures that the fraction of the positive outcome is the same for all the groups. DI ensures the ratio of the fractions is above a threshold. However, both constraints fail when the base rate itself differs, hence EO is the more useful notion of fairness, which ensures an even distribution of false-positive rates and false-negative rates among the groups. All these definitions make sense only when the classifier is well-calibrated. That is, if a classifier predicts an instance belongs to a class with a probability of 0.8, then there should be 80% of samples belonging to that class. Chouldechova 2017 [53] and Pleiss et al. 2017 [165] show that it is impossible to achieve EO with calibration unless we have perfect classifiers. Hence, the major challenge is to devise an algorithm that guarantees the best predictive accuracy while satisfying the fairness constraints to a certain degree.

Towards designing such algorithms, one approach is pre-processing the data. The methods under this approach treat the classifier as a black box and focus on learning fair representations. The fair representations learned may not result in optimal accuracy. The other approach models achieving fairness as constrained optimization [36, 122, 199]. Wu et al. 2018 [199] have provided a generalized convex optimization framework with theoretical guarantees. The fairness constraints are upper-bounded by convex surrogate functions and then directly incorporated into classification models.

There are several limitations in the existing approaches which ensure fairness in classification models. Surrogate constraints may not be a reasonable estimate of the original fairness constraint. Besides, coming up with good surrogate losses for the different definitions of fairness is challenging. In this work, we study how to achieve fairness in classification. In doing so, we do not aim to propose a new fairness measure or new optimization technique. As opposed to the above approaches, we propose to use neural networks for implementing non-convex complex measures like DP, DI, or EO. The network serves as a simple classification model that achieves fairness. One need not define surrogates or do rigorous analysis to design the model. Mainly, it is adaptable to any definition of fairness.

Typically, one cannot evaluate fairness measures per sample as these measures make sense only when calculated across a batch, which contains data points from all the sensitive groups. Given that at every iteration, the network processes mini-batch of data, we can approximate the fairness measure given an appropriate batch size. Hence, we use *mini-batch stochastic gradient descent* (SGD) for optimizing the network. We empirically find that it is possible to train a network using the *Lagrangian Multiplier Method*, which ensures these constraints and achieves accuracy at par with the other complex frameworks. Likewise, it is also possible to incorporate other complex measures like F1-score, H-mean loss, and Q-mean loss,– not related to fairness. We have included an experiment on training a network to minimize Q-mean loss with DP as a constraint.

Our Contribution.

i) We propose to design a fair neural network classifier (FNNC) to achieve fairness in classification. ii) We provide generalization bounds for the different losses and fairness constraints DP and EO (Theorem 7.3) in FNNC. iii) We show that, in some instances, it may be difficult to approximate DI constraint by another surrogate DI constraint (Theorem 7.4). iv) We empirically show that FNNC can achieve the state-of-the-art performance, if not better.

7.2 Existing Approaches

Zemel et al. 2013 [202] discuss DP, EO, and DI are a few of its types. It is a major challenge to enforce these in any general machine-learning framework. We discuss the following approaches that deal with fair classification (elaborately discussed in Chapter 6)

Pre-Processing. The first body of work focuses on pre-processing i.e., coming up with fair representations as opposed to fair classification e.g., [71, 77, 119]. Neural networks have been extensively used in such pursuits. E.g., Louizos et al. 2015 [131] gives a method for learning fair representations with a variational auto-encoder by using maximum mean discrepancies between the two sensitive groups. In [31, 74, 132], the authors explore the notion of adversarially learning a classifier that achieves DP, EO or DI.

Surrogate Loss. The second approach focuses on analytically designing convex surrogates for fairness definitions. In [27, 45, 122], the authors introduce penalty functions to penalize unfairness. In [36, 199] the authors give a generalized convex framework that incorporates all possible surrogates and gives appropriate bounds. Zhang et al. 2018 [203] uses neural network-based adversarial learning, which attempts to predict the sensitive attribute based on the classifier output, to learn an equal opportunity classifier.

Reductionist Approaches. The third is the reductionist approach, in which the task of fair classification is reduced to a sequence of cost-sensitive classification [152], and [5] which can then be solved by a standard classifier. Agarwal et al. 2018 [5] allows for fairness definitions that can be characterized as linear inequalities under conditional moments like DP and EO (DI does not qualify for the same). FNNC does not have such restrictions and hence performs reasonably for DI as well. We are easily able to include complex and non-decomposable loss functions like Q-mean loss, whereas Agarwal et al. 2018 [5] aims to improve only the accuracy of the model.

7.3 Preliminaries

In this section, we introduce the notation used and state the definitions of the fairness measures and the performance measures that we have analyzed.

We consider a binary classification problem with no assumption on the instance space. X is our (d-dimensional) instance space s.t. $X \in \mathbb{R}^d$ and output space $Y \in \{0, 1\}$. We also have a protected attribute $\mathcal{A}$ associated with each individual instance, which for example could be age, sex or caste information. For each $a \in \mathcal{A}$, a could be a particular category of the sensitive attribute like male or female.

Definition 7.1 (*Demographic Parity (DP)*)**.** *A classifier h satisfies demographic parity under a distribution over $(X, \mathcal{A}, Y)$ if its predictions $h(X)$ is independent of the protected attribute A. That is, $\forall a \in \mathcal{A}$ and $p \in \{0, 1\}$*

$$\mathbf{P}[h(X) = p | \mathcal{A} = a] = \mathbf{P}[h(X) = p]$$

Given that $p \in \{0, 1\}$, we can say $\forall$ a

$$\mathbb{E}[h(X)|\mathcal{A} = a] = \mathbb{E}[h(X)]$$

Definition 7.2 (*Equalized Odds (EO)*)**.** *A classifier h satisfies equalized odds under a distribution over $(X, \mathcal{A}, Y)$ if its predictions $h(X)$ are independent of the protected attribute $\mathcal{A}$ given the label Y. That is, $\forall a \in \mathcal{A}$, $p \in \{0, 1\}$ and $y \in Y$*

$$\mathbf{P}[h(X) = p | \mathcal{A} = a, Y = y] = \mathbf{P}[h(X) = p | Y = y]$$

Given that $p \in \{0, 1\}$, *we can say* $\forall\ a, y$

$$\mathbb{E}[h(X)|\mathcal{A} = a, Y = y] = \mathbb{E}[h(X)|Y = y]$$

Definition 7.3 (*Disparate Impact (DI)*)**.** *The outcomes of a classifier* h *disproportionately hurt people with certain sensitive attributes. The following is the definition for completely removing DI,*

$$min\left(\frac{\mathbf{P}(h(x) > 0|a = 1)}{\mathbf{P}(h(x) > 0|a = 0)}, \frac{\mathbf{P}(h(x) > 0|a = 0)}{\mathbf{P}(h(x) > 0|a = 1)}\right) = 1$$

Pleiss et al. 2017 [165] strongly claims that the above mentioned measures are rendered useless if the classifier is not calibrated, in which case the probability estimate p of the classifier could carry different meanings for the different groups.

Definition 7.4 (*Calibration*)**.** *A classifier* h *is perfectly calibrated if* $\forall\ p \in [0, 1]$, $\mathbf{P}(y = 1|h(x) = p) = p$.

Given the definition, the authors prove the following impossibility of calibration with equalized odds.

Theorem 7.1 (*Impossibility Result* (Pleiss et al. 2017 [165]))**.** *Let* h_1, h_2 *be two classifiers for groups* a_1 *and* $a_2 \in \mathcal{A}$ *with* $\mathbf{P}(y = 1|a_1 = 1) \neq \mathbf{P}(y = 1|a_2 = 1)$. *Then* h_1 *and* h_2 *satisfy the equalized odds and calibration constraints if and only if* h_1 *and* h_2 *are perfect predictors.*

Given the above result, we cannot guarantee to ensure the fairness constraints perfectly, hence we relax the conditions while setting up our optimization problem as follows,

7.3.1 Problem Framework

We have used the cross-entropy loss or the Q-mean loss as our performance measures, defined in the next section. We denote this loss by $l(h_\theta(X), Y)$ parameterized by θ, the weights of the network. Our aim is to minimize the loss under the additional constraints of fairness. Below we state the ϵ-relaxed fairness constraints that we implement in our model. $\forall\ a, y$,

$$DP : |\mathbb{E}[h(X = x)|\mathcal{A} = a] - \mathbb{E}[h(X = x)]| \leq \epsilon \tag{7.1}$$

$$EO : |\mathbb{E}[h(X = x)|\mathcal{A} = a, Y = y] - \mathbb{E}[h(X = x)|Y = y]| \leq \epsilon \tag{7.2}$$

DI: It is not possible to completely remove DI but one has to ensure least possible DI specified by the $p\% - rule$,

$$min\left(\frac{\mathbf{P}(h(x) > 0|a = 1)}{\mathbf{P}(h(x) > 0|a = 0)}, \frac{\mathbf{P}(h(x) > 0|a = 0)}{\mathbf{P}(h(x) > 0|a = 1)}\right) \geq \frac{p}{100} \tag{7.3}$$

We have the following generic optimization framework. Both the loss and the constraints can be replaced according to the need,

$$\boxed{\begin{aligned} &\underset{\theta}{min}\ l_\theta \\ &s.t.\ Eq\ 7.1\ or\ 7.2\ or\ 7.3 \end{aligned} \qquad (7.4)}$$

7.4 Proposed Framework: FNNC

In this section, we discuss how we use the neural network for solving the optimization problem framework in Equation 7.4.

7.4.1 Network Architecture

Our network is a two-layered feed-forward neural network. We only consider binary classification in all our experiments, although this method and the corresponding definitions are easily extendable to multiple classes. Let $h_\theta(.)$ be the function parameterized by θ that the neural network learns. In the last layer of this network, we have a softmax function which gives the prediction probability p_i, where p_i is the predicted probability that the i^{th} data sample belongs to one class and $1 - p_i$ is the probability for that it belongs to the other. Hence $p := h_\theta(.)$. We use the output probabilities to define the loss and the fairness measure.

7.4.2 Loss Function and Optimizer

Given the constrained optimization defined by Equation 7.4, we use the Lagrangian Multiplier method to incorporate the constraints within a single overall loss. Since the constraints are non-convex, we can only guarantee that the optimizer converges to local minima. Nevertheless, our experiments show that the model has at par or better performance compared to the existing approaches. We now describe the different loss functions that we have used in the experiments.

Fairness Constraints with Cross Entropy Loss

The fairness constraint DP as in the Def. 7.1 is given by $\forall a \in \mathcal{A}$,

$$\begin{aligned} \mathbb{E}[h(X = x)|\mathcal{A} = a] &= \mathbb{E}[h(X = x)] \\ \mathbb{E}[h(X = x)|\mathcal{A} = 1 - a] &= \mathbb{E}[h(X = x)] \\ \therefore\ \mathbb{E}[h(X = x)|\mathcal{A} = a] &= \mathbb{E}[h(X = x)|\mathcal{A} = 1 - a] \end{aligned}$$

Hence the constraint for a fixed batch size S of samples given by $z_S = (h(x_S), a_S, y_S)$ and $p_i = h(x_i) \in [0, 1]$, can be defined as follows,

$$const^{DP}(z_S) = \left| \frac{\sum_{i=1}^{S} p_i a_i}{\sum_{i=1}^{S} a_i} - \frac{\sum_{i=1}^{S} p_i (1 - a_i)}{\sum_{i=1}^{S} 1 - a_i} \right|$$

For the next constraint EO, we first define the difference in false-positive rate between the two sensitive attributes,

$$fpr(z_S) = \left| \frac{\sum_{i=1}^{S} p_i (1 - y_i) a_i}{\sum_{i=1}^{S} a_i} - \frac{\sum_{i=1}^{S} p_i (1 - y_i)(1 - a_i)}{\sum_{i=1}^{S} 1 - a_i} \right|$$

The difference in false-negative rate between the two sensitive attributes,

$$fnr(z_S) = \left| \frac{\sum_{i=1}^{S}(1-p_i)y_i a_i}{\sum_{i=1}^{S} a_i} - \frac{\sum_{i=1}^{S}(1-p_i)y_i(1-a_i)}{\sum_{i=1}^{S} 1-a_i} \right|$$

Following a similar argument as before the empirical version of EO as defined by Equation 7.2 and also used by Madras et al. 2018 [132] in the experiments is,

$$const^{EO}(z_S) = fpr + fnr$$

EO as defined in [5] is,

$$const^{EO}(z_S) = \max\{fpr, fnr\}$$

Empirical version of DI for a batch of S samples as defined in Equation 7.3 as a constraint for binary classes is given by,

$$const^{DI}(z_S) = -min\left(\frac{\frac{\sum_{i=1}^{S} a_i p_i}{\sum_{i=1}^{S} a_i}}{\frac{\sum_{i=1}^{S}(1-a_i)p_i}{\sum_{i=1}^{S} 1-a_i}}, \frac{\frac{\sum_{i=1}^{S}(1-a_i)p_i}{\sum_{i=1}^{S} 1-a_i}}{\frac{\sum_{i=1}^{S} a_i p_i}{\sum_{i=1}^{S} a_i}} \right)$$

The tolerance for each constraint is given by ϵ, which gives the following inequality constraints, for $const^k, \quad \forall k \in \{DP, EO, DI\}$ the empirical loss for B batches of samples with each batch having S samples denoted by z_S,

$$l_k(h(X), \mathcal{A}, Y) : \frac{1}{B}\sum_{l=1}^{B} const^k(z_S^{(l)}) - \epsilon \leq 0 \tag{7.5}$$

Specifically, for $const^{DI}(z_S)$, $\epsilon = -\frac{p}{100}$, where p is typically set to 80.

For maximizing the prediction accuracy, we use cross-entropy loss which is defined as follows for each sample,

$$l_{CE}(h(x_i), y_i) = -y_i \log(p_i) - (1-y_i)\log(1-p_i)$$

The empirical loss,

$$\hat{l}_{CE}(h(X), Y) = \frac{1}{SB}\sum_{i=1}^{SB} l_{CE}(h(x_i), y_i)$$

Hence, the overall loss by the Lagrangian method is,

$$L_{NN}(h(X), \mathcal{A}, Y) = \hat{l}_{CE}(h(X), Y) + \lambda\ l_k(h(X), \mathcal{A}, Y) \tag{7.6}$$

Satisfying DP with Q-mean Loss

The loss due to DP as already defined before is given by Equation 7.5, when $k = DP$. The empirical version of Q-mean loss for binary classes that is for $\forall i \in \{0, 1\}$ is defined as follows,

$$\sqrt{\frac{1}{2}\sum_{i=0}^{1}\left(1 - \frac{\mathbf{P}(h(x) = i, y = i)}{\mathbf{P}(y = i)}\right)^2} \tag{7.7}$$

The corresponding constraint is given by,

$$l_Q(h(x_S), y_S) = \frac{1}{\sqrt{2}}\sqrt{\left(1 - \frac{\sum_{i=1}^{S} y_i p_i}{\sum_{i=1}^{S} y_i}\right)^2 + \left(1 - \frac{\sum_{i=1}^{S}(1 - y_i)(1 - p_i)}{\sum_{i=1}^{S}(1 - y_i)}\right)^2}$$

The empirical Q-mean loss is,

$$\hat{l}_Q(h(X), Y) = \frac{1}{B}\sum_{l=1}^{B} l_Q(h(x_S^{(l)}), y_S^{(l)})$$

Hence, the overall loss by the Lagrangian method is,

$$L_{NN}(h(X), \mathcal{A}, Y) = \hat{l}_Q(h(X),Y) + \lambda\, l_{DP}(h(X), \mathcal{A}, Y) \tag{7.8}$$

Lagrangian Multiplier Method

The combination of losses and constraints mentioned above are not exhaustive. The generic definition of the loss could thus be given by, $\forall k \in \{DP, EO, DI\}$

$$\boxed{L_{NN} = l_\theta + \lambda l_k \qquad (7.9)}$$

In the equation above, λ is the Lagrangian multiplier. Any combination can be tried by changing l_θ and l_k as defined in Equation 7.6 and Equation 7.8. The overall optimization of Equation 7.9 is as follows,

$$\underset{\theta}{min}\ \underset{\lambda}{max}\ L_{NN}$$

The above optimization is carried out by performing SGD twice, once for minimizing the loss w.r.t. θ and again for maximizing the loss w.r.t. λ at every iteration [73].

7.5 Theoretical Guarantees: Generalization Bounds

In this subsection, we provide uniform convergence bounds using Rademacher complexity [174] for the loss functions and the constraints discussed above. We assume the class of classifiers learned by the neural network has a finite capacity and we use covering numbers to get this capacity. Given the class of neural network, $\mathcal{H}$, for any $h, \hat{h} \in \mathcal{H},\ h : \mathbb{R}^d \to [0,1]$, we define the following l_∞ distance: $\max_x |h(x) - \hat{h}(x)|$. $\mathcal{N}_\infty(\mathcal{H}, \mu)$ is the minimum number of balls of radius μ required to cover $\mathcal{H}$ under the above distance for any $\mu > 0$.

Theorem 7.2. *For each of $k \in \{DP, EO\}$, the relation between the statistical estimate of the constraint given batches of samples, z_S, $\mathbb{E}_{z_S}[const^k(z_S)]$, and the empirical estimate for B batches of samples is listed below. Given that $const^k(z_S) \leq 1$, for a fixed $\delta \in (0,1)$ with a probability at least $1-\delta$ over a draw of B batches of samples from $(h(X), \mathcal{A}, Y)$, where $h \in \mathcal{H}$,*

$$\mathbb{E}\left[const^k(z_s)\right] \leq \frac{1}{B}\sum_{\ell=1}^{B} const^k\left(z_S^{(\ell)}\right) + 2\Omega_k + C\sqrt{\frac{\log(\frac{1}{\delta})}{B}}$$

$$\Omega_{DP,EO} = \inf_{\mu>0}\left\{\mu + \sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{H}, \mu/2S)\right)}{B}}\right\}$$

Similarly for cross entropy loss l_{CE} and Q- mean loss l_Q we get the following bounds.

CE loss: consider $h(x) = \phi(f(x))$ where ϕ is the softmax over the neural network output $f(x)$ where $f \in \mathcal{F}$, assuming $f(x) \leq L$

$$\mathbb{E}[l_{CE}(f(x), y)] \leq \frac{1}{B}\sum_{i=1}^{B} l_{CE}(f(x_i), y_i) + 2\Omega_L + CL\sqrt{\frac{\log(\frac{1}{\delta})}{B}}$$

$$\Omega_L = \inf_{\mu>0}\left\{\mu + L\sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{F}, \mu/S)\right)}{B}}\right\}$$

Q-mean loss:

$$\mathbb{E}\left[l_Q(h(x_S), y_S)\right] \leq \frac{1}{B}\sum_{\ell=1}^{B} l_Q\left(h(x_S^{(\ell)}), y_S^{\ell}\right) + 2\Omega_Q + C\sqrt{\frac{\log(\frac{1}{\delta})}{B}}$$

$$\Omega_Q = \inf_{\mu>0}\left\{\mu + \sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{H}, \mu/S)\right)}{B}}\right\}$$

C is the distribution independent constant.

Proof. We first state the following lemma to prove the above theorem

Lemma 7.1. *[26] Let $\mathcal{S} = \{z_1, \ldots, z_B\}$ be a sample of i.i.d. from some distribution D over Z. Then with probability at least $1-\delta$ over a draw of $\mathcal{S}$ from D, for all $f \in \mathcal{F}$,*

$$\mathbb{E}_{z\in D}[f(z)] \leq \frac{1}{B}\sum_{i=1}^{B} f(z_i) + 2\hat{\mathcal{R}}_B(\mathcal{F}) + 4c\sqrt{\frac{2\log(4/\delta)}{B}}$$

Given $\hat{\mathcal{R}}_B(\mathcal{F})$ the Rademacher complexity of $\mathcal{F}$, the class of neural network, $\mathcal{H}$, for any $h, \hat{h} \in \mathcal{H},\ h : \mathbb{R}^d \to [0,1]$, we define the following l_∞ distance:

$$\max_x |h(x) - \hat{h}(x)| \tag{7.10}$$

Bounds for DP

Given a batch of samples $z_S = (h(x_S), a_S, y_S)$ for a fixed batch size S and $h \in \mathcal{H}$

$$const^{DP}(z_S) = \left|\frac{\sum_{i=1}^{S} h(x_i)a_i}{\sum_{i=1}^{S} a_i} - \frac{\sum_{i=1}^{S} h(x_i)(1-a_i)}{\sum_{i=1}^{S} 1-a_i}\right|$$

Let us consider a class of $\mathcal{DP}$ functions defined on the class of $\mathcal{H}$ as follows,

$$\mathcal{DP} = \{const^{DP} : (h(X), \mathcal{A}, Y) \to \mathbb{R}|\ const^{DP}(z_S) = \left|\frac{\sum_{i=1}^{S} h(x_i)a_i}{\sum_{i=1}^{S} a_i} - \frac{\sum_{i=1}^{S} h(x_i)(1-a_i)}{\sum_{i=1}^{S} 1-a_i}\right| \text{ for some } h \in \mathcal{H}\}$$

Similar to $\mathcal{N}(\mathcal{H}, \mu)$ given by Equation 7.10 we define for $\mathcal{DP}$. Define the l_∞ distance as

$$\max_{z_S} |\ const^{DP}(z_S) - \widehat{const}^{DP}(z_S)|$$

$\mathcal{N}_\infty(\mathcal{DP}, \mu)$ is the minimum number of balls of radius μ required to cover $\mathcal{DP}$ under the above distance for any $\mu > 0$. We apply Lemma 7.1 to the class of demographic parity functions $\mathcal{DP}$. Given a fixed batch size S, we have for any z_S, $const^{DP}(z_s) \leq 1$. By definition of the covering number $\mathcal{N}_\infty(\mathcal{DP}, \mu)$ for any class $const^{DP} \in \mathcal{DP}$, there exists a $\widehat{const}^{DP} \in \hat{\mathcal{DP}}$ where $|\hat{\mathcal{DP}}| \leq \mathcal{N}_\infty(\mathcal{DP}, \mu)$ such that $\max_{z_S} |const^{DP}(z_s) - \widehat{const}^{DP}| \leq \mu$, for a given $\mu \in (0, 1)$ Given B batches of samples where batch is of fixed size S

$$\begin{aligned}
\hat{\mathcal{R}}_B(\mathcal{DP}) &= \frac{1}{B}\mathbb{E}_\sigma\left[\sup_{const^{DP}} \sum_{\ell=1}^{B} \sigma_\ell \cdot const^{DP}(z_S^{(\ell)})\right] \\
&= \frac{1}{B}\mathbb{E}_\sigma\left[\sup_{const^{DP}} \sum_{\ell=1}^{B} \sigma_\ell \cdot \widehat{const}^{DP}\left(z_S^{(\ell)}\right)\right] \\
&+ \frac{1}{B}\mathbb{E}_\sigma\left[\sup_{const^{DP}} \sum_{\ell=1}^{B} \sigma_\ell \cdot const^{DP}\left(z_S^{(\ell)}\right) - \widehat{const}^{DP}\left(z_S^{(\ell)}\right)\right] \\
&\leq \frac{1}{B}\mathbb{E}_\sigma\left[\sup_{\widehat{const}^{DP}} \sum_{\ell=1}^{B} \sigma_\ell \cdot \widehat{const}^{DP}\left(z_S^{(\ell)}\right)\right] + \frac{1}{B}\mathbb{E}_\sigma \parallel \sigma \parallel_1 \mu \\
&\leq \sqrt{\sum_\ell \left(\widehat{const}^{DP}\left(z_S^\ell\right)\right)^2}\sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{DP}, \mu)\right)}{B^2}} + \mu \qquad \text{(By Massart's Lemma)} \\
&\leq \sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{DP}, \mu)\right)}{B}} + \mu
\end{aligned} \tag{7.11}$$

The last inequality is because,

$$\sqrt{\sum_\ell \left(\widehat{const}^{DP}\left(z_S^\ell\right)\right)^2} \leq \sqrt{\sum_\ell \left(const^{DP}\left(z_S^\ell\right) + \mu\right)^2} \leq \sqrt{B}$$

Lemma 7.2. $\mathcal{N}_\infty(\mathcal{DP}, \mu) \leq \mathcal{N}_\infty(\mathcal{H}, \mu/S)$

Proof. For any $h, \hat{h} \in \mathcal{H}$ such that for all x we get

$$|h(x) - \hat{h}(x)| \leq \mu/S \tag{7.12}$$

We know that $h(x) \in [0,1]$ $\forall h \in \mathcal{H}$. Now let us consider for the class of $\mathcal{DP}$,

$$\begin{aligned}
|const^{DP} - \widehat{const}^{DP}| &= \left| \left| \sum_{i=1}^{S} h(x_i) \left\{ \frac{a_i}{\sum_i a_i} - \frac{1-a_i}{\sum_i 1-a_i} \right\} \right| - \left| \sum_{i=1}^{S} \hat{h}(x_i) \left\{ \frac{a_i}{\sum_i a_i} - \frac{1-a_i}{\sum_i 1-a_i} \right\} \right| \right| \\
&\leq \left| \sum_{i=1}^{S} h(x_i) \left\{ \frac{a_i}{\sum_i a_i} - \frac{1-a_i}{\sum_i 1-a_i} \right\} - \sum_{i=1}^{S} \hat{h}(x_i) \left\{ \frac{a_i}{\sum_i a_i} - \frac{1-a_i}{\sum_i 1-a_i} \right\} \right| \\
&\leq \left| \sum_{i=1}^{S} (h(x_i) - \hat{h}(x_i)) \left\{ \frac{a_i}{\sum_i a_i} - \frac{1-a_i}{\sum_i 1-a_i} \right\} \right| \\
&\leq \sum_{i=1}^{S} \left| (h(x_i) - \hat{h}(x_i)) \left\{ \frac{a_i}{\sum_i a_i} - \frac{1-a_i}{\sum_i 1-a_i} \right\} \right| \\
&\leq \sum_{i=1}^{S} \left| (h(x_i) - \hat{h}(x_i)) \right| \left| \frac{a_i}{\sum_i a_i} - \frac{1-a_i}{\sum_i 1-a_i} \right| \\
&\leq \sum_{i=1}^{S} \left| (h(x_i) - \hat{h}(x_i)) \right| \quad \text{As} \left| \frac{a_i}{\sum_i a_i} - \frac{1-a_i}{\sum_i 1-a_i} \right| \leq 1 \\
&\leq \mu \quad \text{By Equation 7.12}
\end{aligned}$$

Hence the lemma holds true. □

Using the above Lemma 7.2, we can say that,

$$\hat{\mathcal{R}}_B(\mathcal{DP}) \leq \sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{DP}, \mu)\right)}{B}} + \mu \leq \sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{H}, \mu/S)\right)}{B}} + \mu$$

Hence applying the Lemma 7.1, we get

$$\mathbb{E}\left[const^{DP}(z_s)\right] \leq \frac{1}{B}\sum_{\ell=1}^{B} const^{DP}\left(z_S^{(\ell)}\right) + 2 \cdot \inf_{\mu>0}\left\{\mu + \sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{H}, \mu/S)\right)}{B}}\right\} + C\sqrt{\frac{\log(1/\delta)}{B}}$$

Bounds for EO

Given a fixed batch size S and $z_S = (h(x_S), a_S, y_S)$,

$$fpr(z_S) = \left| \frac{\sum_{i=1}^{S} h(x_i)(1-y_i)a_i}{\sum_{i=1}^{S} a_i} - \frac{\sum_{i=1}^{S} h(x_i)(1-y_i)(1-a_i)}{\sum_{i=1}^{S} 1-a_i} \right|$$

$$fnr(z_S) = \left| \frac{\sum_{i=1}^{S}(1-h(x_i))y_i a_i}{\sum_{i=1}^{S} a_i} - \frac{\sum_{i=1}^{S}(1-h(x_i))y_i(1-a_i)}{\sum_{i=1}^{S} 1-a_i} \right|$$

$$const^{EO}(z_S) = fpr(z_S) + fnr(z_S)$$

As defined in [5] is,

$$const^{EO}(z_S) = \max\{fpr(z_S), fnr(z_S)\} \leq fpr(z_S) + fnr(z_S)$$

Let us consider a class of $\mathcal{EO}$ functions defined on the class of $\mathcal{H}$ as follows,

$$\begin{aligned}\mathcal{EO} = \{const^{EO} : (h(X), \mathcal{A}, Y) \to \mathbb{R}|\ & const^{EO}(z_S) = \\ & \left| \frac{\sum_{i=1}^{S} h(x_i)(1-y_i)a_i}{\sum_{i=1}^{S} a_i} - \frac{\sum_{i=1}^{S} h(x_i)(1-y_i)(1-a_i)}{\sum_{i=1}^{S} 1-a_i} \right| \\ & + \left| \frac{\sum_{i=1}^{S}(1-h(x_i))y_i a_i}{\sum_{i=1}^{S} a_i} - \frac{\sum_{i=1}^{S}(1-h(x_i))y_i(1-a_i)}{\sum_{i=1}^{S} 1-a_i} \right| \\ & \text{for some } h \in \mathcal{H}\}\end{aligned}$$

Given l_∞ distance as

$$\max_{z_S} |\ const^{EO}(z_S) - \widehat{const}^{EO}(z_S)|$$

$\mathcal{N}_\infty(\mathcal{EO}, \mu)$ is the minimum number of balls of radius μ required to cover $\mathcal{EO}$ under the above distance. We apply Lemma 7.1 to the class of equalized odds class of functions $\mathcal{EO}$. Given a fixed batch size S, we have for any x_S, $const^{EO}(x_s) \leq 1$. By definition of the covering number $\mathcal{N}_\infty(\mathcal{EO}, \mu)$ for any class $const^{EO} \in \mathcal{EO}$, there exists a $\widehat{const}^{EO} \in \hat{\mathcal{EO}}$ where $|\hat{\mathcal{EO}}| \leq \mathcal{N}_\infty(\mathcal{EO}, \mu)$ such that $\max_{x_S} |const^{EO}(x_S) - \widehat{const}^{EO}| \leq \mu$, for a given $\mu \in (0,1)$. Given B batches of samples where batch is of fixed size S, similar to $\mathcal{DP}$ we can show that,

$$\hat{\mathcal{R}}_B(\mathcal{EO}) \leq \sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{EO}, \mu)\right)}{B}} + \mu \tag{7.13}$$

Lemma 7.3. $\mathcal{N}_\infty(\mathcal{EO}, \mu) \leq \mathcal{N}_\infty(\mathcal{H}, \mu/2S)$

Proof. For any $h, \hat{h} \in \mathcal{H}$ such that for all x_S we get

$$|h(x_i) - \hat{h}(x_i)| \leq \mu/2S \tag{7.14}$$

We know that $h(x_i) \in [0,1]$ $\forall h \in \mathcal{H}$. Now let us consider for the class of $\mathcal{EO}$,

$$\begin{aligned}
&|const^{EO} - \widehat{const}^{EO}| \\
=&\Bigg|\left|\frac{\sum_{i=1}^S h(x_i)(1-y_i)a_i}{\sum_{i=1}^S a_i} - \frac{\sum_{i=1}^S h(x_i)(1-y_i)(1-a_i)}{\sum_{i=1}^S 1-a_i}\right| \\
&+ \left|\frac{\sum_{i=1}^S (1-h(x_i))y_i a_i}{\sum_{i=1}^S a_i} - \frac{\sum_{i=1}^S (1-h(x_i))y_i(1-a_i)}{\sum_{i=1}^S 1-a_i}\right| \\
&- \left|\frac{\sum_{i=1}^S \hat{h}(x_i)(1-y_i)a_i}{\sum_{i=1}^S a_i} - \frac{\sum_{i=1}^S \hat{h}(x_i)(1-y_i)(1-a_i)}{\sum_{i=1}^S 1-a_i}\right| \\
&- \left|\frac{\sum_{i=1}^S (1-\hat{h}(x_i))y_i a_i}{\sum_{i=1}^S a_i} - \frac{\sum_{i=1}^S (1-\hat{h}(x_i))y_i(1-a_i)}{\sum_{i=1}^S 1-a_i}\right|\Bigg| \\
\leq&\left|\sum_{i=1}^S h(x_i)\left\{\frac{a_i(1-y_i)}{\sum_i a_i} - \frac{(1-a_i)(1-y_i)}{\sum_i 1-a_i}\right\} - \sum_{i=1}^S \hat{h}(x_i)\left\{\frac{a_i(1-y_i)}{\sum_i a_i} - \frac{(1-a_i)(1-y_i)}{\sum_i 1-a_i}\right\}\right| \\
&+ \left|\sum_{i=1}^S (1-h(x_i))\left\{\frac{a_i(1-y_i)}{\sum_i a_i} - \frac{(1-a_i)(1-y_i)}{\sum_i 1-a_i}\right\} - \sum_{i=1}^S (1-\hat{h}(x_i))\left\{\frac{a_i(1-y_i)}{\sum_i a_i} - \frac{(1-a_i)(1-y_i)}{\sum_i 1-a_i}\right\}\right| \\
\leq&\left|\sum_{i=1}^S (h(x_i) - h(\hat{x}_i))\left\{\frac{a_i(1-y_i)}{\sum_i a_i} - \frac{(1-a_i)(1-y_i)}{\sum_i 1-a_i}\right\}\right| \\
&+ \left|\sum_{i=1}^S (h(\hat{x}_i) - h(x_i))\left\{\frac{a_i(1-y_i)}{\sum_i a_i} - \frac{(1-a_i)(1-y_i)}{\sum_i 1-a_i}\right\}\right| \\
\leq&\sum_{i=1}^S 2\left|(h(x_i) - h(\hat{x}_i))\left\{\frac{a_i(1-y_i)}{\sum_i a_i} - \frac{(1-a_i)(1-y_i)}{\sum_i 1-a_i}\right\}\right| \\
\leq&\sum_{i=1}^S 2\left|(h(x_i) - h(\hat{x}_i))\right|\left|\frac{a_i(1-y_i)}{\sum_i a_i} - \frac{(1-a_i)(1-y_i)}{\sum_i 1-a_i}\right| \\
\leq&\sum_{i=1}^S 2\left|(h(x_i) - h(\hat{x}_i))\right| \quad \text{As} \left|\frac{a_i(1-y_i)}{\sum_i a_i} - \frac{(1-a_i)(1-y_i)}{\sum_i 1-a_i}\right| \leq 1 \\
\leq&\mu \quad \text{By Equation 7.14}
\end{aligned}$$

□

Using the above Lemma 7.3, we can say that,

$$\hat{\mathcal{R}}_B(\mathcal{EO}) \leq \sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{EO},\mu)\right)}{B}} + \mu \leq \sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{H},\mu/2S)\right)}{B}} + \mu$$

Hence applying the Lemma 7.1, we get

$$\mathbb{E}\left[const^{EO}(z_S)\right] \leq \frac{1}{B}\sum_{\ell=1}^{B} const^{EO}\left(z_S^{(\ell)}\right) + 2\cdot\inf_{\mu>0}\left\{\mu + \sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{H},\mu/2S)\right)}{B}}\right\} + C\sqrt{\frac{\log(1/\delta)}{B}}$$

Bounds for Cross Entropy

The loss for a sample i is given by,

$$l_{CE}(h(x_i), y_i) = -y_i \log(h(x_i)) - (1-y_i)\log(1-h(x_i))$$

We have $h(x_i) = \phi(f(x_i))$ where ϕ is the softmax over the neural network output $f(x_i)$ where $f \in \mathcal{F}$

Lemma 7.4. *[129] Let $\mathcal{H}$ be a bounded real-valued function space from some space $\mathcal{Z}$ and $z_1, \ldots, z_n \in \mathcal{Z}$. Let $\xi : \mathbb{R} \to \mathbb{R}$ be a Lipschitz with constant L and $\xi(0) = 0$. Then, we have*

$$E_\sigma \sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i\in[n]} \sigma_i \xi\left(h\left(\mathbf{z}_i\right)\right) \leq L E_\sigma \sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i\in[n]} \sigma_i h\left(\mathbf{z}_i\right)$$

Lemma 7.5. *$l_{CE}(.,.)$ is L Lipschitz with first argument hence,*

$$\hat{\mathcal{R}}_L(\mathcal{CE}\circ\mathcal{F}) \leq \hat{\mathcal{R}}_L(\mathcal{F})$$

where $\mathcal{CE} = \{l_{CE}(f(x), y) | \ \forall f \in \mathcal{F}\}$

Proof. Given that

$$l_{CE}(f(x), y) = y_i \log(y/\phi(f(x_i))) + (1-y_i)\log((1-y_i)/(1-\phi(f(x_i))))$$

It is easy to find that $\partial l_{CE}(f(x), y)/\partial f(x) \in [-1,1]^m$ and thus l_{CE} is a 1-Lipschitz function with its first argument. Given $l'_{CE}(.,.) = l_{CE}(.,.) - l_{CE}(0,.)$ and we can get that $\hat{\mathcal{R}}(l_{CE}\circ f) = \hat{\mathcal{R}}(l'_{CE}\circ f)$, then we apply Lemma 7.4 to l'_{CE} and conclude the proof. □

From Lemma 7.1 and 7.5, we obtain the following,

$$\mathbb{E}_{x\in\mathcal{X}}[l_{CE}(f(x),y)] \leq \frac{1}{B}\sum_{i=1}^{B} l_{CE}(f(x_i),y_i) + 2\hat{\mathcal{R}}_B(\mathcal{F}) + 4c\sqrt{\frac{2\log(4/\delta)}{B}} \tag{7.15}$$

From the above Equation 7.15 we need to compute $\hat{\mathcal{R}_L}(\mathcal{F})$. Given any sample x, $f(x) \leq L$. By definition of the covering number $\mathcal{N}_\infty(\mathcal{F},\mu)$ for any class $f \in \mathcal{F}$, there exists a $\widehat{f} \in \hat{\mathcal{F}}$ where $|\hat{\mathcal{F}}| \leq \mathcal{N}_\infty(\mathcal{F},\mu)$ such that $\max_x |f(x) - \widehat{f}| \leq \mu$, for a given $\mu \in (0,1)$. Given B samples,

$$\begin{aligned}
\hat{\mathcal{R}}_B(\mathcal{F}) &= \frac{1}{B}\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\sum_{\ell=1}^{B}\sigma_\ell \cdot f(x^{(\ell)})\right] \\
&= \frac{1}{B}\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\sum_{\ell=1}^{B}\sigma_\ell \cdot \widehat{f}\left(x^{(\ell)}\right)\right] + \frac{1}{B}\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\sum_{\ell=1}^{B}\sigma_\ell \cdot f\left(x_S^{(\ell)}\right) - \widehat{f}\left(x^{(\ell)}\right)\right] \\
&\leq \frac{1}{B}\mathbb{E}_\sigma\left[\sup_{\widehat{f}}\sum_{\ell=1}^{B}\sigma_\ell \cdot \widehat{f}\left(x^{(\ell)}\right)\right] + \frac{1}{B}\mathbb{E}_\sigma \parallel \sigma \parallel_1 \mu \qquad (7.16) \\
&\leq \sqrt{\sum_\ell \left(\widehat{f}\left(x_S^\ell\right)\right)^2}\sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{F},\mu)\right)}{B^2}} + \mu \quad \text{(By Massart's Lemma)} \\
&\leq L\sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{F},\mu)\right)}{B}} + \mu
\end{aligned}$$

The last inequality is because,

$$\sqrt{\sum_\ell \left(\widehat{f}\left(x_S^\ell\right)\right)^2} \leq \sqrt{\sum_\ell \left(f\left(x_S^\ell\right) + \mu\right)^2} \leq L\sqrt{B}$$

Hence,

$$\hat{\mathcal{R}}_B(\mathcal{F}) \leq L\sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{F},\mu)\right)}{B}} + \mu$$

Hence applying the Lemma 7.1, we get

$$\mathbb{E}[l_{CE}(f(x),y)] \leq \frac{1}{B}\sum_{i=1}^{B} l_{CE}(f(x_i),y_i) + 2\cdot\inf_{\mu>0}\left\{\mu + L\sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{F},\mu/S)\right)}{B}}\right\} + CL\sqrt{\frac{\log(1/\delta)}{B}}$$

Bounds for Q-mean loss

Given batch of size S, having samples (x_S, y_S).

Here we prove for when the dataset has m different classes, but for us $m = 2$

$$l_Q(h(x_S), y_S) = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(1 - \frac{\sum_{i=1}^{S} y_i^j h^j(x_i)}{\sum_{i=1}^{S} y_i^j}\right)^2}$$

Let us consider a class of $\mathcal{Q}$ functions defined on the class of $\mathcal{H}$ as follows,

$$\mathcal{Q} = \{l_Q : (X, Y) \to \mathbb{R} |\ l_Q(x_S) = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(1 - \frac{\sum_{i=1}^{S} y_i^j h^j(x_i)}{\sum_{i=1}^{S} y_i^j}\right)^2} \text{ for some } h \in \mathcal{H}\}$$

Define the $l_{\infty,1}$ distance as

$$\max_{x_S} \parallel\ l_Q(x_S) - \widehat{l}_Q(x_S) \parallel$$

$\mathcal{N}_\infty(\mathcal{Q}, \mu)$ is the minimum number of balls of radius μ required to cover $\mathcal{Q}$ under the above distance.

Let us now apply the Lemma 7.1 to the class of functions $\mathcal{Q}$. Given a fixed batch size S, we have for any x_S, y_S, $l_Q(x_S, y_S) \leq 1$. By definition of the covering number $\mathcal{N}_\infty(\mathcal{Q}, \mu)$ for any class $l_Q \in \mathcal{Q}$, there exists a $\widehat{l}_Q \in \hat{\mathcal{Q}}$ where $|\hat{\mathcal{Q}}| \leq \mathcal{N}_\infty(\mathcal{Q}, \mu)$ such that $\max_{(x_S, y_S)} |l_Q(x_S, y_S) - \widehat{l}_Q(x_S, y_S)| \leq \mu$, for a given $\mu \in (0, 1)$. Given B batches of samples where batch is of fixed size S, similar to $\mathcal{DP}$ we can show that,

$$\hat{\mathcal{R}}_B(\mathcal{Q}) \leq \sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{Q}, \mu)\right)}{B}} + \mu \tag{7.17}$$

Lemma 7.6. $\mathcal{N}_\infty(\mathcal{Q}, \mu) \leq \mathcal{N}_\infty(\mathcal{H}_S, \mu/S)$

Proof. We know that $h(x_i) \in [0, 1]^m\ \forall h \in \mathcal{H}$. For any $h, \hat{h} \in \mathcal{H}$ such that for all x_S we get, the following $l_{\infty,1}$

$$\parallel h(x_i) - \hat{h}(x_i) \parallel \leq \mu/S \tag{7.18}$$

Now let us consider for the class of $\mathcal{Q}$,

$$
\begin{aligned}
|const^Q - \widehat{const}^Q| &= \left| \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(1 - \frac{\sum_{i=1}^{S} y_i^j h^j(x_i)}{\sum_{i=1}^{S} y_i^j}\right)^2} - \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(1 - \frac{\sum_{i=1}^{S} y_i^j \hat{h}^j(x_i)}{\sum_{i=1}^{S} y_i^j}\right)^2} \right| \\
&\leq \frac{1}{\sqrt{m}} \left| \sqrt{\sum_{j=1}^{m}\left(1 - \frac{\sum_{i=1}^{S} y_i^j h^j(x_i)}{\sum_{i=1}^{S} y_i^j}\right)^2} - \sqrt{\sum_{j=1}^{m}\left(1 - \frac{\sum_{i=1}^{S} y_i^j \hat{h}^j(x_i)}{\sum_{i=1}^{S} y_i^j}\right)^2} \right| \\
&\leq \frac{1}{\sqrt{m}} \sqrt{\sum_{j=1}^{m} \left| \frac{\sum_{i=1}^{S} y_i^j \hat{h}^j(x_i)}{\sum_{i=1}^{S} y_i^j} - \frac{\sum_{i=1}^{S} y_i^j h^j(x_i)}{\sum_{i=1}^{S} y_i^j} \right|^2} \quad \text{Triangle Inequality} \\
&\leq \frac{1}{\sqrt{m}} \sum_{j=1}^{m} \left| \frac{\sum_{i=1}^{S} y_i^j \hat{h}^j(x_i)}{\sum_{i=1}^{S} y_i^j} - \frac{\sum_{i=1}^{S} y_i^j h^j(x_i)}{\sum_{i=1}^{S} y_i^j} \right| \\
&\leq \frac{1}{\sqrt{m}} \sum_{j=1}^{m} \left| \sum_{i=1}^{S} y_i^j (\hat{h}^j(x_i) - h^j(x_i)) \right| \quad \text{As } \sum_{i=1}^{S} y_i^j \geq 1 \\
&\leq \sum_{i=1}^{S} \sum_{j=1}^{m} \left| y_i^j (\hat{h}^j(x_i) - h^j(x_i)) \right| \\
&\leq \sum_{i=1}^{S} \sum_{j=1}^{m} \left| y_i^j (\hat{h}^j(x_i) - h^j(x_i)) \right| \\
&\leq \sum_{i=1}^{S} \parallel \hat{h}^j(x_i) - h^j(x_i) \parallel \quad \text{As } y_i^j \leq 1 \\
&\leq \mu \quad \text{By Equation 7.18}
\end{aligned}
$$

Hence the lemma holds true. □

Using the above Lemma 7.6, we can say that,

$$
\hat{\mathcal{R}}_B(\mathcal{Q}) \leq \sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{Q}, \mu)\right)}{B}} + \mu \leq \sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{H}, \mu/S)\right)}{B}} + \mu
$$

Hence applying the Lemma 7.1, we get

$$
\mathbb{E}\left[l_Q(x_S, y_S)\right] \leq \frac{1}{B}\sum_{\ell=1}^{B} l_Q\left(x_S^{(\ell)}, y_S^{(\ell)}\right) + 2 \cdot \inf_{\mu > 0}\left\{\mu + \sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{H}, \mu/S)\right)}{B}}\right\} + C\sqrt{\frac{\log(1/\delta)}{B}}
$$

□

The theorem below gives the bounds for the covering numbers for the class of neural networks that we use for our experiments

Theorem 7.3 (Dütting et al. 2017 [70]). *For the network with R hidden layers, D parameters, and vector of all model parameters $\| w \|_1 \leq W$. Given that w_l is bounded, the output of the network is bounded by some constant L.*

$$\mathcal{N}_\infty(\mathcal{F}, \mu/S) = \mathcal{N}_\infty(\mathcal{H}, \mu/S) \leq \left\lceil \frac{DLS(2W)^{R+1}}{\mu} \right\rceil^D$$

Hence, on choosing $\mu = \frac{1}{\sqrt{B}}$ we get,

$$\Omega_{DP} = \Omega_{EO} = \Omega_Q \leq \mathcal{O}\left(\sqrt{RD\frac{\log(WBSDL)}{B}}\right)$$

$$\Omega_L = \mathcal{O}\left(L\sqrt{RD\frac{\log(WBSDL)}{B}}\right)$$

Proof. Using the following lemma we prove the Theorem 7.3

Lemma 7.7 (Dütting et al. 2017 [70]). *Let $\mathcal{H}_k$ be a class of feed-forward neural networks that maps an input vector $x \in \mathbb{R}^d$ to an output vector $o \in \mathbb{R}$, with each layer l containing T_l nodes and computing $z \mapsto \phi_l(w^l z)$ where each $w^l \in \mathbb{R}^{T_l \times T_{l-1}}$ and $\phi_l : \mathbb{R}^{T_l} \to [-L, +L]^{T_l}$. Further let, for each network in $\mathcal{F}_k$, let the parameters $\| w^l \|_1 \leq W$ and $\| \phi_l(s) - \phi_l(s') \| \leq \Phi \| s - s' \|$ for any $s, s' \in \mathbb{R}^{T_{l-1}}$*

$$\mathcal{N}_\infty(\mathcal{F}_k, \mu) \leq \left\lceil \frac{2LD^2W(2\Phi W)^k}{\mu} \right\rceil^D$$

where D is the total number of parameters

The architecture that we use are 2 layered feed-forward neural networks with at most K hidden nodes per layer. For each layer l we assume, the $\| w_l \|_1 \leq W$. We know that ReLU activation and softmax activation are 1-Lipschitz [70]. Given that the input X has

d dimensions and w_l is bounded, the output of ReLU is bounded by some constant L. By applying Lemma 7.7 with $\Phi = 1$,

$$\mathcal{N}_\infty(\mathcal{H}, \mu/S) \leq \left\lceil \frac{DLS(2W)^{R+1}}{\mu} \right\rceil^D$$

Hence, on choosing $\mu = \frac{1}{\sqrt{B}}$ we get,

$$\begin{aligned}\Omega &\leq \frac{1}{\sqrt{B}} + \sqrt{\frac{2\log(\lceil (DLS(2W)^{R+1}B^{1/2}\rceil^D)}{B}} \\ &\leq \mathcal{O}\left(\sqrt{\frac{RD\log(WBSDL)}{B}}\right)\end{aligned}$$

where $\Omega = \{\Omega_{DP}, \Omega_{EO}, \Omega_Q\}$, similarly proof works for Ω_L □

Theorem 7.4. *Given $h(x) : X \to [0,1]$, for any $\hat{h}(x) : X \to [0,1]$ such that $h(x) \neq \hat{h}(x)$, we cannot define a $\widehat{const}_{DI} : (\hat{h}(X), \mathcal{A}, Y) \to \mathbb{R}$ for a $const_{DI} : (h(X), \mathcal{A}, Y) \to \mathbb{R}$ such that $|const_{DI} - \widehat{const}_{DI}| \leq \gamma$ is guaranteed, for any $\gamma > 0$. Thus, $\mathcal{N}_\infty(\mathcal{DI}, \mu)$ is unbounded for any $\mu > 0$ where $\mathcal{DI}$ is set of all possible $const_{DI}$.*

Proof. We prove the above theorem using the following lemma,

Lemma 7.8. *Given $a, b \geq 0$, $|\min(a, \frac{1}{a}) - \min(b, \frac{1}{b})| \leq |a - b|$*

Proof. It trivially holds true when,

- CASE 1: $\min(a, \frac{1}{a}) = a,\ \min(b, \frac{1}{b}) = b$
- CASE 2: $\min(a, \frac{1}{a}) = \frac{1}{a},\ \min(b, \frac{1}{b}) = b$

Let us consider the following cases,

- CASE 3: $\min(a, \frac{1}{a}) = a,\ \min(b, \frac{1}{b}) = \frac{1}{b}$
  We know that $a \leq 1$ hence, $2a \leq b + \frac{1}{b}$ which gives that $a - \frac{1}{b} \leq b - a$. for this case $a - b \leq a - \frac{1}{b}$, hence $|a - \frac{1}{b}| \leq |a - b|$

- CASE 4: $\min(a, \frac{1}{a}) = \frac{1}{a},\ \min(b, \frac{1}{b}) = \frac{1}{b}$

  In this case $|\frac{1}{a} - \frac{1}{b}| \leq |\frac{b-a}{ab}| \leq |a - b|$ as $a, b \geq 1$

□

Using the above lemma we prove the Theorem 7.4, For any $h, \hat{h} \in \mathcal{H}$ such that for all x_S we get,

$$|h(x_i) - \hat{h}(x_i)| \leq \mu \tag{7.19}$$

We assume that, $S = 100$, $\sum_{i=1}^{S} a_i = 50$ and $\sum_{i=1}^{S} 1 - a_i = 50$, for $a_i = 1, \hat{h}(x) = 1$ and $h(x) = 1$. For $a_i = 0, \hat{h}(x) = \mu$ and $h(x) = \delta$ where $\delta \in (0, 1)$ s.t $|\mu - \delta| \leq \mu$. Now let us consider for the class of $\mathcal{DI}$,

$$\begin{aligned}
|const^{DI} - \widehat{const}^{DI}| =& \left| min\left(\frac{\frac{\sum_{i=1}^{S} a_i h(x_i)}{\sum_{i=1}^{S} a_i}}{\frac{\sum_{i=1}^{S}(1-a_i)h(x_i)}{\sum_{i=1}^{S} 1-a_i}}, \frac{\frac{\sum_{i=1}^{S}(1-a_i)h(x_i)}{\sum_{i=1}^{S} 1-a_i}}{\frac{\sum_{i=1}^{S} a_i h(x_i)}{\sum_{i=1}^{S} a_i}} \right) \right. \\
& \left. - \, min\left(\frac{\frac{\sum_{i=1}^{S} a_i \hat{h}(x_i)}{\sum_{i=1}^{S} a_i}}{\frac{\sum_{i=1}^{S}(1-a_i)\hat{h}(x_i)}{\sum_{i=1}^{S} 1-a_i}}, \frac{\frac{\sum_{i=1}^{S}(1-a_i)\hat{h}(x_i)}{\sum_{i=1}^{S} 1-a_i}}{\frac{\sum_{i=1}^{S} a_i \hat{h}(x_i)}{\sum_{i=1}^{S} a_i}} \right) \right| \\
\leq& \left| \frac{\frac{\sum_{i=1}^{S} a_i h(x_i)}{\sum_{i=1}^{S} a_i}}{\frac{\sum_{i=1}^{S}(1-a_i)h(x_i)}{\sum_{i=1}^{S} 1-a_i}} - \frac{\frac{\sum_{i=1}^{S} a_i \hat{h}(x_i)}{\sum_{i=1}^{S} a_i}}{\frac{\sum_{i=1}^{S}(1-a_i)\hat{h}(x_i)}{\sum_{i=1}^{S} 1-a_i}} \right| \quad \text{By Lemma 7.8} \\
\leq& \left| \frac{\sum_{i=1}^{S}(1 - a_i)}{\sum_{i=1}^{S} a_i} \frac{\sum_{i=1}^{S} a_i h(x_i)}{\sum_{i=1}^{S}(1 - a_i)h(x_i)} - \frac{\sum_{i=1}^{S}(1 - a_i)}{\sum_{i=1}^{S} a_i} \frac{\sum_{i=1}^{S} a_i \hat{h}(x_i)}{\sum_{i=1}^{S}(1 - a_i)\hat{h}(x_i)} \right| \\
\leq& \left| \frac{\sum_{i=1}^{S}(1 - a_i)}{\sum_{i=1}^{S} a_i} \left(\frac{\sum_{i=1}^{S} a_i h(x_i)}{\sum_{i=1}^{S}(1 - a_i)h(x_i)} - \frac{\sum_{i=1}^{S} a_i \hat{h}(x_i)}{\sum_{i=1}^{S}(1 - a_i)\hat{h}(x_i)} \right) \right| \\
\leq& \left| \frac{50}{50\delta} - \frac{50}{50\mu} \right| \\
\leq& \left| \frac{1}{\delta} - \frac{1}{\mu} \right|
\end{aligned}$$

Given a fixed μ, we can have an arbitrarily small δ such that the above becomes unbounded, hence the theorem follows.

□

We emphasize that, Theorem 7.4 indicates that if we approximate DI by a surrogate constraint, however close the learnt classifier is to a desired classifier, the actual DI constraint may get unbounded under specific instances. That is, even two close classifiers (i.e., $|h(x) - \hat{h}(x)| < \mu$ for any $\mu \in (0, 1)$) may have arbitrarily different DI. For our problem, due to this negative results, we cannot give generalization guarantees by using $\mathcal{N}_\infty(\mathcal{DI}, \mu)$ as an upper bound. The few cases where, DI becomes unbounded may not occur in practice as we observe in our experiments that DI results are also comparable.

While training the network, in the loss we use the ϵ-relaxed fairness constraints as defined in Equation 7.5. We believe that, given the above generalization bounds for the constraints, the trained model will be ϵ-fair with the same bounds.

7.6 Implementation Details and Experimental Analysis

In this section, we discuss the network parameters and summarize the results. The architecture that we used is a simple two-layered feed-forward network. The number of hidden neurons in both the layers was one of the following $(100, 50), (200, 100), (500, 100)$. As fairness constraint has no meaning for a single sample, SGD optimizer cannot be used. Hence we use batch sampling. We fix the batch size to be either 1000 or 500 depending on the dataset, to get proper estimates of the loss while training. It is to be noted that batch processing is mandatory for this network to be trained efficiently. For training, we have used the Adam Optimizer with a learning rate of 0.01 or 0.001 and the training typically continues for a maximum of 5000 epochs for each experiment before convergence. The results are averaged over 5-fold cross-validation performance on the data.

Performance across Datasets

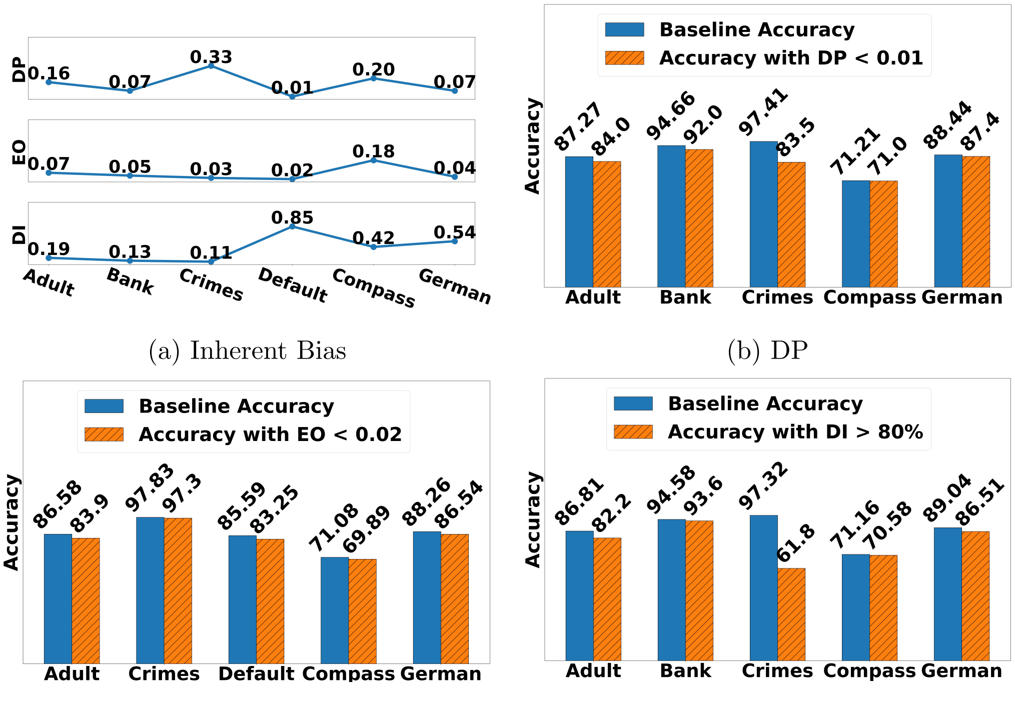
We have conducted experiments on the six most common datasets used in fairness domain. In Adult, Default, and German dataset, we use gender as the sensitive attribute while predicting income, crime rate, and quality of the customer, respectively, in each of the datasets. In Default and Compass datasets that we used, the race was considered as the sensitive attribute while predicting default payee and recidivism respectively. In the Bank dataset, age is the sensitive attribute while predicting the income of the individual.

In Figure 7.1a we observe the inherent biases corresponding to each fairness measure within the datasets considered. In order to obtain the values, we set $\lambda = 0$ in Equation 7.9 while training. We compare the baseline accuracy, that is obtained by setting $\lambda = 0$, and accuracy using FNNC. In Figure 7.1b, we observe a drop in accuracy when the model is trained to limit DP violations within 0.01, i.e., $\epsilon = 0.01$. There is a significant drop in the accuracy of the Crimes dataset, where the DP is violated the most. Similarly, in Figure 7.1c and Figure 7.1d, we study the effect of training the models to limit EO and DI, respectively. We observe that the drop in accuracy is more for datasets that are inherently more biased. In the following section, we compare with other works and all the results are mostly reported on Adult and Compass dataset.

Comparative Results

In this subsection, we compare our results with related work.

- Bilal Zafar et al. 2015 [36]: In this work, the authors propose C-SVM and C-LR to maintain DI while maximizing accuracy. We compare our results with theirs on Adult and bank datasets as observed in the Figure 7.2. We can see that FNNC obtains higher accuracy for ensuring $p\%$ DI rule for upto $p = 80$, for $p > 80$, the accuracy reduces by 2 %. For obtaining the results we train our network using the loss given in Equation 7.5 with $const^{DI}$.



(a) Inherent Bias

(b) DP

(c) EO

(d) DI

Figure 7.1: Comparison across datasets

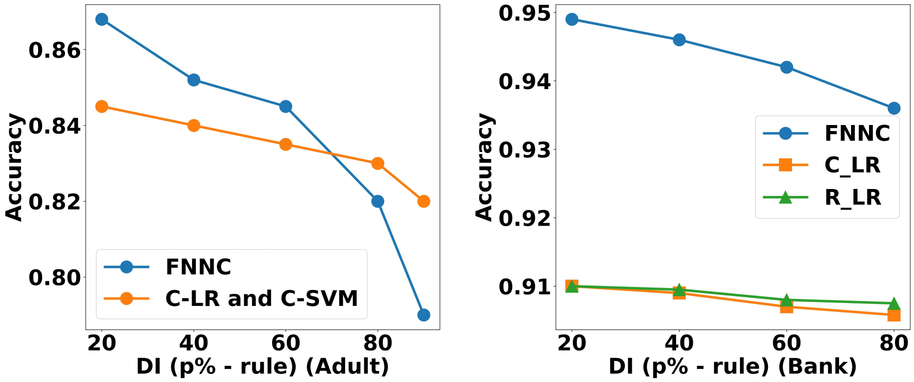


Figure 7.2: Accuracy vs $p\% - rule$ comparison of results with Zafar *et al.* on Adult dataset in the left subplot and Bank dataset in the right subplot

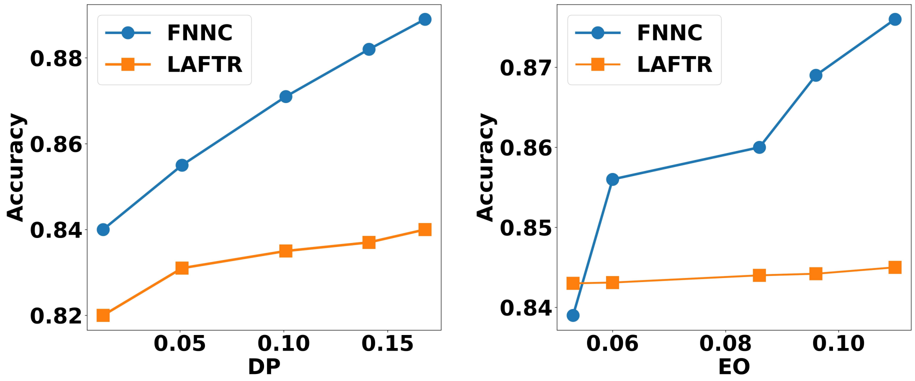


Figure 7.3: Accuracy vs ϵ (ϵ is tolerance for DP and EO respectively) and compare with Madras *et al.* on Adult dataset

		Female	Male
Zhang et al. 2018 [203]	FPR	0.0647	0.0701
	FNR	0.4458	0.4349
FNNC	FPR	0.1228	0.1132
	FNR	0.0797	0.0814

Table 7.1: False-Positive Rate (FPR) and False-Negative Rate (FNR) for income prediction for the two sex groups in Adult dataset

- Madras et al. 2018 [132]: In this work, the authors propose LAFTR to ensure DP and EO while maximizing accuracy on Adult dataset. We have compared our results with theirs in Figure 7.3. For this, we have used loss defined in Equation 7.5 with $const^{DP}, const^{EO}$.
- Zhang et al. 2018 [203]: The authors have results for EO on Adult Dataset as can be found in Table 7.1. Less violation of EO implies that the FPR and FNR values are almost same across different attributes. We get FPR (female) 0.1228 $\sim$ FPR (male) 0.1132 and FNR values for female and male are 0.0797 $\sim$ 0.0814. The accuracy of

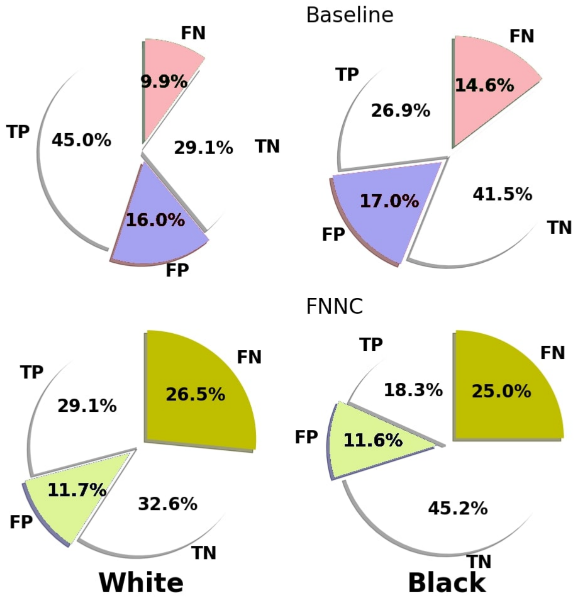


Figure 7.4: Compass dataset: The FPR and FNR is comparable across race in FNNC as observed in the bottom left and right pie charts

the classifier remains at 85%. Similarly, we have experiments on Compass dataset and compare FNNC with the baseline (trained only for accuracy) in Figure 7.4

- Agarwal et al. 2018 [5]: We compare our results with theirs on Adult and Compass Dataset both for DP and EO as given in Figure 7.5. On observing the plots we find our performance is better for Compass dataset but worse for Adult dataset. The violation of EO in Compass dataset is less compared to the Adult dataset as observed in Figure 7.1a. Hence, the cost of maintaining fairness is higher in Adult dataset. We can observe in Figs. 7.2 7.3, 7.5, that as the fairness constraint is too strict, i.e., ϵ is very small or $p > 80$, the accuracy reduce or error increases.
- Narasimhan 2018 [152]: The authors propose COCO and FRACO algorithm for fractional convex losses with convex constraints. In Table 7.2, we have results for Q-mean loss with DP as the constraint, whose loss function is given by Equation 7.8.

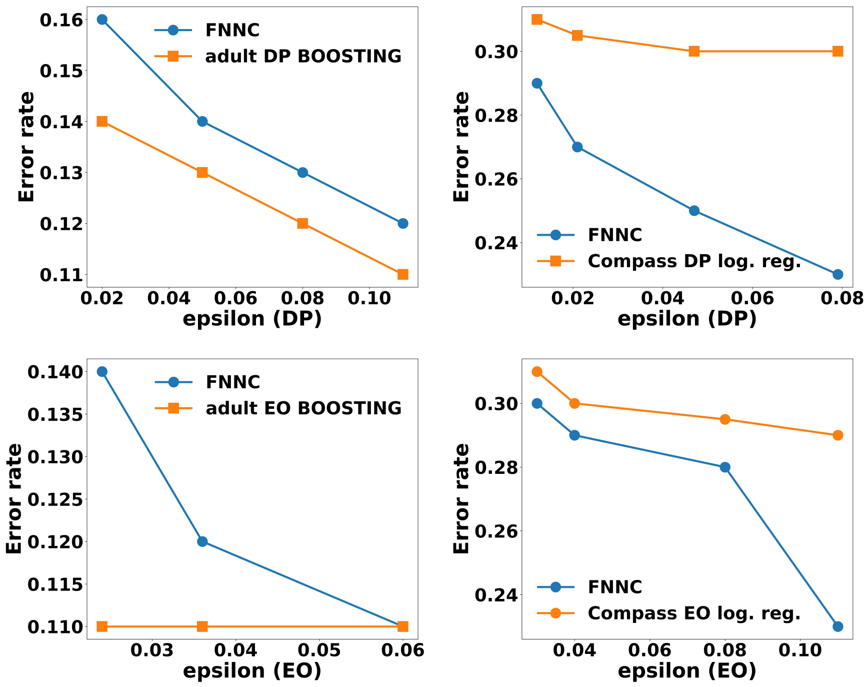


Figure 7.5: We compare our results with Agarwal et al. 2018 [5] for Error rate vs (ϵ) tolerance of DP in top row and EO in bottom row

Dataset	ϵ	FNNC	COCO	LinCon
adult	0.05	0.28 (0.027)	0.33 (0.035)	0.39 (0.027)
compass	0.20	0.32 (0.147)	0.41 (0.206)	0.57 (0.107)
crimes	0.20	0.28 (0.183)	0.32 (0.197)	0.52 (0.190)
default	0.05	0.29 (0.011)	0.37 (0.032)	0.54 (0.015)

Table 7.2: Q-mean loss s.t. DP is within ϵ (actual DP in parentheses)

In the table the values inside the parenthesis correspond to the DP obtained during testing and the values outside the parenthesis is the Q-mean loss. We achieve lower Q-mean loss when compared on 4 datasets while DP stays within ϵ.

7.7 Conclusion

The results prove that neural networks perform remarkably well on complex and non-convex measures using batch training. From the analysis on generalization bounds, in Theorem 7.3, we see that, as $B \to \infty$, $\Omega \to 0$. As the number of batches of samples increase, the generalization error asymptotically reduces to zero. The batch size S that we use during the training of the network is a crucial parameter. The generalization error increases in $\sqrt{\log S}$ and also increasing S would reduce B (for fixed data-set). Thus, a smaller value of S is preferable for better generalization. On the other hand, having a very small S, would not give a good estimate of the fairness constraint itself. We may end up with sub-optimal classifiers with high loss and less generalization error. Hence, the right balance between S and B is needed to get optimal results.

We believe that the neural networks can learn optimal feature representation of the data to ensure fairness while maintaining accuracy in an end-to-end manner. Hence, our method, FNNC, combines the traditional approach which learns fair representations by pre-processing the data and the approach for training a fair classifier using surrogate losses. One could consider implementing other non-decomposable performance measures like F1-score, Precision, recall, etc., using this approach, and we leave this for future work.

Chapter 8

Towards Building Ethical AI – Fair and Private Classifier

Deep learning's unprecedented success raises several ethical concerns ranging from biased predictions to data privacy. Researchers tackle these issues by introducing fairness metrics, or federated learning, or differential privacy. A first, in this work we present an ethical federated learning model, incorporating all three measures simultaneously. Experiments on the Adult, Bank and Dutch datasets highlight the resulting "empirical interplay" between accuracy, fairness, and privacy.

8.1 Introduction

The success in DL is made possible due to the availability of big datasets. In recent times, the data collection process is being outsourced – typically through *crowdsourcing* [1]. To build a DL system that uses big dataset, the system designer faces the following three significant challenges: (C1) *computational* challenges of big datasets, (C2) ensuring predictions that are socially *fair* towards all demographic groups even when trained on imbalanced data and (C3) preserving the privacy of (often) delicate individual information present in the dataset. This information comprises any individual information based on the training data [1] and any information related to the sensitive attribute [189].

For handling big datasets and towards addressing C1, researchers propose to *distribute* the training process in DL. Such a proposal, referred to as *Federated learning* (FL) [139], aims to increase the training efficiency by allowing 'clients' (or *agents*) to train individual models parallelly over their *private* dataset. The parallelization significantly reduces the training time. A 'central server,' referred henceforth as an *aggregator*, receives the individual models and arrives at an overall model through different heuristics [194].

Machine Learning models supplement human evaluations in applications like criminal risk assessment, credit approvals, online advertisements. C2 relates to the fact that these model's decisions are known to discriminate towards certain demographic groups like gender, age, or race [25, 30, 53]. Researchers have proposed notions of group-fairness that bounds the differences in the model's performance on different demographic groups [71, 105]. From [53], we note that achieving these perfectly unbiased model is impossible. Hence various approaches minimize the bias while maintaining high accuracy [5, 36, 132, 158]. Invariably all these approaches require the information of the sensitive attribute. Typically, the law regulations at various places prohibit using such attributes to develop models such as EU General Data Protection Regulation prevents the collection of protected user attributes. One must address the issue of discrimination while protecting the sensitive attribute information of samples [189].

The aggregator in FL has no direct access to private data, which prima facie preserves privacy. However, there exist several attacks that highlight the information leak in an FL setting [146]. To concretely address C3, the existing literature either uses cryptographic solution based mainly on complex *Partial Homomorphic Encryption* (PHE) or through Differential Privacy (DP). While private FL solutions using PHE exist in the literature [76, 144, 200, 204], these suffer from computational inefficiency and post-processing attacks. To this end, researchers focus on the rigorous privacy guarantees provided by a *differentially-private* solution [154, 162, 177, 196].

C1-C3 raise an important *ethical* challenge: We must design the training process such that the predictions from the learned model are non-discriminatory and also preserve information regarding the training data and sensitive attributes. We aim to resolve this challenge by incorporating appropriate ethical notions in an FL model. We focus on the following notions.

Fairness.

- *Demographic Parity (DemP)*: states that a model's rate of prediction of the positive outcome is equal across the demographic groups, i.e. independent of the sensitive attribute [71]. Note that the base rate of the positive outcome may not be equal in the training label itself. Therefore, DemP is a bias transforming fair metric [193]. It ensures a fair outcome despite the bias in the existing data.

- *Equalized Odds (EO)*: states that the false positive rates and false negative rates of a model are equal across different groups or independent of the sensitive attribute [105]. This metric is defined under the assumption that the existing data is non-discriminatory and hence bias preserving [193].

Privacy. As aforementioned, we quantify the privacy guarantees with the strong notion of Differential Privacy (DP) [72]. As in the FL training process, each agent trains its model using its data, and we focus on the stronger *local*-Differential Privacy (DP) model [72, Chapter 12]. We look towards ensuring the privacy of the training data and the sensitive attribute from an observant adversary. We consider the "black-box" model for our adversary, i.e., our adversary has access to the trained model and can interact with it via inputs and outputs. With this information, the adversary can perform model-inversion attacks [79], among others.

In our FL setting, the aggregator acts as an adversary with access to each agent's model. Consequently, it suffices to provide rigorous DP guarantees for any possible information leak to the aggregator towards designing a private FL system. The post-processing properties of

DP will further preserve the DP guarantee for the training data and the sensitive attribute from any other party.

Our Contributions

1. We present our novel framework *FPFL*: Fair and Private Federated Learning (Figure 8.2), which learns fair and accurate models while providing a strong local-DP guarantee.
2. We prove that FPFL provides the local-DP guarantee for both the training data and the sensitive attribute(s) (Proposition 8.1).
3. Our experiments on the Adult, Bank and Dutch datasets show the *three-way trade-off*, i.e., the simultaneous trade-off between fairness, privacy, and accuracy of an FL model (Section 8.6).

8.2 Existing Approaches

Handful of works are present that simultaneously study fairness and privacy in ML [20, 60, 150, 189]. The authors in [20] show that privacy has a *negative* impact on a model's fairness; while Tran et al. 2020 [189] considers the notion of DemP with DP. What is more, to the best of our knowledge, only [189] looks at the confidentiality of the sensitive attribute. The critical difference in our work and [189] is that the authors there simultaneously train the model to achieve accuracy and fairness while ensuring DP. They only add noise to the gradients from the fairness loss to not comprise the resulting accuracy. The implication is that their approach only preserves the sensitive attribute and *not* the training data. In contrast, in FPFL, by decoupling the training process by first improving fairness followed by ensuring DP, we end up not compromising on the accuracy while protecting both – the sensitive attribute and the training data.

Concerning FL, researchers look at the trade-off between a model's fairness and accuracy [67, 123] and between privacy and accuracy [154]; but *not* both simultaneously. In particular, Wei et al. 2020 [196] presents NbAFL, a differentially-private FL algorithm for model aggregation. In NbAFL, the authors add noise to the model parameters at the client's side before and at the server's side after aggregating. The authors argue that the added noise will make the model resistant to an observing adversary who can learn sensitive information during client-server communication(s); and present theoretical bounds for the information leakage. However, their proposal adds noise to the trained model but not the gradients and one can use efficient *symmetric encryption* schemes such as SSH [201] to ensure that an adversary observing the client-server communication does not gain any delicate information. Their models also are susceptible to an information leak from gradients [11].

8.3 Preliminaries

In this work, we consider a binary classification problem with $\mathcal{X}$ as our (d-dimensional) instance space, $\mathcal{X} \in \mathbb{R}^d$; and our output space as $\mathcal{Y} \in \{0, 1\}$. We consider the input space comprising a *protected* (sensitive) attribute $\mathcal{A}$ associated with each individual instance. Such an attribute may represent sensitive information like age, gender or caste. Each $a \in \mathcal{A}$ represents a particular category of the sensitive attribute like male or female.

8.3.1 Federated Learning Model

Federated Learning (FL) decentralizes the classical machine learning training process. FL comprises two type of actors: (i) a set of agents $\mathbb{A} = \{1, \ldots, m\}$ where each agent i owns a *private* dataset $\mathcal{X}_i$[1][2]; and (ii) an *Aggregator*. Each agent provides its model, trained on its dataset, to the aggregator. The aggregator's job then is to derive an overall model, which is then communicated back to the agents. This back-and-forth process continues until a model with sufficient accuracy is derived.

[1] Let $|\mathcal{X}_i|$ denote the cardinality of $\mathcal{X}_i$ with $\mathbf{X} = \sum_i |\mathcal{X}_i|$.

[2] We use the sub-script "i" when referring to a particular agent i and drop it when not referring to any particular agent.

More formally, at the start of an FL training, the aggregator communicates an initial, often random, set of model parameters to the agents. Let us refer to the initial parameters as θ_0. At each timestep t each agent updates their individual parameters denoted by $\theta_{(i,t)}$, using their private datasets. These agents communicate the update parameters to the aggregator, who derives an overall model through different heuristics [194]. We focus on the weighted sum heuristics, i.e., the overall model parameters take the form $\theta_t = \sum_{j \in \mathbb{A}} \frac{|\mathcal{X}_j|}{\mathbf{X}} \cdot \theta_{(j,t)}$. We refer to the final overall model with θ^*, calculated at a time step T.

8.3.2 Fairness Metrics

Definition 8.1 (*Demographic Parity (DemP)*)**.** *A classifier h satisfies Demographic Parity under a distribution over $(\mathcal{X}, \mathcal{A}, \mathcal{Y})$ if its predictions $h(\mathcal{X})$ is independent of the protected attribute $\mathcal{A}$. That is, $\forall a \in \mathcal{A}$ and $p \in \{0,1\}$,*

$$\Pr[h(\mathcal{X}) = p | \mathcal{A} = a] = \Pr[h(\mathcal{X}) = p]$$

Given that $p \in \{0,1\}$, we have $\forall a$

$$\mathbb{E}[h(\mathcal{X}) | \mathcal{A} = a] = \mathbb{E}[h(\mathcal{X})].$$

Definition 8.2 (*Equalized Odds (EO)*)**.** *A classifier h satisfies Equalized Odds under a distribution over $(\mathcal{X}, \mathcal{A}, \mathcal{Y})$ if its predictions $h(\mathcal{X})$ are independent of the protected attribute $\mathcal{A}$ given the label $\mathcal{Y}$. That is, $\forall a \in \mathcal{A},\ p \in \{0,1\}$ and $y \in \mathcal{Y}$*

$$\Pr[h(\mathcal{X}) = p | \mathcal{A} = a, \mathcal{Y} = y] = \Pr[h(\mathcal{X}) = p | \mathcal{Y} = y]$$

Given that $p \in \{0,1\}$, we can say $\forall a, y$

$$\mathbb{E}[h(\mathcal{X}) | \mathcal{A} = a, \mathcal{Y} = y] = \mathbb{E}[h(\mathcal{X}) | \mathcal{Y} = y].$$

8.3.3 Differential Privacy

We now define local differential privacy in the context of our FL model.

Definition 8.3 (*Local Differential Privacy (LDP)* [72]). *For an input set $\mathcal{X}$ and the set of noisy outputs $\mathcal{Y}$, a randomized algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is said to be (ϵ, δ)-LDP if $\forall x, x' \in \mathcal{X}$ and $\forall y \in \mathcal{Y}$ the following holds,*

$$\Pr[\mathcal{M}(x) = y] \leq \exp(\epsilon) \Pr[\mathcal{M}(x') = y] + \delta. \tag{8.1}$$

Informally, LDP provides a statistical guarantee against an inference which the adversary can make based on the output of $\mathcal{M}$. This guarantee is upper-bounded by ϵ, which is often referred to as the *privacy budget*. ϵ is a metric of *privacy loss* defined as,

$$L^{y}_{\mathcal{M}(x)||\mathcal{M}(x')} = \ln\left(\frac{\mathcal{M}(x) = y}{\mathcal{M}(x') = y}\right). \tag{8.2}$$

The privacy budget, ϵ, controls the trade-off between the quality (or, in our case, the accuracy) of the output vis-a-vis the privacy guarantee. That is, there is no "free-dinner" – lower the budget, better the privacy. However, at the cost of quality. The "δ" parameter in Equation (8.1) allows for the violation of the upper-bound ϵ, but with a small probability.

DP-SGD: Moments Accountant

Differentially private ML solutions focus on preserving an individual's privacy within a dataset. Such privacy may be compromised during the training process or based on the predictions of the trained model [79]. The most famous of such an approach is the DP-SGD algorithm, introduced in [1]. In DP-SGD, the authors sanitize the gradients provided by the Stochastic Gradient Descent (SGD) algorithm with *Gaussian* noise ($\mathcal{N}(0, \sigma^2)$). This step aims at controlling the impact of the training data in the training process. The added noise is calibrated to w.r.t. the metric *sensitivity* which is defined as the maximum possible change in $\mathcal{M}$'s output. More formally, $\mathcal{M}$'s sensitivity $\Delta_{\mathcal{M}}$ is defined as:

$$\Delta_{\mathcal{M}} = \max_{x,x'} |\mathcal{M}(x) - \mathcal{M}(x')|. \tag{8.3}$$

The authors then present the *moments accountant*: a method which keeps track of the increasing privacy budget through the algorithms training process. The specific ϵ, δ

bounds for DP-SGD, given by the moments accountant, is provided in [1, Theorem 1]. For completeness, we restate the bound as the following equation.

$$\sigma \geq c_2 \frac{q\sqrt{T \ln(1/\delta)}}{\epsilon}. \tag{8.4}$$

(8.4) holds with constants c_1 and c_2 s.t. $\epsilon < c_1 q^2 T$ and $\delta > 0$. Here, q is the sampling probability, i.e., $q = \frac{B}{|\mathcal{X}|}$.

8.4 Proposed Framework: FPFL

In FPFL (Figure 8.1), we consider a classification problem. Each agent i deploys two multi-layer *neural networks* (NNs) to learn the model parameters in each phase. The training comprises of *two* phases: (i) In Phase 1, each agent privately trains a model on its private dataset to learn a highly fair and accurate model; and (ii) In Phase 2: each agent sends its trained model to the aggregator. That is, in FPFL, only the model trained in Phase 2 is broadcasted to the aggregator. This process is akin to knowledge distillation [109].

To enhance readability and to remain consistent with FL notations, we denote the model parameters learned for Phase 1 with ϕ and Phase 2 with θ. Likewise, we represent the total number of training steps in Phase 1 with T_1, and for Phase 2, we use T_2.

8.4.1 Phase 1: Fair-SGD

In this phase, we train the network to maximize accuracy while achieving the best possible fairness on each agent's private dataset. We adapt the *Lagrangian Multiplier method* [158] to achieve a fair and accurate model. We denote the model for agent i as h^{ϕ_i} with parameters ϕ_i. Briefly, the method trains a network with a unified loss that has two components. The first component of the loss maximizes accuracy, i.e., the cross-entropy

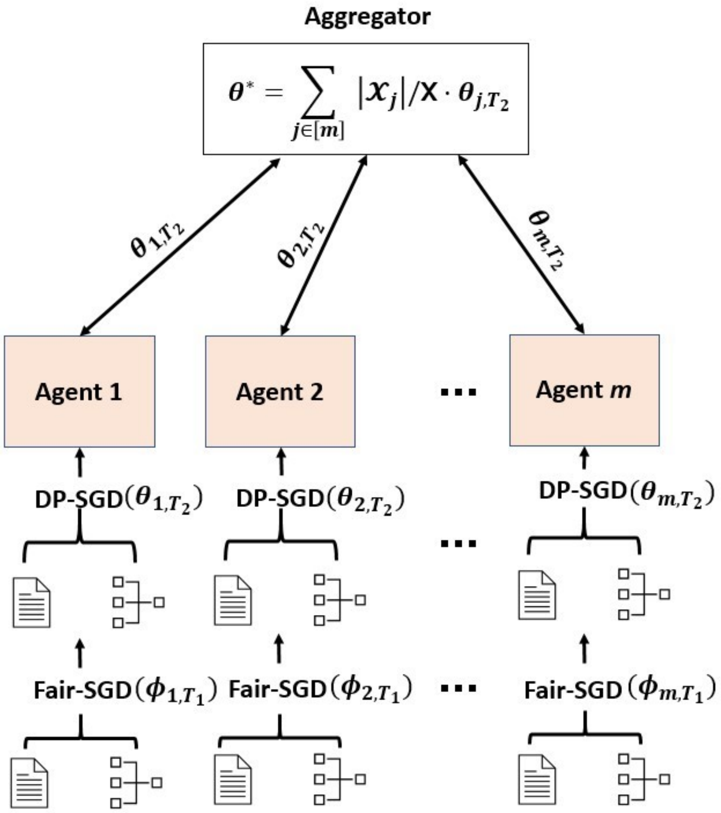


Figure 8.1: FPFL Model.

loss,

$$l_{CE}(h^{\phi_i}, \mathcal{X}, \mathcal{Y}) = \mathop{\mathbb{E}}_{(x,y)\sim(\mathcal{X},\mathcal{Y})}\big[-y_i \log(h^{\phi_i}(x)) - (1-y)\log(1-h^{\phi_i}(x))\big]$$

The second component of the loss is a specific fairness measure. For achieving DemP (Definition 8.1), the loss function is given by,

$$l_{DemP}(h^{\phi_i}, \mathcal{X}, \mathcal{A}) = |\mathbb{E}[h^{\phi_i}(x)|\mathcal{A} = a] - \mathbb{E}[h^{\phi_i}(x)]| \tag{8.5}$$

For achieving EO (Definition 8.2), the corresponding loss function is,

$$l_{EO}(h^{\phi_i}, \mathcal{X}, \mathcal{A}, \mathcal{Y}) = |\mathbb{E}[h^{\phi_i}(x)|\mathcal{A} = a, y] - \mathbb{E}[h^{\phi_i}(x)|y]| \tag{8.6}$$

FPFL Framework

1. **Initialization**
2. **Local Training Process.** Each agent $i \in \mathbb{A}$, invokes Algorithm 10
3. **Local Training Process.** Each agent $i \in \mathbb{A}$, invokes Algorithm 11
4. Local training process ends
5. **Model Aggregation.** Aggregator computes and then broadcasts an overall model
6. Agents re-initialize their local models with the overall model received
7. Repeat steps from 3 to 6 until a sufficient overall accuracy is reached

Figure 8.2: FPFL Framework

Hence, the overall loss from the Lagrangian method is,

$$L_1(h^{\phi_i}, \mathcal{X}, \mathcal{A}, Y) = l_{CE} + \lambda l_k, \quad k \in \{DemP, EO\} \tag{8.7}$$

In the equation above, $\lambda \in \mathbb{R}^+$ is the Lagrangian multiplier. The overall optimization is as follows,

$$\underset{\phi}{min}\ \underset{\lambda}{max}\ L_1$$

Thus, each agent trains the Fair-SGD model h_i^{ϕ} to obtain the best accuracy w.r.t. a given fairness metric. The formal algorithm for this phase is presented in Algorithm 10.

8.4.2 Phase 2: DP-SGD

In this phase, the agents train a model that is communicated with the aggregator. This model denoted by h^{θ_i} is trained by each agent i to learn the predictions of its own Fair-SGD model (h^{ϕ_i}) from Phase 1. The loss function is given by,

$$L_2(h^{\theta_i}, h^{\phi_i}) = \underset{x \sim \mathcal{X}}{\mathbb{E}}[-h^{\phi_i}(x)\log(h^{\theta_i}(x)) - (1 - h^{\phi_i}(x))\log(1 - h^{\phi_i}(x))] \tag{8.8}$$

To preserve the privacy of training data and sensitive attributes, we use the local version of (ϵ, δ)-DP (Definition 8.3). In particular, we deploy the known DP-SGD algorithm [1, Algorithm 1]. In DP-SGD, the privacy of the training data is preserved by sanitizing the gradients provided by SGD with *Gaussian* noise ($\mathcal{N}(0, \sigma^2)$). This step aims at controlling the impact of the training data in the training process.

Given that the learnt model h^{θ_i}, mimics h_i^{ϕ}, the model is reasonably fair and accurate. For completeness, the formal algorithm for this phase is presented in Algorithm 11.

Algorithm 10 Fair-SGD for an Agent i

```
**Input:** Training dataset $\mathcal{X}_i = \{x_1, \ldots, x_n\}$, Loss function $L_1(\cdot)$ as defined in (8.7).
Hyperparameters: learning rate $\eta$, batch size $B$, sampling probability $q = B/|\mathcal{X}_i|$.
**Output:** $\phi_{(i,T_1)}$
**Initialization:** $\phi_{(i,0)} \leftarrow$ randomly
for $t \in [T_1]$ do
    Take a random sample $B_t$ with probability $q$
    $\forall k \in B_t$: $\mathbf{g}_t(x_k) \leftarrow \nabla_{\phi_{(i,t)}} L_1(\cdot)$
    $\phi_{(i,t+1)} \leftarrow \phi_{(i,t)} - \eta_t \mathbf{g}_t(x_k)$
end for
```

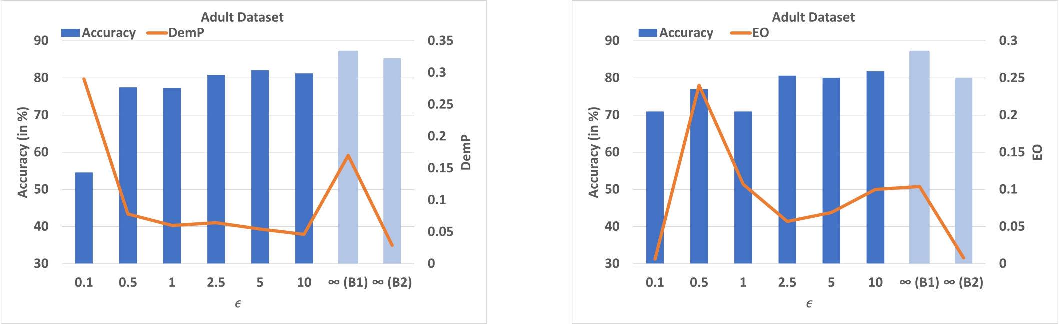
Algorithm 11 DP-SGD for an Agent i [1]

```
Input: Training dataset $\mathcal{X}_i = \{x_1, \ldots, x_n\}$, Loss function $L_2(\cdot)$ as defined in (8.8).
Hyperparameters: learning rate $\eta$, standard deviation $\sigma$, batch size $B$, sampling probability $q = B/|\mathcal{X}_i|$ and clipping norm $C$.
Output: $\theta_{(i,T_2)}$
Initialization: $\theta_{(i,0)} \leftarrow$ randomly
for $t \in [T_2]$ do
    Take a random sample $B_t$ with probability $q$
    $\forall k \in B_t$: $\mathbf{g}_t(x_k) \leftarrow \nabla_{\theta_{(i,t)}} L_2(\cdot)$
    $\bar{\mathbf{g}}_t(x_k) \leftarrow \mathbf{g}_t(x_k) / \max\left(1, \frac{||\mathbf{g}_t(x_k)||_2}{C}\right)$
    $\tilde{\mathbf{g}}_t(x_k) \leftarrow \frac{1}{B}\left(\sum_i \bar{\mathbf{g}}_t(x_k) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$
    $\theta_{(i,t+1)} \leftarrow \theta_{(i,t)} - \eta_t \tilde{\mathbf{g}}_t(x_k)$
end for
```

FPFL: Framework

The θ_i's from each agent are communicated to the aggregator for further performance improvement. The aggregator takes a weighted sum of the individual θ_i's and broadcasts it to the agents. The agents further train on top of the aggregated model before sending it to the aggregator. This process gets repeated and then repeated.

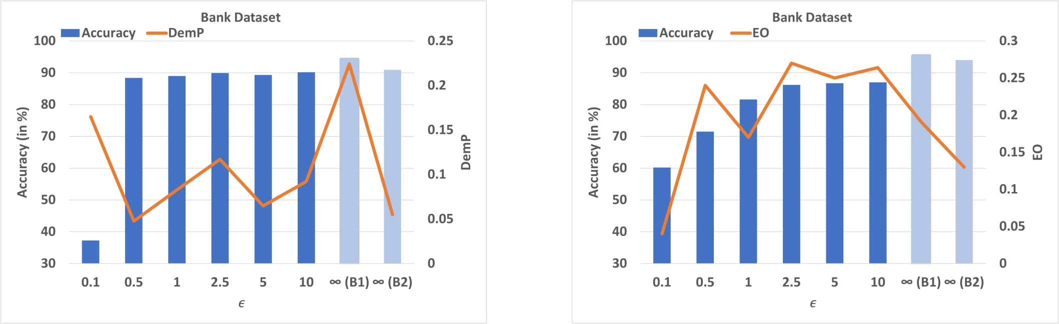
We now couple these processes above to present the FPFL framework with Figure 8.2. The framework presents itself as a plug-and-play system, i.e., a user can use any other loss function instead of L_1, L_2, or change the underlying algorithms for fairness and DP, or do any other tweak it so desires.



(a) Demographic Parity (DemP)

(b) Equalized Odds (EO)

Figure 8.3: Three-way trade-off for the Adult dataset



(a) Demographic Parity (DemP)

(b) Equalized Odds (EO)

Figure 8.4: Three-way trade-off for the Bank dataset

8.5 FPFL: Differential Privacy Bounds

We now present the theoretical guarantees provided by FPFL. Observe that the model learned in Phase 1, h^{ϕ}, requires access to both the training data ($\mathcal{X}$) and the sensitive attribute ($\mathcal{A}$). Fortunately, this phase is entirely independent of the FL aggregation process. In contrast, the model learned in Phase 2, h^{θ} – trained to mimic the predictions of h^{ϕ} – is communicated to the aggregator.

As a result, any information leak in FPFL may take place in the following two ways. Firstly, training data may get compromised through h^{θ}. Secondly, mimicking the predictions from h^ϕ may, in turn, leak information about the sensitive attribute. To this end, we observe that the DP guarantee for the training data follows from [1, Theorem 1] directly. Further, the following proposition proves that the training process in Phase 2 does not leak any additional information regarding $\mathcal{A}$ to the aggregator.

Proposition 8.1. *With the differentially private FPFL framework (Figure 8.2), the aggregator with access to the model h^θ learns no additional information, over the DP guarantee, regarding the sensitive attribute $\mathcal{A}$.*

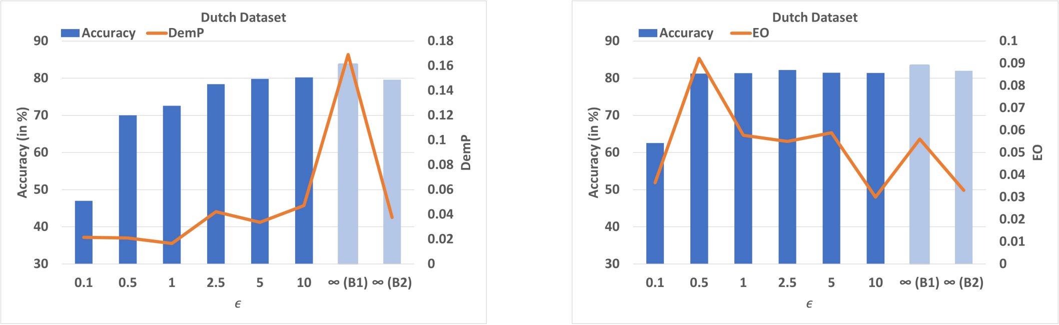
Proof. From [189], we note that a model, i.e., h^θ, can leak information of the sensitive attribute if the training process involves any step which uses the attribute. E.g., to compute the loss incurred at each timestep. Since h^θ does not involve such a step, we can rule out any direct leak.

However, the Fair-SGD model parameters given by ϕ are used in the loss function L_2 for training h^θ. In order for an adversary to gain information about $\mathcal{A}$, it should be able to derive ϕ from θ. Since we add controlled noise during the training for θ, the model is diferentially private w.r.t. ϕ. In addition, as the predictions from ϕ, i.e., $h^\phi(x)$ are private, the DP guarantee protects both $\mathcal{X}$ and $\mathcal{A}$. □

Corollary 8.1. *For the FPFL framework (Figure 8.2), $\forall i \in \mathbb{A}$ there exists constants c_1 and c_2, with the sampling probability $q_i = B_i/\mathcal{X}_i$ and the total number of timesteps T in Phase 2, such that for any $\epsilon_i < c_1 q_i^2 T$, the framework satisfies (ϵ_i, δ_i)-LDP for $\delta_i > 0$ and for*

$$\sigma_i \geq c_2 \frac{q_i \sqrt{T \ln(1/\delta_i)}}{\epsilon_i}.$$

Proof. The result follows from Proposition 8.1 and [1, Theorem 1]. □



(a) Demographic Parity (DemP)

(b) Equalized Odds (EO)

Figure 8.5: Three-way trade-off for the Dutch dataset

8.6 Implementation Details and Experiment Analysis

Datasets. We conduct experiments on the following three datasets: Adult [4], Bank [22] and Dutch [205]. The first two have $\approx 40k$ samples, while the Dutch dataset has $\approx 60k$. In the Adult dataset, the task is a binary prediction of whether an individual's income is above or below USD 50000. The sensitive attribute is *gender* and is available as either male or female, i.e., $|\mathcal{A}| = 2$. In the Bank dataset, the task is to predict if an agent has subscribed to the term deposit or not. In this case, we consider *age* as the sensitive attribute. We group the samples such that $|\mathcal{A}| = 2$. People between the ages 25 to 60 form the majority group, and those under 25 or over 60 form the minority group. In the Dutch dataset, similar to Adult, we consider *gender* as the binary sensitive attribute. The task is to predict the occupation. For training an FL model, we split the datasets such that each agent has an equal number of samples. In order to do so, we duplicate the samples in the existing data – especially the minority group – to get exactly $50k$ samples for the first two datasets. Dutch dataset is relatively well balanced. We hold 20% of the data from each dataset as the test set for which we provide our results.

Hyperparameters. For each agent, we train two fully connected neural networks having the same architecture. Each network has two hidden layers with $(500, 100)$ neurons and

ReLU activation. For DemP, we consider 5 agents in our experiments and split datasets accordingly. To estimate EO, we need sufficient samples for both sensitive groups such that each group has enough samples with both the possible outcomes. In the Adult dataset, we find only 3% female samples earning above USD 50000. Similarly, in the Bank dataset, the minority group that has subscribed to the term deposit forms only 1% of the entire data. Due to this, in our experiments for EO, we consider only 2 agents.

Training Fair-SGD (Phase 1). For training we use Algorithm 10, with $\eta = 0.001$ and $B = 500$. The optimizer used is Adam for updating the loss using the Lagrangian multiplier method. For the Adult dataset, we initialize with $\lambda = 10$, and for the Bank and Dutch datasets, we initialize with $\lambda = 5$. The model is trained for 200 epochs.

Training DP-SGD (Phase 2). For training we use Algorithm 11, with $\eta = 0.25$, $B = 500$, and the clipping norm $C = 1.5$. For the optimizer we use the Tensorflow-privacy library's Keras DP-SGD optimizer[3]. We train the model in this phase for 20 epochs.

Baselines. To compare the resultant three-way trade-off with FPFL, we create the following two baselines.

B1 In this, we let the agents train the model only for maximizing accuracy without any fairness constraints in the loss.

B2 To obtain B2, each agent trains the model for both accuracy and fairness using Algorithm 10 with DemP loss (8.5) or EO loss (8.6).

For both B1 and B2, the final aggregated model is used to report the results. These baselines maximize accuracy and ensure fairness without any privacy guarantee. This lack of a privacy guarantee implies that for both baselines, we skip Phase 2.

(ϵ, δ)-bounds. We calculate the bounds as defined in Corollary 8.1. That is, we plot the three-way trade-off for an agent's ϵ, δ bound. To remain consistent with the broad DP-ML literature, we vary ϵ in the range $(0, 10]$ by appropriately selecting σ (noise multiplier).

[3] https://github.com/tensorflow/privacy

Observe that $\epsilon \to \infty$ for our baselines B1 and B2. This is because the sensitivity in these cases is not bounded. As standard $\forall i \in \mathbb{A}$, we keep $\delta = 10^{-4} < 1/|\mathcal{X}_i|$ for DemP and $\delta = 0.5 \times 10^{-4} < 1/|\mathcal{X}_i|$ for EO.

DemP and EO. When the loss for DemP (Equation 8.5) and EO (Equation 8.6) is exactly zero, the model is perfectly fair. As perfect fairness is impossible, we try to minimize the loss. In our results, to quantify the fairness guarantees, we plot l_{DemP} and l_{EO} on the test set. *Lower* the values, *better* is the guarantee. For readability we refer l_{DemP} and l_{EO} as DemP and EO in our results.

Demographic Parity: Figures 8.3(a), 8.4(a), and 8.5(a)

We consider an FL setting with 5 agents for ensuring DemP. For the Adult dataset, Figure 8.3(a), we find that for B1, we get an accuracy of 87% and a DemP of 0.17. We observe that a model trained with fairness constraints, i.e., for B2, has a reduced accuracy of 85%, but DemP reduces to 0.029. We find similar trends in the baselines for the Bank (Figure 8.4(a)) and the Dutch datasets (Figure 8.5(a)).

Introducing privacy guarantees with FPFL, we observe a further comprise in either accuracy and fairness as compared to our baselines. In general, with increasing ϵ, i.e., increasing privacy loss, there is an improvement in the trade-off of accuracy and DemP. For $\epsilon = 10$, the accuracy and DemP are similar to that in B2. While the drop in accuracy is consistent with decrease in ϵ, DemP values do not always follow this trend.

Equilized Odds: Figures 8.3b, 8.4b, and 8.5b

For EO, we consider FL setting with only 2 agents. From Figure 8.3(b), we find in B1 the accuracy is 87% for the Adult dataset with EO as 0.104. With B2, we obtain reduced accuracy of 80%, but EO reduces to 0.008. We find similar trends in the baselines for the Bank (Figure 8.4(b)) and the Dutch datasets (Figure 8.5(b)).

When we compare the FPFL training, which also guarantees privacy, we observe a trade-off in fairness and accuracy. We note that ensuring EO, especially in the Bank dataset, is very challenging. Therefore, the trade-off is not as smooth. With decrease in ϵ, the accuracy decreases, but the EO values do not follow any trend. We believe this is due to the lack of distinct samples for each sub-group after splitting the data (despite duplication) for FL.

Remark. In our experiments, we obtain privacy, fairness, and accuracy at a cost to each other. With FPFL framework, a user can customize λ and σ to ensure the desired three-way trade-off. The framework allows the use of any fairness measure of choice by appropriately modifying the loss in Phase 1. Exploring these on other relevant datasets and exploring other fairness and privacy techniques in the FPFL framework is left for future work.

8.7 Conclusion

We presented FPFL: a framework that learns fair and accurate models while preserving privacy. A first, we provided a DP guarantee for the training data and sensitive attributes in an FL setting. We then applied FPFL on the Adult, Bank, and Dutch datasets to highlight the relation between accuracy, fairness, and privacy of an FL model.

Chapter 9

Conclusion and Future Work

In this work, we focused on ensuring fairness in existing AI systems. We explored both the notions of i) individual fairness in **Part A** – FAIR ALLOCATIONS WITH STRATEGIC AGENTS. and ii) group fairness in **Part B** – FAIR DECISIONS FOR GROUPS. We believe by shifting the focus from standard performance measures and ensuring other measures related to fairness, we make the existing AI systems more inclusive and user friendly.

Part A – FAIR ALLOCATIONS WITH STRATEGIC AGENTS. We considered the setting of resource allocation, with multiple items and multiple strategic agents who have preferences for these items. The agents may manipulate their preferences to obtain higher gains. We design strategy-proof mechanisms both with payments and without payments for the following scenarios. We design neural-networks to learn payments rules in the setting of i) *redistribution mechanisms* and ii) *multi-armed bandit based mechanisms*. The learnt redistribution mechanism is DSIC and satisfies allocative efficiency while ensuring maximum refund compared to the existing approaches. The learnt MAB-based mechanism for expertsourcing ensures WP-DSIC while considering the agents and the auctioneer's individual rationality. Further, we consider mechanisms without payments for *fair resource allocation*. We analyze the existence of strategy-proof mechanisms without money to ensure allocations that satisfy fairness notions like envy-freeness, proportionality and max-min share allocations. Such notions ensure the individual fairness of the agents involved.

Part B – Fair Decisions for Groups. We considered machine learning-based classification algorithms, which are shown to be biased towards certain demographic groups. We proposed FNNC an end-to-end neural network-based framework to ensure fair classification. We show that with FNNC, we can strike a desirable balance between accuracy and fairness measures like demographic parity or equalized odds on existing real-world datasets. Moving a step ahead, we present FPFL: a framework that learns fair and accurate models while preserving privacy. A first, we provided a DP guarantee for the training data and sensitive attributes in a federated learning setting.

Future Work

- **Scalability and Explanaibility.** In most of our contributions, we propose data-driven approaches. *Scalability* and *Explanability* of such models is still unresolved and requires rigorous research. In future, we would like to extend the existing architectures for larger input sizes, rather make it independent of input sizes by using convolutional network architectures. We would also like to look at the explainability of such models, giving us better insight into the representations for fairness measures versus the representations for other performance measures.

- **Complex Settings.** In the real world, the settings are typically more complex, with agents having a variety of valuations. We aim to provide mechanisms that are deployable in the real world with good empirical performances. Towards this, we must conduct experiments on real-world auction data or for complex valuation structures. Further, the proposed frameworks for group fairness, FNNC and FPFL, have to be extended for data with multiple sensitive attributes, each having multiple categories. It is interesting to study and, if required, extend FNNC and FPFL to settings when the data is heavily skewed and imbalanced.

- **Theoretical Guarantees.** Similar to generalization bounds we have provided for FNNC, we aim to provide the same for other mechanisms we have proposed. Further, it is interesting to have convergence bounds for fairness measures in federated learning settings while also providing DP-guarantees.

Bibliography

[1] 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*. 2016, pp. 308–318.

[2] Kumar Abhishek, Shweta Jain, and Sujit Gujar. "Designing Truthful Contextual Multi-Armed Bandits Based Sponsored Search Auctions". In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '20. Auckland, New Zealand: International Foundation for Autonomous Agents and Multiagent Systems, 2020, pp. 1732–1734. ISBN: 9781450375184.

[3] Peter Adby. *Introduction to optimization methods*. Springer Science & Business Media, 2013.

[4] *Adult income dataset*. https://www.kaggle.com/wenruliu/adult-income-dataset. 2016.

[5] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudik, John Langford, and Hanna Wallach. "A Reductions Approach to Fair Classification". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, 2018, pp. 60–69. URL: http://proceedings.mlr.press/v80/agarwal18a.html.

[6] Shipra Agrawal and Navin Goyal. "Analysis of Thompson Sampling for the multi-armed bandit problem". In: *CoRR* abs/1111.1797 (2011). URL: http://arxiv.org/abs/1111.1797.

[7] *AI Market Share*. https://menafn.com/1104686471/AI-Edge-Computing-Market-Top-Key-Players-Industry-Growth-Analysis-Forecast-2030. 2020.

[8] *AI Ubiqitous*. https://becominghuman.ai/. 2020.

[9] Georgios Amanatidis, Georgios Birmpas, George Christodoulou, and Evangelos Markakis. "Truthful allocation mechanisms without payments: Characterization and implications on fairness". In: *Proceedings of the 2017 ACM Conference on Economics and Computation*. 2017, pp. 545–562.

[10] Georgios Amanatidis, Georgios Birmpas, and Evangelos Markakis. "On truthful mechanisms for maximin share allocations". In: *arXiv preprint arXiv:1605.04026* (2016).

[11] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al. "Privacy-preserving deep learning via additively homomorphic encryption". In: *IEEE Transactions on Information Forensics and Security* 13.5 (2017), pp. 1333–1345.

[12] *Applications*. https://www.google.com/imghp?hl=en. 2020.

[13] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. "Finite-time Analysis of the Multiarmed Bandit Problem". In: *Machine Learning* 47.2-3 (May 2002), pp. 235–256. ISSN: 0885-6125. DOI: 10.1023/A:1013689704352. URL: http://dx.doi.org/10.1023/A:1013689704352.

[14] Haris Aziz, Hervé Moulin, and Fedor Sandomirskiy. "A polynomial-time algorithm for computing a Pareto optimal and almost proportional allocation". In: *Operations Research Letters* 48.5 (2020), pp. 573–578.

[15] Haris Aziz, Gerhard Rauchecker, Guido Schryen, and Toby Walsh. "Algorithms for max-min share fair allocation of indivisible chores". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 2017.

[16] Moshe Babaioff, Shaddin Dughmi, Robert Kleinberg, and Aleksandrs Slivkins. "Dynamic pricing with limited supply". In: *Thirteenth ACM Conference on Electronic Commerce*. ACM, 2012, pp. 74–91. ISBN: 978-1-4503-1415-2. DOI: 10.1145/2229012.2229023. URL: http://doi.acm.org/10.1145/2229012.2229023.

[17] Moshe Babaioff, Tomer Ezra, and Uriel Feige. "Fair and truthful mechanisms for dichotomous valuations". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2021, pp. 5119–5126.

[18] Moshe Babaioff, Robert D. Kleinberg, and Aleksandrs Slivkins. "Truthful mechanisms with implicit payment computation". In: *Eleventh ACM Conference on Electronic Commerce*. ACM, 2010, pp. 43–52.

[19] Moshe Babaioff, Yogeshwer Sharma, and Aleksandrs Slivkins. "Characterizing truthful multi-armed bandit mechanisms: extended abstract". In: *Tenth ACM Conference on Electronic Commerce*. ACM, 2009, pp. 79–88.

[20] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. "Differential privacy has disparate impact on model accuracy". In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 15479–15488.

[21] Martin J Bailey. "The Demand Revealing Process: To Distribute the Surplus". In: *Public Choice* 91.2 (1997), pp. 107–26. URL: https://EconPapers.repec.org/RePEc:kap:pubcho:v:91:y:1997:i:2:p:107-26.

[22] *Bank Loan Status Dataset*. https://www.kaggle.com/zaurbegiev/my-dataset. 2017.

[23] Siddharth Barman and Paritosh Verma. “Truthful and fair mechanisms for matroid-rank valuations”. In: *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 36. 2022, pp. 4801–4808.

[24] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. http://www.fairmlbook.org. fairmlbook.org, 2019.

[25] Solon Barocas and Andrew D Selbst. “Big data’s disparate impact”. In: *Cal. L. Rev.* 104 (2016), p. 671.

[26] Peter L Bartlett and Shahar Mendelson. “Rademacher and Gaussian complexities: Risk bounds and structural results”. In: *Journal of Machine Learning Research* 3.Nov (2002), pp. 463–482.

[27] Yahav Bechavod and Katrina Ligett. “Learning Fair Classifiers: A Regularization-Inspired Approach”. In: *CoRR* abs/1707.00044 (2017).

[28] Xiaohui Bei, Ning Chen, Guangda Huzhang, Biaoshuai Tao, and Jiajun Wu. “Cake Cutting: Envy and Truth”. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17.* 2017, pp. 3625–3631. DOI: 10.24963/ijcai.2017/507. URL: https://doi.org/10.24963/ijcai.2017/507.

[29] Kristóf Bérczi, Erika R Bérczi-Kovács, Endre Boros, Fekadu Tolessa Gedefa, Naoyuki Kamiyama, Telikepalli Kavitha, Yusuke Kobayashi, and Kazuhisa Makino. “Envy-free relaxations for goods, chores, and mixed items”. In: *arXiv preprint arXiv:2006.04428* (2020).

[30] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. “Fairness in criminal justice risk assessments: The state of the art”. In: *Sociological Methods & Research* (2018), p. 0049124118782533.

[31] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed Huai-hsin Chi. “Data Decisions and Theoretical Implications when Adversarially Learning Fair Representations”. In: *CoRR* abs/1707.00075 (2017).

[32] Satyanath Bhat, Shweta Jain, Sujit Gujar, and Yadati Narahari. "An Optimal Bidimensional Multi-Armed Bandit Auction for Multi-unit Procurement". In: *Fourteenth International Conference on Autonomous Agents and Multiagent Systems, AAMAS'15*. 2015, pp. 1789–1790.

[33] Satyanath Bhat, Swaprava Nath, Onno Xoeter, Sujit Gujar, Yadati Narahari, and Chris Dance. "A Mechanism to Optimally Balance Cost and Quality of Labeling Tasks Outsourced to Strategic Agents". In: *Thirtheenth International Conference on Autonomous Agents and Multiagent Systems*. 2014, pp. 917–924.

[34] *Bias in ML*. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing. 2020.

[35] *Bias in ML*. https://www.infoq.com/presentations/unconscious-bias-machine-learning/. 2022.

[36] M. Bilal Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi. "Fairness Constraints: Mechanisms for Fair Classification". In: *ArXiv e-prints* (July 2015). arXiv: 1507.05259 [stat.ML].

[37] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. Springer, 2006.

[38] Léon Bottou et al. "Online learning and stochastic approximations". In: *On-line learning in neural networks* 17.9 (1998), p. 142.

[39] Léon Bottou and Olivier Bousquet. "Learning using large datasets". In: *Mining Massive Data Sets for Security*. IOS Press, 2008, pp. 15–26.

[40] Sylvain Bouveret and Jérome Lang. "A General Elicitation-Free Protocol for Allocating Indivisible Goods". In: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One*. IJCAI'11. Barcelona, Catalonia, Spain: AAAI Press, 2011, pp. 73–78. ISBN: 9781577355137.

[41] Sylvain Bouveret and Michel Lemaıtre. “Characterizing conflicts in fair division of indivisible goods using a scale of criteria”. In: *Autonomous Agents and Multi-Agent Systems* 30.2 (2016), pp. 259–290.

[42] Simina Branzei, Vasilis Gkatzelis, and Ruta Mehta. “Nash social welfare approximation for strategic agents”. In: *Proceedings of the 2017 ACM Conference on Economics and Computation*. 2017, pp. 611–628.

[43] Eric Budish. “The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes”. In: *Journal of Political Economy* 119.6 (2011), pp. 1061–1103.

[44] *Cake Cutting*. https://www.jioforme.com/a-new-ai-framework-introduced-to-cut-multi-layer-cakes/349363/. 2020.

[45] Toon Calders and Sicco Verwer. “Three naive Bayes approaches for discrimination-free classification”. In: *Data Mining and Knowledge Discovery* 21 (2010), pp. 277–292.

[46] 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*. FAT* ’19. Atlanta, GA, USA: Association for Computing Machinery, 2019, pp. 309–318. ISBN: 9781450361255. DOI: 10.1145/3287560.3287561. URL: https://doi.org/10.1145/3287560.3287561.

[47] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. “The unreasonable fairness of maximum Nash welfare”. In: *ACM Transactions on Economics and Computation (TEAC)* 7.3 (2019), pp. 1–32.

[48] Ruggiero Cavallo. “Optimal Decision-Making With Minimal Waste: Strategyproof Redistribution of VCG Payments”. In: *Proc. of the 5th Int. Joint Conf. on Au-*

tonomous Agents and Multi Agent Systems (AAMAS'06). Hakodate, Japan, 2006, pp. 882–889. URL: http://econcs.seas.harvard.edu/files/econcs/files/cavallo-redis.pdf.

[49] Olivier Chapelle and Lihong Li. “An Empirical Evaluation of Thompson Sampling”. In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger. Curran Associates, Inc., 2011, pp. 2249–2257. URL: http://papers.nips.cc/paper/4321-an-empirical-evaluation-of-thompson-sampling.pdf.

[50] Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. “EFX exists for three agents”. In: *Proceedings of the 21st ACM Conference on Economics and Computation*. 2020, pp. 1–19.

[51] Changyao Chen. *Platt scaling for probability calibration*. 2018. URL: https://changyaochen.github.io/platt-scaling/.

[52] Yiling Chen, John K Lai, David C Parkes, and Ariel D Procaccia. “Truth, justice, and cake cutting”. In: *Games and Economic Behavior* 77.1 (2013), pp. 284–297.

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

[54] Edward Clarke. “Multipart pricing of public goods”. In: *Public Choice* 11.1 (1971), pp. 17–33. URL: https://EconPapers.repec.org/RePEc:kap:pubcho:v:11:y:1971:i:1:p:17-33.

[55] Geoffroy de Clippel, Victor Naroditskiy, Maria Polukarov, Amy Greenwald, and Nicholas R. Jennings. “Destroy to save”. In: *Games and Economic Behavior* 86.C (2014), pp. 392–404.

[56] Edith Cohen, Michal Feldman, Amos Fiat, Haim Kaplan, and Svetlana Olonetsky. “Truth, Envy, and Truthful Market Clearing Bundle Pricing”. In: *Internet and Net-*

work Economics. Ed. by Ning Chen, Edith Elkind, and Elias Koutsoupias. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 97–108. ISBN: 978-3-642-25510-6.

[57] Richard Cole, Vasilis Gkatzelis, and Gagan Goel. "Mechanism design for fair division: allocating divisible items without payments". In: *Proceedings of the fourteenth ACM conference on Electronic commerce*. 2013, pp. 251–268.

[58] Vincent Conitzer, Rupert Freeman, and Nisarg Shah. "Fair public decision making". In: *Proceedings of the 2017 ACM Conference on Economics and Computation*. 2017, pp. 629–646.

[59] Vincent Conitzer and Tuomas Sandholm. "Automated mechanism design for a self-interested designer". In: *Proceedings of the 4th ACM conference on Electronic commerce*. 2003, pp. 232–233.

[60] Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. "On the compatibility of privacy and fairness". In: *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*. 2019, pp. 309–315.

[61] George Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.

[62] D. Garg, Y. Narahari, and S. Gujar. "Foundations of mechanism design: A tutorial - Part 1: Key Concepts and Classical Results". In: *Sadhana - Indian Academy Proceedings in Engineering Sciences* 33.2 (2008), pp. 83–120.

[63] Y. Narahari D. Garg and S. Gujar. "Foundations of mechanism design: A tutorial - Part 2: Advanced Concepts and Results". In: *Sadhana - Indian Academy Proceedings in Engineering Sciences* 33.2 (2008), pp. 121–174.

[64] Sankarshan Damle, Aleksei Triastcyn, Boi Faltings, and Sujit Gujar. "Differentially Private Multi-Agent Constraint Optimization". In: *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. WI-IAT '21. Melbourne, VIC, Australia: Association for Computing Machinery, 2022, pp. 422–429.

ISBN: 9781450391153. DOI: 10.1145/3486622.3493929. URL: https://doi.org/10.1145/3486622.3493929.

[65] Shantanu Kumar Das, Swapnil Dhamal, Ganesh Ghalme, Shweta Jain, and Sujit Gujar. “Individual Fairness in Feature-Based Pricing for Monopoly Markets”. In: *Conference on Uncertainty in Artificial Intelligence*. 2022.

[66] Nikhil R. Devanur and Sham M. Kakade. “The price of truthfulness for pay-per-click auctions”. In: *Tenth ACM Conference on Electronic Commerce*. 2009, pp. 99–106.

[67] Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong. “Fairness-aware Agnostic Federated Learning”. In: *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM. 2021, pp. 181–189.

[68] John Duchi, Elad Hazan, and Yoram Singer. “Adaptive subgradient methods for online learning and stochastic optimization.” In: *Journal of machine learning research* 12.7 (2011).

[69] Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David Parkes, and Sai Srivatsa Ravindranath. “Optimal auctions through deep learning”. In: *International Conference on Machine Learning*. 2019, pp. 1706–1715.

[70] Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David C Parkes, and Sai Srivatsa Ravindranath. “Optimal auctions through deep learning”. In: *arXiv preprint arXiv:1706.03459* (2017).

[71] 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*. ITCS ’12. Cambridge, Massachusetts: ACM, 2012, pp. 214–226. ISBN: 978-1-4503-1115-1. DOI: 10.1145/2090236.2090255.

[72] Cynthia Dwork, Aaron Roth, et al. “The algorithmic foundations of differential privacy.” In: *Foundations and Trends in Theoretical Computer Science* 9.3-4 (2014), pp. 211–407.

[73] Elad ET Eban, Mariano Schain, Alan Mackey, Ariel Gordon, Rif A Saurous, and Gal Elidan. “Scalable learning of non-decomposable objectives”. In: *arXiv preprint arXiv:1608.04802* (2016).

[74] Harrison Edwards and Amos Storkey. “Censoring Representations with an Adversary”. English. In: *International Conference in Learning Representations (ICLR2016)*. Feb. 2016.

[75] Boi Faltings. “A Budget-balanced, Incentive-compatible Scheme for Social Choice”. In: *Proceedings of the 6th AAMAS International Conference on Agent-Mediated Electronic Commerce: Theories for and Engineering of Distributed Mechanisms and Systems*. AAMAS’04. New York, NY: Springer-Verlag, 2005, pp. 30–43. DOI: 10.1007/11575726_3. URL: http://dx.doi.org/10.1007/11575726_3.

[76] Haokun Fang and Quan Qian. “Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning”. In: *Future Internet* 13.4 (2021), p. 94.

[77] 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*. KDD ’15. Sydney, NSW, Australia: ACM, 2015, pp. 259–268. ISBN: 978-1-4503-3664-2. DOI: 10.1145/2783258.2783311. URL: http://doi.acm.org/10.1145/2783258.2783311.

[78] Duncan Karl Foley. *Resource allocation and the public sector*. Yale University, 1966.

[79] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. “Model inversion attacks that exploit confidence information and basic countermeasures”. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 2015, pp. 1322–1333.

[80] Ghalme Ganesh, Jain Shweta, Gujar Sujit, and Narahari Y. “A Deterministic MAB Mechanism for Crowdsourcing with Logarithmic Regret and Immediate Payments”. In: *Proceedings of Fifteenth International Confrence on Autonomous Agents Multi-Agent Systems*. AAMAS’16. 2016, To Appear.

[81] Simson Garfinkel, John M Abowd, and Christian Martindale. “Understanding data base reconstruction attacks on public data”. In: *Communications of the ACM* 62.3 (2019), pp. 46–53.

[82] Nicola Gatti, Alessandro Lazaric, and Francesco Trovò. “A truthful learning mechanism for contextual multi-slot sponsored search auctions with externalities”. In: *Thirteenth ACM Conference on Electronic Commerce*. 2012, pp. 605–622.

[83] Ganesh Ghalme, Shweta Jain, Sujit Gujar, and Y. Narahari. “Thompson Sampling Based Mechanisms for Stochastic Multi-Armed Bandit Problems”. In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. AAMAS ’17. São Paulo, Brazil: International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 87–95.

[84] Yiannis Giannakopoulos and Elias Koutsoupias. “Duality and optimality of auctions for uniform distributions”. In: *Proceedings of the fifteenth ACM conference on Economics and computation*. 2014, pp. 259–276.

[85] Allan Gibbard. “Manipulation of voting schemes: a general result”. In: *Econometrica: journal of the Econometric Society* (1973), pp. 587–601.

[86] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 2010, pp. 249–256. URL: http://proceedings.mlr.press/v9/glorot10a.html.

[87] Naman Goel and Boi Faltings. "Personalized peer truth serum for eliciting multi-attribute personal data". In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 18–27.

[88] Andrew V. Goldberg and Jason D. Hartline. "Envy-Free Auctions for Digital Goods". In: *Proceedings of the 4th ACM Conference on Electronic Commerce*. EC '03. San Diego, CA, USA: Association for Computing Machinery, 2003, pp. 29–35. ISBN: 158113679X. DOI: 10.1145/779928.779932. URL: https://doi.org/10.1145/779928.779932.

[89] Noah Golowich, Harikrishna Narasimhan, and David C Parkes. "Deep Learning for Multi-Facility Location Mechanism Design." In: *IJCAI*. 2018, pp. 261–267.

[90] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[91] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[92] Alex Graves. "Generating Sequences With Recurrent Neural Networks". In: *CoRR* abs/1308.0850 (2013).

[93] Jerry R. Green and Jean-Jacques Laffont. *Incentives in Public Decision Making*. Amsterdam: North-Holland, 1979.

[94] Theodore Groves. "Incentives in Teams". In: *Econometrica* 41.4 (1973), pp. 617–31. URL: https://EconPapers.repec.org/RePEc:ecm:emetrp:v:41:y:1973:i:4:p:617-31.

[95] Sujit Gujar and Y. Narahari. "Redistribution Mechanisms for Assignment of Heterogeneous Objects". In: *J. Artif. Int. Res.* 41.2 (May 2011), pp. 131–154. ISSN: 1076-9757.

[96] Faruk Gul and Ennio Stacchetti. “Walrasian Equilibrium with Gross Substitutes”. In: *Journal of Economic Theory* 87.1 (1999), pp. 95–124. URL: https://EconPapers.repec.org/RePEc:eee:jetheo:v:87:y:1999:i:1:p:95-124.

[97] Mingyu Guo. “An asymptotically optimal VCG redistribution mechanism for the public project problem”. In: *Autonomous Agents and Multi-Agent Systems* 35.2 (2021), pp. 1–27.

[98] Mingyu Guo. “VCG Redistribution with Gross Substitutes”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 25.1 (2011), pp. 675–680. URL: https://ojs.aaai.org/index.php/AAAI/article/view/7883.

[99] Mingyu Guo. “Worst-case Optimal Redistribution of VCG Payments in Heterogeneous item Auctions with Unit Demand”. In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*. AAMAS ’12. Valencia, Spain: International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 745–752.

[100] Mingyu Guo and Vincent Conitzer. “Better Redistribution with Inefficient Allocation in Multi-unit Auctions with Unit Demand”. In: *Proceedings of the 9th ACM Conference on Electronic Commerce*. EC ’08. Chicago, Il, USA: ACM, 2008, pp. 210–219. ISBN: 978-1-60558-169-9. DOI: 10.1145/1386790.1386825. URL: http://doi.acm.org/10.1145/1386790.1386825.

[101] Mingyu Guo and Vincent Conitzer. “Optimal-in-expectation Redistribution Mechanisms”. In: *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*. AAMAS ’08. Estoril, Portugal: International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 1047–1054. ISBN: 978-0-9817381-1-6.

[102] Mingyu Guo and Vincent Conitzer. “Worst-case Optimal Redistribution of VCG Payments”. In: *Proceedings of the 8th ACM Conference on Electronic Commerce*.

EC ’07. San Diego, California, USA: ACM, 2007, pp. 30–39. ISBN: 978-1-59593-653-0.

[103] Mingyu Guo and Vincent Conitzer. “Worst-case optimal redistribution of VCG payments in multi-unit auctions”. In: *Games and Economic Behavior* 67.1 (2009), pp. 69–98.

[104] Daniel Halpern, Ariel D Procaccia, Alexandros Psomas, and Nisarg Shah. “Fair division with binary valuations: One rule to rule them all”. In: *International Conference on Web and Internet Economics*. Springer. 2020, pp. 370–383.

[105] Moritz Hardt, Eric Price, and Nathan Srebro. “Equality of Opportunity in Supervised Learning”. In: *NIPS*. 2016.

[106] Jason D. Hartline and Tim Roughgarden. “Optimal Mechanism Design and Money Burning”. In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*. STOC ’08. Victoria, British Columbia, Canada: ACM, 2008, pp. 75–84. ISBN: 978-1-60558-047-0. DOI: 10.1145/1374376.1374390. URL: http://doi.acm.org/10.1145/1374376.1374390.

[107] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[108] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Identity Mappings in Deep Residual Networks”. In: *CoRR* abs/1603.05027 (2016). arXiv: 1603.05027. URL: http://arxiv.org/abs/1603.05027.

[109] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. “Distilling the knowledge in a neural network”. In: *arXiv preprint arXiv:1503.02531* (2015).

[110] Kurt Hornik. “Approximation Capabilities of Multilayer Feedforward Networks”. In: *Neural Networks* 4 (1991), pp. 251–257. DOI: 10.1016/0893-6080(91)90009-T.

[111] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural networks* 4.2 (1991), pp. 251–257.

[112] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5 (1989), pp. 359–366.

[113] Leonid Hurwicz. "On Informationally Decentralized Systems". In: *Decision and Organization: A Volume in Honor of Jacob Marschak*. Ed. by C. B. McGuire and Roy Radner. Amsterdam: North-Holland, 1972.

[114] Leonid Hurwicz. "Optimality and Informational Efficiency in Resource Allocation Processes". In: *Mathematical Methods in the Social Sciences*. Ed. by Kenneth J. Arrow, Samuel Karlin, and Patrick Suppes. Stanford, CA: Stanford University Press, 1959.

[115] Leonid Hurwicz. "The design of mechanisms for resource allocation". In: *The American Economic Review* 63.2 (1973), pp. 1–30.

[116] Shweta Jain, Sujit Gujar, Onno Xoeter, and Y. Narahari. "A Quality Assuring Multi-Armed Bandit Crowdsourcing Mechanism with Incentive Compatible Learning". In: *Thirtheenth International Conference on Autonomous Agents and Multiagent Systems*. 2014, pp. 1609–1610.

[117] Shweta Jain, Balakrishnan Narayanaswamy, and Y. Narahari. "A Multiarmed Bandit Incentive Mechanism for Crowdsourcing Demand Response in Smart Grids". In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*. 2014, pp. 721–727. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8355.

[118] Taeuk Jang, Pengyi Shi, and Xiaoqian Wang. "Group-aware threshold adaptation for fair classification". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 2022, pp. 6988–6995.

[119] F. Kamiran and T. Calders. “Classifying without discriminating”. In: *2009 2nd International Conference on Computer, Control and Communication*. 2009, pp. 1–6. DOI: 10.1109/IC4.2009.4909197.

[120] F. Kamiran and T.G.K. Calders. “Classification with no discrimination by preferential sampling”. English. In: *Informal proceedings of the 19th Annual Machine Learning Conference of Belgium and The Netherlands (Benelearn'10, Leuven, Belgium, May 27-28, 2010)*. 2010, pp. 1–6.

[121] Faisal Kamiran, Asim Karim, and Xiangliang Zhang. “Decision theory for discrimination aware classification”. In: *2012 IEEE 12th International Conference on Data Mining*. IEEE. 2012, pp. 924–929.

[122] T. Kamishima, S. Akaho, and J. Sakuma. “Fairness-aware Learning through Regularization Approach”. In: *2011 IEEE 11th International Conference on Data Mining Workshops*. 2011, pp. 643–650. DOI: 10.1109/ICDMW.2011.83.

[123] Samhita Kanaparthy, Manisha Padala, Sankarshan Damle, Ravi Kiran Sarvadevabhatla, and Sujit Gujar. “F3: Fair and Federated Face Attribute Classification With Heterogeneous Data”. In: *Advances in Knowledge Discovery and Data Mining: 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2023, Osaka, Japan, May 25–28, 2023, Proceedings, Part I*. Osaka, Japan: Springer-Verlag, 2023, pp. 483–494. ISBN: 978-3-031-33373-6.

[124] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. “Thompson Sampling: An Asymptotically Optimal Finite-Time Analysis”. In: *Algorithmic Learning Theory - 23rd International Conference, ALT 2012, Lyon, France, October 29-31, 2012. Proceedings*. 2012, pp. 199–213. DOI: 10.1007/978-3-642-34106-9_18. URL: http://dx.doi.org/10.1007/978-3-642-34106-9_18.

[125] Kenji Kawaguchi. “Deep Learning without Poor Local Minima”. In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V.

Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc., 2016, pp. 586–594. URL: http://papers.nips.cc/paper/6112-deep-learning-without-poor-local-minima.pdf.

[126] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *stat* 1050 (2014), p. 10.

[127] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *CoRR* abs/1412.6980 (2014). arXiv: 1412.6980. URL: http://arxiv.org/abs/1412.6980.

[128] David Kurokawa, Ariel D Procaccia, and Junxing Wang. “Fair enough: Guaranteeing approximate maximin shares”. In: *Journal of the ACM (JACM)* 65.2 (2018), pp. 1–27.

[129] M Ledoux. *M. Talagrand Probability in Banach Spaces First Reprint*. 2002.

[130] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. “On approximately fair allocations of indivisible goods”. In: *Proceedings of the 5th ACM conference on Electronic commerce*. 2004, pp. 125–131.

[131] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard S. Zemel. “The Variational Fair Autoencoder”. In: *CoRR* abs/1511.00830 (2015).

[132] David Madras, Elliot Creager, Toniann Pitassi, and Richard S. Zemel. “Learning Adversarially Fair and Transferable Representations”. In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. 2018, pp. 3381–3390.

[133] Debmalya Mandal, Matthew Leifer, David C Parkes, Galen Pickard, and Victor Shnayder. “Peer prediction with heterogeneous tasks”. In: *arXiv preprint arXiv:1612.00928* (2016).

[134] Alejandro M Manelli, Daniel R Vincent, et al. “Bundling as an optimal selling mechanism for a multiple-good monopolist”. In: *Journal of Economic Theory* 127.1 (2006), pp. 1–35.

[135] Padala Manisha and Sujit Gujar. “Thompson Sampling Based Multi-Armed-Bandit Mechanism Using Neural Networks.” In: *AAMAS*. 2019, pp. 2111–2113.

[136] Padala Manisha, C. V. Jawahar, and Sujit Gujar. “Learning Optimal Redistribution Mechanisms Through Neural Networks”. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS ’18. Stockholm, Sweden: International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 345–353. URL: http://dl.acm.org/citation.cfm?id=3237383.3237438.

[137] Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[138] Eric Maskin and JJ Laffont. “A differential approach to expected utility maximizing mechanisms”. In: *Aggregation and revelation of preferences* (1979).

[139] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. “Communication-efficient learning of deep networks from decentralized data”. In: *Artificial Intelligence and Statistics*. PMLR. 2017, pp. 1273–1282.

[140] John McMillan. “Selling spectrum rights”. In: *Journal of Economic Perspectives* 8.3 (1994), pp. 145–162.

[141] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. “A Survey on Bias and Fairness in Machine Learning”. In: *ACM Comput. Surv.* 54.6 (July 2021). ISSN: 0360-0300. DOI: 10.1145/3457607. URL: https://doi.org/10.1145/3457607.

[142] Vijay Menon and Kate Larson. “Deterministic, Strategyproof, and Fair Cake Cutting”. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 2017, pp. 352–358. DOI: 10.24963/ijcai.2017/50. URL: https://doi.org/10.24963/ijcai.2017/50.

[143] Shaily Mishra, Manisha Padala, and Sujit Gujar. “EEF1-NN: Efficient and EF1 Allocations Through Neural Networks”. In: *PRICAI 2022: Trends in Artificial Intelligence: 19th Pacific Rim International Conference on Artificial Intelligence, PRICAI 2022, Shanghai, China, November 10–13, 2022, Proceedings, Part II*. Shangai, China: Springer-Verlag, 2022, pp. 388–401. ISBN: 978-3-031-20864-5.

[144] Payman Mohassel and Yupeng Zhang. “Secureml: A system for scalable privacy-preserving machine learning”. In: *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2017, pp. 19–38.

[145] Elchanan Mossel and Omer Tamuz. “Truthful fair division”. In: *International Symposium on Algorithmic Game Theory*. Springer. 2010, pp. 288–299.

[146] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. “A survey on security and privacy of federated learning”. In: *Future Generation Computer Systems* 115 (2021), pp. 619–640.

[147] Hervé Moulin. *Efficient, strategy-proof and almost budget-balanced assignment*. Tech. rep. March 2007. Working Paper, 2007.

[148] Hervé Moulin. “On strategy-proofness and single peakedness”. In: *Public Choice* 35.4 (1980), pp. 437–455.

[149] Herve Moulin. “Almost budget-balanced VCG mechanisms to assign multiple objects”. In: *Journal of Economic Theory* 144.1 (2009), pp. 96–119.

[150] Hussein Mozannar, Mesrob Ohannessian, and Nathan Srebro. “Fair learning with private demographic data”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 7066–7075.

[151] Yadati Narahari. *Game theory and mechanism design*. Vol. 4. World Scientific, 2014.

[152] Harikrishna Narasimhan. “Learning with Complex Loss Functions and Constraints”. In: *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*. 2018, pp. 1646–1654. URL: http://proceedings.mlr.press/v84/narasimhan18a.html.

[153] Arvind Narayanan and Vitaly Shmatikov. “Robust de-anonymization of large sparse datasets”. In: *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE. 2008, pp. 111–125.

[154] Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. “Toward robustness and privacy in federated learning: Experimenting with local and central differential privacy”. In: *arXiv preprint arXiv:2009.03561* (2020).

[155] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press, 2007. ISBN: 0521872820.

[156] Paul Ohm. “Broken promises of privacy: Responding to the surprising failure of anonymization”. In: *UCLA l. Rev.* 57 (2009), p. 1701.

[157] Manisha Padala, Sankarshan Damle, and Sujit Gujar. “Federated learning meets fairness and differential privacy”. In: *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part VI 28*. Springer. 2021, pp. 692–699.

[158] Manisha Padala and Sujit Gujar. “FNNC: Achieving Fairness through Neural Networks”. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. Ed. by Christian Bessiere. Main track. International Joint Conferences on Artificial Intelligence Organization, July 2020, pp. 2277–2283.

[159] Manisha Padala and Sujit Gujar. "Mechanism Design without Money for Fair Allocations". In: *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology.* WI-IAT '21. Melbourne, VIC, Australia: Association for Computing Machinery, 2022, pp. 382–389. ISBN: 9781450391153.

[160] David C. Parkes. "Online mechanisms". In: *Algorithmic Game Theory, ed. N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani.* New York, NY, USA: Cambridge University Press, 2007, pp. 411–439. ISBN: 0521872820.

[161] David C. Parkes, Jayant R. Kalagnanam, and Marta Eso. "Achieving Budget-Balance with Vickrey-Based Payment Schemes in Exchanges". In: *Proc. 17th International Joint Conference on Artificial Intelligence (IJCAI'01).* 2001, 1161–1168. URL: http://econcs.seas.harvard.edu/files/econcs/files/combexch01.pdf.

[162] Manas A Pathak, Shantanu Rane, and Bhiksha Raj. "Multiparty Differential Privacy via Aggregation of Locally Trained Classifiers." In: *NIPS.* Citeseer. 2010, pp. 1876–1884.

[163] Gregory Pavlov. "Optimal mechanism for selling two goods". In: *The BE Journal of Theoretical Economics* 11.1 (2011).

[164] Benjamin Plaut and Tim Roughgarden. "Almost envy-freeness with general valuations". In: *SIAM Journal on Discrete Mathematics* 34.2 (2020), pp. 1039–1068.

[165] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. "On Fairness and Calibration". In: *Advances in Neural Information Processing Systems 30.* Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 5680–5689.

[166] Boris T Polyak. "Some methods of speeding up the convergence of iteration methods". In: *Ussr computational mathematics and mathematical physics* 4.5 (1964), pp. 1–17.

[167] Ariel D Procaccia and Junxing Wang. "Fair enough: Guaranteeing approximate maximin shares". In: *Proceedings of the 15th ACM conference on Economics and computation*. 2014, pp. 675–692.

[168] Stephen J Rassenti, Vernon L Smith, and Robert L Bulfin. "A combinatorial auction mechanism for airport time slot allocation". In: *The Bell Journal of Economics* (1982), pp. 402–417.

[169] Herbert Robbins. "Some aspects of the sequential design of experiments". In: *Bull. Amer. Math. Soc.* 58.5 (Sept. 1952), pp. 527–535. URL: https://projecteuclid.org:443/euclid.bams/1183517370.

[170] Jean-Charles Rochet. "A necessary and sufficient condition for rationalizability in a quasi-linear context". In: *Journal of mathematical Economics* 16.2 (1987), pp. 191–200.

[171] Tuomas Sandholm. "Automated mechanism design: A new application area for search algorithms". In: *International Conference on Principles and Practice of Constraint Programming*. Springer. 2003, pp. 19–36.

[172] Mark Allen Satterthwaite. "Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions". In: *Journal of economic theory* 10.2 (1975), pp. 187–217.

[173] Data Science and View my complete profile. *Confusion Matrix*. https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html. [Online; accessed 2022-09-29]. 2019.

[174] Shai Shalev-Shwartz and Shai Ben-David. "Rademacher Complexities". In: *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014, pp. 325–336. DOI: 10.1017/CBO9781107298019.027.

[175] Akash Das Sharma, Sujit Gujar, and Y. Narahari. “Truthful multi-armed bandit mechanisms for multi-slot sponsored search auctions”. In: *Current Science* Vol. 103 Issue 9 (2012), pp. 1064–1077.

[176] Weiran Shen, Pingzhong Tang, and Song Zuo. “Automated mechanism design via neural networks”. In: *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*. 2019, pp. 215–223.

[177] Reza Shokri and Vitaly Shmatikov. “Privacy-preserving deep learning”. In: *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. 2015, pp. 1310–1321.

[178] Sambhav Solanki, Samhita Kanaparthy, Sankarshan Damle, and Sujit Gujar. “Differentially Private Federated Combinatorial Bandits with Constraints”. In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Massih-Reza Amini, Stéphane Canu, Asja Fischer, Tias Guns, Petra Kralj Novak, and Grigorios Tsoumakas. Cham: Springer Nature Switzerland, 2023, pp. 620–637. ISBN: 978-3-031-26412-2.

[179] Daniel Soudry and Yair Carmon. “No bad local minima: Data independent training error guarantees for multilayer neural networks”. In: *CoRR* abs/1605.08361 (2016). arXiv: 1605.08361. URL: http://arxiv.org/abs/1605.08361.

[180] H Steihaus. “The problem of fair division”. In: *Econometrica* 16 (1948), pp. 101–104.

[181] Walter Stromquist. “How to cut a cake fairly”. In: *The American Mathematical Monthly* 87.8 (1980), pp. 640–644.

[182] Xin Sui, Craig Boutilier, and Tuomas Sandholm. “Analysis and optimization of multi-dimensional percentile mechanisms”. In: *Twenty-Third International Joint Conference on Artificial Intelligence*. 2013.

[183] Harini Suresh and John Guttag. “Understanding Potential Sources of Harm throughout the Machine Learning Life Cycle”. In: *MIT Case Studies in Social and Ethical Responsibilities of Computing* (Aug. 10, 2021). DOI: 10.21428/2c646de5.c16a07bb. URL: https://mit-serc.pubpub.org/pub/potential-sources-of-harm-throughout-the-machine-learning-life-cycle.

[184] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. “On the importance of initialization and momentum in deep learning”. In: *International conference on machine learning*. PMLR. 2013, pp. 1139–1147.

[185] Andrea Tacchetti, DJ Strouse, Marta Garnelo, Thore Graepel, and Yoram Bachrach. “A neural architecture for designing truthful and efficient auctions”. In: *arXiv preprint arXiv:1907.05181* (2019).

[186] Bo Tang and Jinshan Zhang. “Envy-Free Sponsored Search Auctions with Budgets”. In: *IJCAI*. 2015, pp. 653–659. URL: http://ijcai.org/Abstract/15/098.

[187] William R. Thompson. “On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples”. English. In: *Biometrika* 25.3/4 (1933), pp. 285–294. ISSN: 00063444. URL: http://www.jstor.org/stable/2332286.

[188] Tijmen Tieleman and Geoffrey Hinton. “Lecture 6.5-rmsprop, coursera: Neural networks for machine learning”. In: *University of Toronto, Technical Report* 6 (2012).

[189] Cuong Tran, Ferdinando Fioretto, and Pascal Van Hentenryck. “Differentially Private and Fair Deep Learning: A Lagrangian Dual Approach”. In: *arXiv preprint arXiv:2009.12562* (2020).

[190] W. Vickrey. “Counterspeculation, Auctions, and Competitive Sealed Tenders”. In: *Journal of Finance* 16.1 (1961), pp. 8–37.

[191] William Vickrey. “Counterspeculation, auctions, and competitive sealed tenders”. In: *The Journal of finance* 16.1 (1961), pp. 8–37.

[192] Rakesh V Vohra. *Mechanism design: a linear programming approach.* Vol. 47. Cambridge University Press, 2011.

[193] Sandra Wachter, Brent Mittelstadt, and Chris Russell. “Bias Preservation in Machine Learning: The Legality of Fairness Metrics Under EU Non-Discrimination Law”. In: *West Virginia Law Review, Forthcoming* (2021).

[194] Omar Abdel Wahab, Azzam Mourad, Hadi Otrok, and Tarik Taleb. “Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems”. In: *IEEE Communications Surveys & Tutorials* (2021).

[195] Guanhua Wang, Wuli Zuo, and Mingyu Guo. “Redistribution in Public Project Problems via Neural Networks”. In: *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology.* WI-IAT ’21. Melbourne, VIC, Australia: Association for Computing Machinery, 2022, pp. 406–413. ISBN: 9781450391153.

[196] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. “Federated Learning With Differential Privacy: Algorithms and Performance Analysis”. In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 3454–3469. DOI: 10.1109/TIFS.2020.2988575.

[197] Wikipedia contributors. *Differential privacy — Wikipedia, The Free Encyclopedia.* https://en.wikipedia.org/w/index.php?title=Differential_privacy&oldid=1049555252. [Online; accessed 23-October-2021]. 2021.

[198] Gerhard J Woeginger. “A polynomial-time approximation scheme for maximizing the minimum machine completion time”. In: *Operations Research Letters* 20.4 (1997), pp. 149–154.

[199] Yongkai Wu, Lu Zhang, and Xintao Wu. "Fairness-aware Classification: Criterion, Convexity, and Bounds". In: *CoRR* abs/1809.04737 (2018).

[200] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. "Federated machine learning: Concept and applications". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019), pp. 1–19.

[201] Tatu Ylonen. "SSH–secure login connections over the Internet". In: *Proceedings of the 6th USENIX Security Symposium*. Vol. 37. 1996.

[202] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. "Learning Fair Representations". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research. Atlanta, Georgia, USA: PMLR, 2013, pp. 325–333.

[203] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. "Mitigating Unwanted Biases with Adversarial Learning". In: *CoRR* abs/1801.07593 (2018). arXiv: 1801.07593. URL: http://arxiv.org/abs/1801.07593.

[204] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. "Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning". In: *2020 {USENIX} Annual Technical Conference*. 2020, pp. 493–506.

[205] Indre Žliobaite, Faisal Kamiran, and Toon Calders. "Handling Conditional Discrimination". In: *2011 IEEE 11th International Conference on Data Mining*. 2011, pp. 992–1001. DOI: 10.1109/ICDM.2011.72.