On the Intersection of Communication and Machine Learning

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Yawen Fan

August 2019

© by Yawen Fan, 2019

All Rights Reserved.

Thank you to my academic adviser who guided me in this process and the committee who kept me on track.

Acknowledgments

This dissertation marks the end of an exciting and fruitful journal for the past 6 years. I was blessed to have the opportunity to explore some interesting research fields and tried to make the contribution to them. I would like to express my sincere gratitude to those who have helped me along the way.

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Husheng Li. His breadth of knowledge and endless curiosity on tackling hard problem guides me to enjoy my PhD study. There is an old saying in China, 'A teach for a day is a father for lifetime'. He not only teaches me how to admire good research, but also teaches me to be strict with ourselves while lenient with others. Thank you for providing an excellent research environment and for your relentless support.

I would like to express my sincere gratitude to all my collaborators Zhiyang Zhang, Jingchao Bao, Chao Tian, Mustafa. S. L. Aljumaily, Matthew Trinkle, Aleksandar D. Dimitrovski, and Ju Bin Songand, with special thanks to Zhenghao Zhang. It was a pleasure to collaborate with and to learn from you all.

I would like to thank the faculty, staff and colleagues in the Department of Electrical Engineering and Computer Scienceat at the University of Tennessee, Knoxville for providing a great work environment. I would like to thank my mentor at Google, Lu Han. I had a great time, thank you all.

I would like to thank my parents whose dedication to my education provided the foundation for my studies. I would like to thank my family and friends for their support. I would like to thank my furry companion for her consistent waiting at the door to welcome me every late night.

Abstract

The intersection of communication and machine learning is attracting increasing interest from both communities. On the one hand, the development of modern communication system brings large amount of data and high performance requirement, which challenges the classic analytical-derivation based study philosophy and encourages the researchers to explore the data driven method, such as machine learning, to solve the problems with high complexity and large scale. On the other hand, the usage of distributed machine learning introduces the communication cost as one of the basic considerations for the design of machine learning algorithm and system.

In this thesis, we first explore the application of machine learning on one of the classic problems in wireless network, resource allocation, for heterogeneous millimeter wave networks when the environment is with high dynamics. We address the practical concerns by providing the efficient online and distributed framework. In the second part, some sampling based communication-efficient distributed learning algorithm is proposed. We utilize the trade-off between the local computation and the total communication cost and propose the algorithm with good theoretical bound. In more detail, this thesis makes the following contributions

- We introduced an reinforcement learning framework to solve the resource allocation problems in heterogeneous millimeter wave network. The large state/action space is decomposed according to the topology of the network and solved by an efficient distribtued message passing algorithm. We further speed up the inference process by an online updating process.

- We proposed the distributed coreset based boosting framework. An efficient coreset construction algorithm is proposed based on the prior knowledge provided by

clustering. Then the coreset is integrated with boosting with improved convergence rate. We extend the proposed boosting framework to the distributed setting, where the communication cost is reduced by the good approximation of coreset.

- We propose an selective sampling framework to construct a subset of sample that could effectively represent the model space. Based on the prior distribution of the model space or the large amount of samples from model space, we derive a computational efficient method to construct such subset by minimizing the error of classifying a classifier.

Table of Contents

1 Introduction **1**

1.1 Motivation . . . 1

1.1.1 Machine Learning for Communication . . . 1

1.1.2 Communication for Machine Learning . . . 3

1.2 Connection to Existing Work . . . 4

1.2.1 Resource Allocation in Wireless Cellar Network . . . 4

1.2.2 Communication Efficient Distributed Learning . . . 13

1.3 Contributions of this Thesis . . . 22

1.4 Previously published works . . . 23

2 Background **24**

2.1 Millimeter Wave Communication . . . 24

2.1.1 Advantages for Millimeter Communication . . . 25

2.1.2 Challenges for Millimeter Communication . . . 26

2.2 Heterogeneous Network . . . 26

2.3 Boosting . . . 28

2.3.1 Relation to Functional Gradient Descent . . . 28

2.3.2 Difficulty in Distributed Boosting . . . 29

2.4 Coreset . . . 30

3 Message Passing Based Distributed Learning for Joint Resource Allocation in Millimeter Wave Heterogeneous Networks **32**

3.1 Introduction . . . 32

3.2 System Model 36
3.2.1 Deployment Model 36
3.2.2 Channel Model 36
3.2.3 Interference Model 37
3.2.4 Q-learning 39
3.3 State/Action Space Decomposition 41
3.3.1 Coordination Graph 43
3.3.2 Agent-Based Decomposition 43
3.3.3 Edge-based Decomposition 48
3.4 Distributed Message Passing on Coordination Graph 49
3.4.1 Max-sum Problem 49
3.4.2 Efficient Belief Propagation for Repeated Inference 52
3.5 Model-based Acceleration 53
3.6 Experiment and Simulation Result 56
3.6.1 Max-Sum Result 56
3.6.2 Simulation Parameters 57
3.6.3 Simulation Results 61
3.7 Conclusion 64

4 Distributed Coreset Boosting **66**
4.1 Introduction 66
4.2 Problem Setting 67
4.3 Generalized Coreset Construction 68
4.4 Coreset Boosting 72
4.5 Distributed Coreset Boosting 77
4.6 Result 81
4.6.1 Approximation Quality 81
4.6.2 Learning Quality 82
4.6.3 Communication Cost 89
4.6.4 Robustness 89

4.7 Proof 90
4.7.1 Proof for Theorem 2 93
4.7.2 Definition of ϵ-Approximation 94
4.7.3 Proof of Corollary 1 95

5 Selective Sampling Based Efficient Classifier Representation in Distributed Learning 97

5.1 Introduction 97
5.2 Related Work 99
5.3 System Model 100
5.3.1 Classification 100
5.3.2 Network 100
5.3.3 Division of Classifier Space 101
5.3.4 Learning Goal: Good Classifiers 102
5.3.5 Learning and Communication Model 103
5.4 Classifier Representation Via Sample Selection 103
5.4.1 Formulation and Simplification 103
5.4.2 Algorithm of Sample Selection 108
5.5 Numerical Results 109
5.5.1 Synthetic Data 109
5.5.2 Real World Data 111
5.5.3 Computational Cost 112
5.5.4 Selection of α 113
5.5.5 Performance of Classification of Classifiers 113
5.6 Conclusion 115

6 Open Problems and Future Work 116

Bibliography 118

Vita 138

List of Tables

3.1 Comparison between Markov random field and coordination graph 50
3.2 Max-Sum result 57
3.3 Simulation Parameters 62

4.1 Classification accuracy on various data sets using subset. 80
4.2 Classification accuracy on various data sets using all the training set. 81

5.1 Learning result for the synthetic data 111

List of Figures

1.1 When only a subset of data is sample, random may fail to find the accurate structure of the data set while reweighting the data set according to the prior may improve the performance. 20

2.1 A 3-tier heterogeneous network, where the BSs are located in the center of the cell with the pico and femto BSs located along the BSs. 27

3.1 Coordination graph for macrocell mmWave network 44
3.2 Agent-based Decomposition 44
3.3 Edge-based Decomposition 49
3.4 Coordination graph and Markov random field 51
3.5 Convergeence for BP 58
3.6 Image of outdoor measurement for LOS 59
3.7 Measured path loss values relative to distance of 20m. 59
3.8 Angle Gain 60
3.9 Transmission rate distributions for different settings 61
3.10 Transmission rate versus transmission power 63
3.11 Switching cost and power cost 64
3.12 Running time for different RL frameworks 65

4.1 Cumulative distribution for ϵ 82
4.2 WebSpam training process 83
4.3 WebSpam training loss 84
4.4 CovType training process 85

4.5 CovType training loss 86
4.6 Yahoo training process 87
4.7 Yahoo training loss 88
4.8 Communication cost 89
4.9 Peformance on adversary distribution 89

5.1 Accuracy comparison between the proposed sampling algorithm and random sampling on synthetic data 110
5.2 Illustration of informative data close to the boundary 111
5.3 Accuracy comparison on real world data 112
5.4 Influence of threshold α on the learning performance 113
5.5 Analysis of $P_{err}(S)$ with respect to different α's. 114
5.6 False alarm and missed detection rates 114

Chapter 1

Introduction

The marriage of communication and machine learning is attracting increasing attention currently. On one side, with its successful application on image recognition, nature language process and game playing, machine learning is considered as the potential solution for many modern communication problems that deal with large data volumes and high complexity. Meanwhile, the machine learning community seeks to borrow tool from communication, such as information theory [81], to tackle the bottleneck of the learning problem, such as the model's interpretability [142] and the communication efficiency in distributed learning [? 170]. In this thesis, we explore the application of advanced learning algorithm on complex communication system and provide communication efficient sampling framework for distributed learning system. [82]

1.1 Motivation

1.1.1 Machine Learning for Communication

Modern machine learning research is focused on complex system with high dimensional data and enjoy great breakthrough on various fields and different size, ranging from identifying planet in atmospheric physics to analyzing genome sequencing data in genomics. Such success encourages the researchers from different backgrounds to rethink the way of study using machine learning technique.

The application of machine learning techniques in communication network has a long and continuous history covering almost all layers. For example, in network layer, machine learning algorithms are used for optimal packet routing [86], traffic classification [5] and network prediction [91] to improve the overall throughput of the network. In application layer, the massive mobile data has been used for health care [90], anomaly detection[141] and some privacy related issues [137]. The recent success of deep learning has further underpinned new and powerful tools to solve those problems.

However, comparing to the data driven philosophy in machine learning, for the physical layer study in communication networks, especially for the wireless communications study, the previous studies are predominantly model based. A significant degree of analytical derivations based on probabilistic models are well characterized. Besides, the transmit signals in communication are designed by human, comparing to computer vision and natural language processing problem, where there is no rigid mathematical models. Such prior knowledge enables the research to design straightforward algorithms based on the probabilistic models. Therefore, there is a high bar of performance for machine learning technique to defeat to provide reasonable new benefits. For example, Polar Code provably achieves the channel capacity for symmetric binary-input, discrete, memoryless channels.

With the emerge of the fifth generation (5G) cellular communications, massive new features are introduced into the design of wireless communication, such as beamforming, multiple-input and multiple-output (MIMO). The increasing features often entail considerable complexity for previous algorithms, which creates a serious gap between theoretical design/analysis and real-time processing. Besides, some assumptions on the previous probabilistic models may not be practical in 5G network. To address the challenge, the machine learning techniques are reconsidered as the promising solutions, since it requires little prior knowledge and assumptions and the model is directly trained from the labeled data. In the first part of the thesis, we will propose a reinforcement learning based framework to efficient solve a joint resource allocation in millimeter wave heterogeneous network.

1.1.2 Communication for Machine Learning

The big trend of modern machine learning research is about the scalability. The typical image classification problem is trained on millions of labeled data, the machine translation service from Google uses tens of millions of bilingual sentence pairs and the learning algorithm for logs searching collects data from billions of users[153]. To improve the performance, the corresponding learning model for those big data problem becomes huge. For example, the DistBelief model has over 10^10 weights [38]. The training of such large models would further require large computation resource [29].

To process such big data, big model and big computation problem, the classic learning framework is no longer practical. The massive data set may not be fitted into a single computer's storage and the computation resource on one computer is far from enough to complete the training in reasonable time scale. It is also subject to the constraints of privacy and data sovereignty laws that moving large amounts to process in a centralized way is not practical. Thus, in recent years, the computational paradigm for large scale machine learning has shifted towards massively large distributed systems, where the computation and data are distributed over individually small and unreliable computational nodes. Then the distributed machine learning algorithms are employed to process such problems.

In machine learning, the performance of the model is evaluated based on its accuracy and computational efficiency. The former measures how accurate the prediction is made on new instances and the latter focused on how much computation is needed to achieve the corresponding accuracy. The common philosophy of designing a good machine learning algorithm will require the consideration on both dimensions. In distributed machine learning, as studied by a variety of distributed computing platforms, the communication cost becomes the third dimension. In distributed learning, the massive message, including data and the model parameters, would be transferred throughout the computer network. Although the speed of Ethernet could achieve 10 megabits per second, it is still slow comparing to the CPU's operation time. In fact, the overhead for a single message exchange can be long enough for thousands or more floating point operations. Besides, synchronization becomes

a big issue with the increase number of computing nodes. Recent studies confirmed that the communication could be the bottleneck for the distributed learning system.

To design a good distributed learning algorithm, the trade off between accuracy, computation and communication efficiency must be considered jointly. For the trade off between communication and computation, higher level parallelization for the computation would improve the computation efficiency as more distributed nodes are involved in the training. The running time could be impressively reduced comparing to the serialized pattern. In the meantime, the coordination and synchronization between the distributed nodes would require more extra communication overhead with the increase of the distributed nodes. The trade off between the communication and accuracy could be considered in the view of information. Learning the optimal model distributedly is similar to collecting information from separated nodes. The best solution is to aggregate all the local information together for a centralized learning process using massive communication. In contrast, if no communication is allowed to exchanged information among the distributed nodes, then the learning can only be performed on single machine and the model could not guarantee a global optimal solution.

Given the situation, in the second part of the thesis, we investigate some sampling based approximate algorithms, which allow us to sample the inexact but good enough approximation (subset) of the data for distributed learning.

1.2 Connection to Existing Work

In this section, we list some of the existing research that related to the topic in this thesis.

1.2.1 Resource Allocation in Wireless Cellar Network

In heterogeneous cellar network, multiple base stations (BS) are deployed within the cell to serve the User Equipment (UE) simultaneously and increase the capacity. Since those BSs share the same frequency, the signal from different BSs become interference to each other. It is critical to manage the user association policy for each UE to maximize the network throughput. On the BS side, the increasing number of UEs requires larger transmission power

according to the quality of service (QoS) requirement for each UE. The energy efficiency becomes another key dimension such that each BS need to properly control its transmission power to satisfy the QoS requirement while avoiding making large interference and costing unnecessary transmission power. Such joint consideration on the user association and power control within the cellar network is considered as the resource allocation problem.

In the existing Long Term Evolution (LTE) systems, the most prevalent solution for the resource allocation problem is based on the received power [42], where the choice of connected BS for each UE is determined when the signal power from the corresponding BS is the largest. Although this policy works well in existing LTE system, previous study pointed out that this could cause serious load balancing problem and the system's total throughput is far from optimal[161].

There are numerous of previous works focusing on the resource allocation problem in heterogeneous networks. [8] attributed the heterogeneous networks' success to seven key factors, which are performance metric, topology, cell association, downlink vs uplink, mobility, backhaul and interference management. The resource allocation problem is directly related to two of them. [59] introduced new theoretical models for understanding the heterogeneous cellular networks, identifying the practical constraints and challenges to tackle. The authors pointed out that in heterogeneous networks most UEs would connect to the BS with strongest transmission power while the picocells with smaller transmission power have less connected UE. This is highly suboptimal from a network-wide point as moving a UE from a heavily-loaded macro-base station to a nearby lightly loaded picocell would benefit both that UE as well as the macrocell users by achieving better load balancing. The detailed overview about the load balancing in heterogeneous networks could be found in [9] and some potential strategy, such as biasing, blanking, small cell planning are discussed. The energy efficiency, one of the key dimension in resource allocation, is discussed in [114] for heterogeneous networks. The paper systematically reviewed and evaluated the various studies performed in the area of energy efficient resource management in cellular networks. The solution considering the long-term time scale and short-term time scale are introduced while the former could be modeled as the user association problem and the latter could be considered as the BS operation subproblem. They provided the analysis to jointly solve

the combined long and short term time scale problem by formulating weighted optimization problem. [146] studied the mathematical modeling for network selection in heterogeneous networks, where UE could switch between different radio access technologies rather than connect to different base stations.

While all above comprehensive surveys provide thorough insight into the resource allocation problem for classic heterogeneous networks, new challenges arise in 5G era. Millimeter wave is introduced in 5G era to provide enormous spectrum. However, millimeter wave communication suffers from high space path loss and is easy to be blocked. This unique radio propagation characteristics introduces high dynamics and new feature to the channel model. The communication may switch between the Line-Of-Sight, None-Line-Of-Sight and even blockage state, where the channel model for each state is totally different, and the state switch may happen in the order of hundreds of milliseconds or even less[113], much faster than LTE and other previous technologies. Besides, in millimeter wave communication, beamforming is used to provide higher transmission gain using narrow beam and antenna arrays. Such directional transmission pattern could help suppress the interference arriving from neighboring cells and the network could be considered as noise-limited, comparing to the previous interference-limited networks. The above new features make it necessary to rethink the resource allocation problem in 5G era.

The resource allocation problem in wireless networks could be generally considered as a utility maximization problem subject to a resource or/and power constraint. In the 5G case, the utility includes spectrum efficiency, energy efficiency, QoS with linear [157], logarithmic [128], exponential [163], and sigmoidal [33] forms depending on the problem setting. Relaxed optimization, game theory, stochastic geometry and Markov decision processes are the popular tools to solve the utility maximization problem.

Relaxed Optimization

The typical user association for resource allocation could be modeled as constraint binary association problem, where each UE is assigned for one BS. It is NP hard and not computable even for modest-sized wireless networks. In 5G era, as the BS density and the number of UE within the cellar is large and the high dynamics of the channel makes the problem even more

complex as the handover cost or the re-association has to be considered, it is not practical to solve the utility problem directly. One way to make the problem convex is to relax the binary constraint. Instead of allowing the UE to connect to only one BS, the multi-connectities is allowed for each UE, transforming the binary constraint to a real number between 0 and 1. Although the multi-connectities assumption is not practical as it requires large amount of overhead for the control, it could provides some insight into the system's performance. This is because the relaxed problem could upper bound the performance for the binary constraint problem. Then the relaxed utility maximization could be solved by some standard optimization tools, such as dual decomposition in a distributed manner, which could efficiently converge to the near-optimal solution.

There are extensive of previous researches working on resource allocation problem for 5G network using relaxed optimization. [36] developed a new theoretical framework to study cell association for the downlink of multi-cell networks and derive an upper bound on the achievable sum rate. The heuristic based solution is proposed to achieve the near-optimal solution. [53] leveraged the benefits of small cell network and proposed a cooperative small cell network architecture that jointly considering the user handover, channel borrowing sensing and base station coordination. [100] described new paradigms for design and operation of heterogeneous cellular networks focusing on cell splitting , semi-static resource negotiation, range expansion and fast interference management. They proposed a simple and efficient solution to solve the problem. [156] modeled the resource allocation in millimeter wave network as a novel multi-dimensional assignment problem, for which an original solution method is established by a series of transformations that lead to a tractable minimum cost flow problem. [155] considered a hybrid heterogeneous network, where macro cells adopt massive MIMO, and small cells adopt millimeter wave transmissions. There simulation proved that, compared with massive MIMO macro cells, millimeter wave small cells play a dominant role in enhancing the throughput of the networks due to the larger bandwidths. [14] converts the resource allocation problem to a minimum cost flow problem and allows to design an efficient algorithm by a combination of auction algorithms. The solution algorithm exploits the network optimization structure of the problem with much more powerful than

computationally intensive general-purpose solvers. In [136], joint optimization of the long-term base station sleep-mode activation, user association, and sub-carrier allocation was considered for maximizing the energy efficiency or minimizing the total power consumption under the constraints of maintaining the fairness for each UE within the network. The utility function has been relaex to the convex function and the corresponding near-optimal solution is obtained with significant gains in saving power and increasing energy efficiency. [121] analyzes the impact of user mobility in multi-tier heterogeneous networks for 5G communication.The optimal bias factors for user association is obtained to maximize the coverage. From their simulation, when the user is mobile, and the network is sensitive to handoffs, both the optimum tier association and the probability of coverage depend on the users speed; a speed-dependent bias factor can then adjust the tier association to effectively improve the coverage, and hence system performance. In [37], the authors consider a dynamic control problem for mobile association and solved the problem using a Semi Markov Decision Process framework. Numerical results showed that mobility can even be beneficial to the system performance.

Game Theory

Game theory is the study of mathematical models, which focuses on the conflict and cooperation between intelligent rational decision-makers. It has widely application ranging from economics, political science to psychology and computer science. The optimal strategy for all the players to achieve the maximum utility is known as equilibrium. The resource allocation problem could be considered as the game, where all the UEs and/or BSs are considered as the players. Game theory is a powerful tool since it provides tractable methods for the investigation of very large decentralized optimization problems. While the Relaxed Optimization model is used to maximize the overall utility for the whole system, game theory provides more flexible. The problem could be considered as the competing game if all the players seek to maximize their own utility and compete against each other with different strategies. By contrast, when the target is to maximize the overall system's utility, the problem could be regarded as a cooperative game where players bargain with each other for the sake of attaining mutual advantages [89]. However, it is important to note that game

theory operates under the assumption that all the players are rational, which might not be the case for wireless networks[80]. Another drawback for game theory approach is that the convergence of the resulting algorithms is, in general, not guaranteed [66]. Besides, since game theory only provide the decision making strategy, there is no closed form between utility metric and the network's parameters, it could provide less insight into the design of the system, comparing the following stochastic geometry approach.

In [24], the author considered the optimal user-cell association problem for massive MIMO heterogeneous networks and simple decentralized user-centric association schemes, where each user individually and selfishly connects to the base station with the highest promised throughput. The users make local association decisions in a probabilistic manner can be viewed as games and are known to converge to Nash equilibria. [12] proposed two general classes of throughput models that capture the basic properties of random access and and scheduled access. Based on the proposed models, a non-cooperative game is formulated and an efficient solution with good convergence is provided. [108] formulated the dynamics of network selection problem in a heterogeneous wireless network using the theory of evolutionary games. With the help of reinforcement learning, a user can gradually learn and adapt the decision on network selection to reach evolutionary equilibrium without any interaction with other users, such that the computation is totally distributed. [158] developed a repeated game model, which leads to distributed user association algorithms with proven convergence to the Nash equilibrium. [63] proposed a universal joint BS association and power control algorithm for heterogeneous cellular networks where the algorithm iteratively update the user association policy and transmission power based on previous iteration. They proved that the proposed algorithm is the solution to a non-cooperative game. In [124], The resource allocation problem is formulated as a many-to-one matching game in which the UE and BS rank one another based on utility functions that account for both the achievable performance, in terms of rate and fairness to cell edge users, as captured by newly proposed priorities.

Stochastic Geometry

For more than three decades, stochastic geometry has been used to model large-scale wireless networks, and it has succeeded to provide tractable models to capture and better understand the performance of the networks[49]. The BS and UE with the heterogeneous networks are modeled by a point process, including Poisson point process (PPP), the Binomial point process (BPP), the Hard core point process (HCPP), and the Poisson cluster process (PCP). By using the Poisson point process and Rayleigh Fading assumption, we could obtain the tractable expression for the key metric, such as the coverage probability, energy efficiency. The Stochastic Geometry could provide the insight of some system parameters' impact on the system performance based on the derived analytical expression. Another advantage for Stochastic Geometry is its computation efficiency as calculating the closed form expression require much less computation comparing to solving large scale relaxed convex optimization. However, the performance of Stochastic Geometry is highly dependent on the model assumption. When Poisson point process and Rayleigh Fading is not a good approximation for the true system, the result from Stochastic Geometry may suffer from large performance gap.

Modeling and analysis of heterogeneous cellular wireless networks is increasingly attracting the attention of the research community. [39] develop a tractable, flexible, and accurate model for a downlink heterogeneous cellular network consisting of K tiers of randomly located BSs, where each tier may differ in terms of average transmit power, supported data rate and BS density. An expression for the probability of coverage is derived over the entire network under both open and closed access. In [132], the author considered a general and tractable model that consists of multiple different radio access technology, each with different tiers. The the distribution of rate over the entire network is then derived for a weighted association strategy. [133] proposed a general and tractable millimeter wave cellular model capturing high near-field path loss and poor diffraction for millimeter transmission. The BSs backhauling in a mesh architecture is proposed. The analysis of the proposed framework showed that increasing the system bandwidth does not significantly influence the cell edge rate, although it boosts the median and peak rates. [16] proposes a general framework

to evaluate the coverage and rate performance in millimeter wave cellular networks. The locations of the LOS and non-LOS base stations are modeled as two independent non-homogeneous Poisson point processes, to which different path loss laws are applied. The results show that dense mmWave networks can achieve comparable coverage and much higher data rates than conventional wireless networks. [73] derives the outage probability of a typical user in the whole network or a certain tier, which is equivalently the downlink SINR cumulative distribution function for heterogeneous networks under the implicitly assumption all base stations have full queues. The result indicates that the biasing factor for user association has large impact on various metrics of the system. A joint resource partitioning and offloading in a two-tier cellular network is considered in [131]. It is shown that show that load balancing, by itself, is insufficient, and resource partitioning is required in conjunction with offloading to improve the rate of cell edge users in co-channel heterogeneous networks. In [19], the average downlink user data rate is derived for the joint spectrum allocation and user association in heterogeneous cellular networks. Then the data rate is employed as the objective function in jointly optimizing spectrum allocation and user association and a computationally efficient solution is proposed. A Surcharge Pricing Scheme is also presented, such that the designed association bias values can be achieved in Nash equilibrium. In [120], the author explored the optimality of the intuitive solution that the fraction of spectrum allocated to each tier should be equal to the tier association probability in heterogeneous networks.

Markov Decision Processes

The user association and power control in wireless networks could be considered as sequential decision making problem of discrete time stochastic systems in the presence of uncertainty. For the millimeter communication, the uncertainty has larger impact on the systems' performance as

- Due to the high path loss, millimeter introduces multiple channel states, namely the Line-Of-Sight (LOS), None-Line-Of-Sight (NLOS) and outage state, to characterize the different transmission properties of the channel.

- By deploying beamforming, the directional transmission technology, the alignment between the transmitter and receiver would have an impact on the channel gain and thus introduce extra randomness to the system model.

Markov decision processes is a powerful tool to model the sequential decision making problem. The objective is to perform actions in the current state to maximize the future expected reward. Under the assumption that the future state s^{t+1} is only dependent on the current state s^t and action s^t, the optimization problem defined by the sequential decision making could be solved via dynamic programming and reinforcement learning. In millimeter wave communication, the channel may switch between LOS and NLOS state on the order of hundreds of milliseconds. This requires an efficient decision making framework. Reinforcement learning is considered as the potential solution. For example, in Q-learning, the optimal policy is learned via value iteration and the result is described by a Q-function. The current state s^t and available action a^t are the parameters of Q-function and the optimal decision in current state s^t is obtained by solving the

$$\arg\max_{a \in A} Q(s^t, a) \tag{1.1}$$

In most cases, the Q-function is a discrete table characterized by the state space s and action space t and thus solving (1.1) is efficient.

However, as the size of the network increases, the state space and the action space increases exponentially, which makes it hard to solve exactly. Besides, since Q-function is modeled as the discrete table, it has limitations when dealing with continuous state spaces. Besides, the performance of MDP based approach is highly dependent on the choice of states and a reasonable state transition mode. As the resource allocation problem in heterogeneous networks is always complex and unstructured, it is still an open question to find the good state space.

Recent breakthrough in machine learning community helped to overcome the limitation of MDP based approach. The deep reinforcement learning framework approximates the Q-function with the multi-layers neural network. Using stochastic gradient descent, the network could be trained to have good representation for the Q-function. The striking

success of Alpha Go [129] proves that such framework could handle the state/action space with size of more than one billion. Since neural network could use continuous input, it could be extended to the control task with continous state space.

[159] gave an extensive overview on the application of reinforcement learning on wireless network. [135] studied the choice of wireless network with Markov decision process. The objective is to maximize the total expected reward per connection. The problem is solved via decomposing the complicated MDP problem into a hierarchy of simpler and more manageable subproblems. [32] determined the conditions under which a mobile terminal switches from one network to another should be performed. The value iteration algorithm is used to compute a stationary deterministic policy. [48] propose hybrid schemes where the wireless users are assisted in their decisions by the network that broadcasts aggregated load information. The equilibria is obtained using a Bayesian framework. In [47] an alternative channel allocation scheme is proposed in mobile cellular networks that supports multiple heterogeneous traffic classes. It is asymptotically optimal, computationally inexpensive, model-free, and can adapt to changing traffic conditions. The goal of [82] is to maximize the secondary users' performance while bounding the performance degradation of the primary users. The secondary user is modeled as the agent and its instantaneous is based on long-term impact on the temporal evolution of the network. An iterative method is proposed to calculate the optimal strategy with no a prior knowledge of the statistics of the Markov process.

1.2.2 Communication Efficient Distributed Learning

Most machine learning problems could be formulated as the statistical optimization problem. The generalized target function could be defined as

$$f(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(\theta, z_i) \tag{1.2}$$

where $l(\theta, z_i)$ is the loss function for data z_i and n is the size of training set. The goal is to find the minimizer of $E[f(\theta)]$. Here the expectation is over the distribution of z_i. Since the distribution of the data is unknown, instead of minimizing the generalization risk, the

empirical risk minimization of (1.2) is computed. The performance of the empirical minimizer could be further bounded by VC theory [143] or Rademacher complexity [20].

The optimization in (1.2) could be convex or even none convex. For the convex problem, the gradient descent method[106, 107] could be applied to calculate the global empirical risk minimizer. Further advanced method based on gradient descent have been proposed to speed up the convergence [123]. Although it is easy to parallelize the gradient descent method in distributed setting [21], it requires to go through the entire data set for each iteration, which makes it less efficient to complete one iteration on the massive data set.

The stochastic gradient method and its variants are considered as the solutions for large scale optimization in machine learning, as for each iteration, only a subset of the data is required to calculated the stochastic gradient while it could still achieve the same accuracy[167, 154, 26, 65], comparing to the full gradient version. The power of stochastic gradient has been extensively studied. Some recent researches reveal that empirically, by adding noise to the stochastic gradient, the performance of stochastic gradient could be improved such that we could escape the poor local minima[105] even for none-convex target function. [58] proved that by adding isotropic noise n and choosing the sufficiently small step size η, the iterative update

$$\theta = \theta - \eta(\bigtriangledown f(\theta, z) + n) \tag{1.3}$$

could guarantee to escape strict saddle points. [72] shown that a perturbed form of gradient descent can converge to a second-order-stationary point at almost the same rate as standard gradient descent converges to a first-order-stationary point. [170] analyzes the hitting time of such noise gradient based algorithm could finds an approximate local minimum of the population risk in polynomial time, escaping suboptimal local minima that only exist in the empirical risk.

We denote the communication complexity for distributed learning as

$$B = N * M * T \tag{1.4}$$

where B is total communication complexity, N is the number of message for each iteration, M is the size for each message and T is the number of iteration required for the convergence. We could design the communication efficient distributed learning algorithm by

- *Modify the Communication Pattern*: As synchronization is required in the distributed learning framework, the simple solution is in each iteration, every node broadcast its local information to all the other nodes. Then the total communication is in the order of $O(n^2)$, where n is the number of total nodes. By modifying the pattern for exchanging message among the distributed nodes, we could improve the communication efficiency while maintaining the good statistical performance.

- *Decrease the Message Size*: In stochastic gradient framework, the message is the gradient itself and thus the message size is proportional to the model complexity. When the model is large, it is necessary to utilize the structure of the gradient, for example, the sparsity, to reduce the message size.

- *Speed up the convergence*: Comparing to the communication, the computation is cheap in the distributed setting. By increasing the computation cost on the distributed node, for example, using advanced sampling instead of random sampling, we could speed up the convergence of the distributed learning algorithm.

In this subsection, we survey some of the existing lines of research that explore themes related to communication efficient distributed learning.

Modify the Communication Pattern

Because of the incremental nature of the stochastic gradient, where individual update relies on the outcome of all previous updates, it is none-trivial to extend the stochastic gradient to the distributed learning setting. For example, the momentum based stochastic gradient descent need to remember the momentum in each iteration as follows

$$\theta^{t+1} = \theta^t + \Delta\theta^t \tag{1.5}$$

where

$$\Delta\theta^t = \alpha\Delta\theta^{t-1} - \eta \bigtriangledown f_i(\theta, z) \tag{1.6}$$

Although each distributed node could calculated $\bigtriangledown f_i(\theta, z)$ based on its local data set, the update include the previous direction in the parameter space θ^{t-1}. This requires all the distributed nodes share the parameters for θ^{t-1} and they have frequently access to the shared parameters while when they perform the computation to refine it [83]. When the size for the model θ is large, for example, the complex model for modern deep learning application may have 10^9 to 10^{12} parameters, the previous assumption is not practical. Accessing the parameters requires an enormous amount of network bandwidth and the cost of synchronization is high.

Some recent theoretical result proved that the distributed stochastic gradient with asynchronous update to the parameters could asymptotically achieve comparable performance as synchronized version, which reduced the total communication cost by allowing less access to the global parameters for each distributed nodes. [1] showed that for smooth stochastic problems, the delays are asymptotically negligible, where a master node performs parameter updates while worker nodes compute stochastic gradients based on local information in parallel. [117] proposed an update scheme which allows processors access to shared memory with the possibility of overwriting each other. When the associated optimization problem is sparse, meaning most gradient updates only modify small parts of the decision variable, the proposed method achieves a nearly optimal rate of convergence. [45, 168] establish lower bounds on minimax risks for distributed statistical estimation under a communication budget. The lower bounds reveal the minimum amount of communication required by any procedure to achieve the centralized minimax-optimal rates for statistical estimation.

Another line of study focus on the incremental sub-gradient methods, which involve every machine minimizing its own objective function instead of calculating the global minimizer. Then the information is exchanged locally with other machines in the network over a time-varying topology, very similar to the message passing model in belief propagation [162]. [112] consider a distributed multi-agent network system where the goal is to minimize a

sum of convex objective functions of the agents subject to a common convex constraint set. Each agent combines weighted averages of the received message from the neighboring agent with its local update, and adjusts the update by using subgradient information (known with stochastic errors) of its own function and by projecting onto the constraint set. [74] presents an algorithm that generalizes the randomized incremental subgradient method with fixed step size. The stochastic component in the algorithm is described by a Markov chain, which can be constructed in a distributed fashion using only local information. [104] study the case where each agent has a locally known, different, convex and potentially non-smooth cost function. The global objective of the agent is to cooperatively minimize the cost function via exchanging the local information with the neighbors.

Decrease the Message Size

Another dimension for the design of communication-efficient distributed learning framework is to reduce the amount of information required to transmit in each iteration while maintaining the accurate result. The idea for this area of research is similar to the data compression in information theory, where the structure of the information is utilized to design the communication protocol. More specifically, it is observed from the empirical experiment that the messages transmit in each iteration are always sparse. For example, when training the Deep Neural Network, 99.9% of the gradient exchange in distributed stochastic gradient descent is redundant. Such sparsity makes it possible to reduce the communication cost for training via gradient sparsification and gradient quantization. [3] map the 99% of smallest updates to zero and update the parameters. [57] adopt the similar idea to sparsify the gradient and prove the convergence to the correct solution in constant number of iterations. [151] aggressively reduce the communication cost by setting the value of the gradient to $\{-1, 0, 1\}$ only. The layer-wise ternarizing and gradient clipping is proposed to improve its convergence. [31] proposes a gradient compression based on localized selection of gradient residues and automatically tunes the compression rate depending on local activity. [148] propose a convex optimization formulation to minimize the coding length of stochastic gradients. Several simple and fast algorithms are proposed for approximate solution, with theoretical guaranteed for sparseness. In [160], the advanced coding method is introduced

to mitigate the effect of stragglers in gradient computation to the gradient. the computation load, straggler tolerance and communication cost are considered and an explicit coding scheme that achieves the optimal trade-off based on recursive polynomial constructions is proposed. As there are lots of overlap of communication and computation during the training and inference of deep neural network, [67] develop a system for communication scheduling which realizes near-optimal overlap of communication and computation in graph-based models. Thorough review for this line of study could be found in [23].

Speed up the convergence

Since the total communication is related to the number of iteration needed for convergence and the information required for each iteration. Some previous utilized the trade-off between communication and computation. By allowing more computation cost for each iteration, the convergence could be accelerated. Communication is critical to the system's performance while the computation is reletively cheap comparing to the communication. [127] presents novel Newton-type method for distributed optimization, which is particularly well suited for stochastic optimization. The method enjoys a linear rate of convergence which provably improves with the data size, requiring an essentially constant number of iterations under reasonable assumptions. [22] propose a distributed Frank-Wolfe algorithm. In each iteration, each node finds largest entry of the local gradient in absolute value. Then the system compute index of node with largest overall gradient. The update for each iteration depends on largest overall gradient instead of the average of all the distributed gradient. [126] introduce an accelerated mini-batch version of stochastic dual coordinate ascent and prove a fast convergence rate for this method. The results indicates its outperformance over vanilla stochastic dual coordinate ascent and to the accelerated deterministic gradient descent method. The algorithm in [171] is based on an inexact damped Newton method, where the inexact Newton steps are computed by a distributed preconditioned conjugate gradient method.

In statistical optimization setting, we assume that the data z_i among all the distributed nodes are i.i.d. This assumption may not be accurate for practical problem. Consider the case where the user data from China and England may stored distributedly and we want

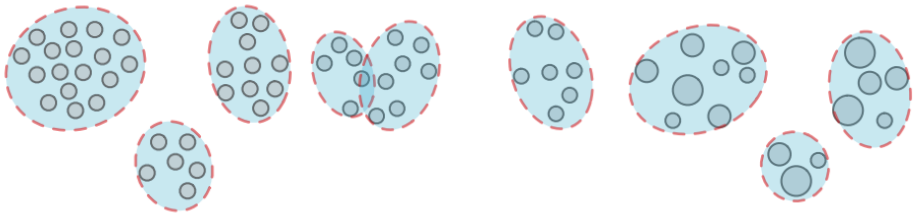
to calculate the global optimizer based on these data. It is easy to verify that the data distributions from China and England may have large difference and the i.i.d assumption no longer hold. To speed up the convergence, the discussion in last paragraph may not be valid when the distribution on each distributed node are not i.i.d.

For such more general case, [152] considers a number of fundamental statistical and graph problems in the message-passing model. They show shows that exact computation of many statistical and graph problems in this distributed setting requires a prohibitively large amount of communication, and often one cannot improve upon the communication of the simple protocol in which all machines send their data to a centralized server. Considering the speed of communication through the cable could not achieve the comparable performance to the CPU's computation time, to design the communication-efficient distributed learning algorithm, we need to allow approximation of the data sets.

Random sampling is a good baseline approach for data set approximation. In [17], instead of being in the statistical estimation framework, the authors consider the problem of PAC-learning from distributed data and analyze fundamental communication complexity questions involved. The general upper and lower bounds on the amount of communication needed to learn well with random sampling is provided. [164] consider the case when the distributions for the distributed nodes are arbitrary and potentially adversarial. They develop distributed learning algorithms that are provably robust against such adversarial distributions with a focus on achieving optimal statistical performance. The proposed algorithm is median-based with random sampling such that it is robust to a small fraction of adversarial distributions.

In the stochastic framework for distributed setting, a subset of data (batch) via random sampling is handled instead of the whole data set. However, random sampling is not a good approximation when the size of the subset is small. Fig 1.1 demonstrates the case when we run clustering based on a subset of data and fail to find the correct clusters for the whole data set.

When random sampling could not provide good approximation, extra iterations are necessary to achieve the good performance, which decreases the convergence rate.



(a) Original Dataset **(b)** Random Sampling **(c)** Coreset Sampling

Figure 1.1: When only a subset of data is sample, random may fail to find the accurate structure of the data set while reweighting the data set according to the prior may improve the performance.

Coreset, the small and weighted summary of large data set, is proposed as the solution for efficient approximation. In many applications, the size of the coreset is independent to the size of the whole data set. For large scale problem in distributed setting, by allowing more computation on constructing the coreset, we could speed up the convergence by utilizing the advantage of coreset over random sampling and consequently reducing the communication cost.

The applications of coreset on general machine learning problems have been widely studied. [27] give deterministic, low-order polynomial-time algorithms to construct the coreset with approximation guarantees, together with lower bounds. [97] propose a single, practical algorithm to construct strong coresets for a large class of hard and soft clustering problems based on Bregman divergences. Their theoretical results further imply a randomized polynomial-time approximation scheme for hard clustering. [98] shows that Gaussian mixtures admit coresets of size polynomial in dimension and the number of mixture components, while being independent of the data set size and one can harness computationally intensive algorithms to compute a good approximation on a significantly smaller data set. [118] studied the coreset construction in classification. They present a general framework for analyzing coreset-based optimization and provide interesting insights into existing algorithms from this perspective. A new coreset construction is proposed and a wide class of problems that include logistic regression and support vector machines is discussed to integrate with the proposed coreset construction.

Distributed Learning System

The progress of theoretical breakthrough encourage the develop of distributed learning system for large scale machine learning problem. The engineering challenges for the design of distributed learning system includes the following key features [83].

- **Communication efficiency**: As synchronizing costs massive amount of communication, for the real distributed system, the asynchronous communication model is required. The message for each iteration should be carefully designed to further reduce the communication overhead.
- **Flexible consistency models**: Maintaining the consistency of the model among all the distributed nodes will block parallel computation and therefore cause large latency.
- **Elstic scalability**: New nodes can be added without restarting the running framework.
- **Fault tolerance and durability**: Recovery from the failed running machine and interrupting computation is required.
- **Ease of Use**: The system should support multiple kinds of machine tasks.

Recent years witnessed a flurry of research on the design of distributed machine learning system. [84] offers two relaxations to balance system performance and algorithm efficiency for the proposed distributed learning system, asynchronous task dependency and flexible consistency model. The workload is decomposed to into multiple tasks and the tasks are executed asynchronously. Each distributed node would first pull the parameters from the server and then complete its local computation in parallel. The results are pushed back to the server. The worker nodes do not stop pulling the new parameters from the server unless the parameters on the server have not been update since τ seconds ago. This delay bounded pattern could make sure all the distributed nodes could keep running for most of time. The convergence analysis is provided. [69] introduce a new, general geo-distributed ML system that decouples the communication within a data center from the communication between data centers, enabling different communication and consistency

models for each. The key idea is to dynamically eliminate insignificant communication between data centers while still guaranteeing the correctness of ML algorithms. [169] propose a general framework for parallelizing stochastic algorithms on multi-node distributed systems. Using the programming interface, the user develops sequential stochastic algorithms without concerning any detail about distributed computing

1.3 Contributions of this Thesis

This section highlight the our contributions to address the problems in Section 1.2

- In Chapter 3, we propose an efficient distributed message passing algorithm to solve the resource allocation problem in heterogeneous mmWave networks. To utilize the dynamics of the mmWave networks, the Q-learning approach is considered to find the optimal policy to maximize the overall throughput of the system while reducing the energy cost. The large state/action space in Q-learning is decomposed according to the coordination graph defined by the network topology. Then the max-sum problem in the decomposed Q-learning problem is solved by distributed message passing algorithm. We further speed up the learning process by introducing the prior of the channel dynamics and inference process by modifying the message passing algorithm such that it could be updated in an online manner.

- In chapter 4, we propose an communication efficient coreset construction algorithm for distributed boosting framework. By utilizing the prior structure of the data set by clustering in the preprocessing stage, we could efficiently construct the coreset with the size independent of the total data set. Then the proposed coreset construction algorithm is integrated with boosting framework, which is robust to the outliers and enjoys good convergence property. Then we extend the proposed coreset boosting to the distributed setting, where we prove it is robust to the adversary distribution.

- In chapter 5, we propose an selective sampling framework to construct a subset of sample that could effectively represent the model space. The sample space and the model space are considered as two mutually dual spaces. Based on the prior distribution

of the model space or the large amount of samples from model space, we derive a computational efficient method to construct such subset by minimizing the error of classifying a classifier.

1.4 Previously published works

This dissertation has greatly benefited from collaboration with several colleagues. In particular, my adviser, prof. Husheng Li was actively involved in all the work presented in this dissertation. Chapter 3 was done in close collaboration with Zhiyang Zhang. Chao Tian provided insight discussion for the work in Chapter 5.

Chapter 2

Background

In chapter, we introduce the basic concepts for the problems discussed in this thesis. The readers are encouraged to read this chapter before going to the details in the following chapters.

2.1 Millimeter Wave Communication

Millimeter wave is the band of radio frequencies from 30 to 300 gigahertz(GHz). With the overwhelming capacity demands for current wireless deployed wireless technologies, the millimeter wave communication is considered as the new solution for its orders of magnitude greater bandwidths, further gains via beamforming and spatial multiplexing from multi-element antenna arrays.

The millimeter communication is not a new concept as it is first investigated in the 1890s by Bengali-Indian scientist Jagadish Chandra Bose [25, 113]. The near 60 GHz millimeter wave is used for satellite-based remote sensing to determine temperature in the upper atmosphere [119]. In Europe, millimeter wave was considered for the backhaul communication [6, 64].

Previously, the wireless engineering community considered the millimeter wave to be useless for mobile communication for its absorption by atmospheric gases and additional attenuation by raindrops as their wavelengths are the same order of size. However, recent studies with extensive field measurements [116] revealed that the rain attenuation and

atmospheric absorption characteristics does not create significant path loss for millimeter wave when the transmission distance is in the order of 200m. Since today's cell sizes in urban area are with the similar size, there are tentative plans to use millimeter waves in future 5G mobile communication.

2.1.1 Advantages for Millimeter Communication

To understand the advantage of millimeter communication, we could first look at the channel capacity. With the development of the modern coding theory, the engineers are able to design the communication system that approach this channel capacity and thus it is a good indicator for the performance of communication system. ShannonHartley theorem provides the channel capacity for an additive white Gaussian noise (AWGN) channel

$$C = B \log_2(1 + SINR) \tag{2.1}$$

where B is the total bandwidth and $SINR$ is the signal-to-noise-plus-interference ratio. It is straightforward to conclude that increasing the bandwidth B and improving the $SINR$ could improve the system's performance and millimeter wave could provide enormous improvement on these two features.

- Millimeter wave could provide 200 times more specturm than the current technologies [110]. This is because over 90% of the allocated radio spectrum falls in the millimeter wave band. According to equation (2.1), increasing the bandwidth B could linearly improve the system's throughput.

- Since its wavelength is smaller, it is possible to deploy large numbers of antennas on the devices and base stations for directional communication. On the one hand, the increased number of antennas could used to form very high gain arrays and increase the received signal quality. On the other hand, the antenna arrays enables the directional transmission, which effectively reduces the interference from neighboring transmitters. As a result, the $SINR$ for the communication links would be improved.

2.1.2 Challenges for Millimeter Communication

Despite the potential of millimeter communication, there are a few key challenges for the implementation of millimeter communication system.

- High path loss. According to the Friis transmission law [115], the the free space omnidirectional path loss grows with the square of the frequency. Thus the increased frequency requires larger transmission power or gain to compensate.

- Blockage. Millimeter wave is easily to be blocked. For example, the attenuation caused by brick could be as high as 40-80 dB [110] and even for human body the blockage could cause 20-35 dB [94]. Besides, the reflection from human body and other outdoor material should be considered when designing the millimeter communication system.

- High dynamics. For a given mobile velocity, channel coherence time is linear in the carrier frequency [115]. One mobile device with speed of 60 km/h, its channel may suffer the change in the order of hundreds of microseconds. Besides, the blockage would further increase the dynamics of the millimeter wave's channel in the urban area, where the density of building is high and the moving object would frequently block the channel.

- Power consumption. Although the antenna array could help provide larger transmission gain, maintaining the operation of such antenna array would require more power consumption. Due to the high dynamics of the channel, the devices may switch between different directions to maintain good channel quality, which would further increase the power consumption.

2.2 Heterogeneous Network

In heterogeneous network, shown in Fig 2.1, there is a few base stations located at the central area of the cell with strong transmission power. Small cells, such as picocells, femtocells and relays, are located at the edge or crowded area and transmit at a low power for the traffic offloading.

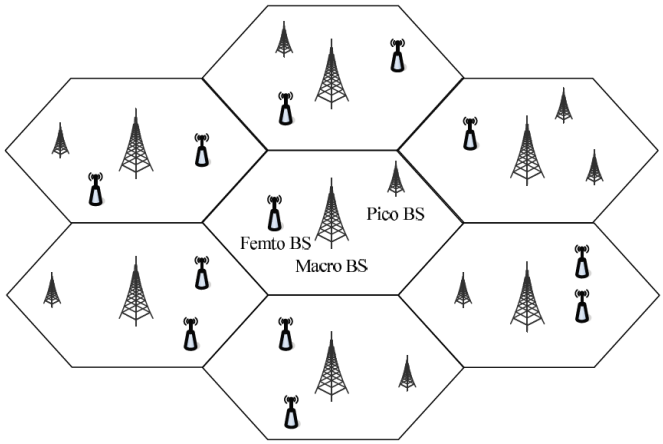


Figure 2.1: A 3-tier heterogeneous network, where the BSs are located in the center of the cell with the pico and femto BSs located along the BSs.

Increased density of BS could improve the coverage quality and enhance the edge users' performance. Besides, many previous studies confirmed that the heterogeneous could improve the spectral efficiency and the energy efficiency [93].

Although heterogeneous introduces many benefits, the increased complexity would challenge the design and control of the network. The resource allocation problem is addressed in many previous studies. Since it requires large overhead for each UE to main multi-links to more than one BSs, in practice, each UE is assigned with one BS. It is important to design the good user association policy. For example, if all the UEs choose to connect to the BS that provides the largest transmission power, most of the UEs would connect to the macro BS, which would potentially resulting in inefficient small cell operation [42].

New challenges are introduced for the resource allocation problem in millimeter wave network. Since the channel suffers high dynamics, the link between the UE and BS because unstable. Tho user association policy should not only consider the long term expected channel quality for each link, but also consider the power consumption as frequent switch between different BS may lead to a better channel quality and more power cost. In mobile communication, such trade-off should be carefully handled. Due to the large path loss, in millimeter wave heterogeneous network, large densities of BSs are required to deploy.

2.3 Boosting

Boosting is a machine learning ensemble meta-algorithm that combines a set of weak learners into a strong learner by assigning weight to the result of each learners and outputting their majority votes. The first successful boosting algorithm, AdaBoost, was proposed by [56] and the authors were awarded by prestigious Gdel Prize in 2003. The boosting based algorithm gained a great success for its generalization performance. It is the choice for the winners of the KDD Cup from 2007 to 2014 and it still plays an important roles in many fields with unstructured data or data with limited size, where the deep learning method failed to provide convincing results [173].

As a meta-algorithm, the framework of boosting is described in Algorithm 1. The performance of boosting depends on the choice of how to find the weak learner, how to calculate α^t for each weaker and how to update to weight for the data set.

Algorithm 1 The Boosting Framework

```
Initialize the weight for the whole dataset D
for t = 1 : T do
    Normalize the weight
    Calculate the weak learner h^t based on the current weight
    Calculate the weight α^t for the current weak learner h^t based on its performance on D
    Update the weight for D
end for
Output the strong learner by weighed sum H^T = Σ_{t=1}^{T} α^t h^t
```

2.3.1 Relation to Functional Gradient Descent

Although the motivation of boosting is to answer the question posed by Kearns and Valiant that if a set of weak learners create a single strong learner, recent studies revealed that boosting could be fitted into the Loss Minimization framework, where the learning problem is formulated as an optimization problem on the data set D

$$L(F) = \frac{1}{|D|} \sum_{i=1}^{|D|} l(F, x_i) \tag{2.2}$$

where $l(F, x)$ is the loss function and F is the function describing the learner. For example, in Adaboost, $l(F, x)$ is an exponential function that upper bounds the 0-1 loss.

To optimize (2.2), gradient descent is a standard approach, where in every iteration we take the small steps in the direction of steepest descent.

$$F^t = F^{t-1} - \alpha^t \bigtriangledown L(F^{t-1}) \tag{2.3}$$

Therefore, find one weak learner in boosting is equal to calculating the gradient in the function space $\mathcal{F}$. However, since the weak learner always fall into some certain family of function, for example the decision tree or linear classifier, it may not be feasible to find a weak learner f that is in the direction of the gradient. Instead, we could choose the weak learner that is closest to the negative gradient by maximizing the inner product with the negative gradient on the whole training data set

$$- \bigtriangledown L(F^{t-1}) \cdot f = -\sum_{i=1}^{|D|} \frac{\partial l(F, x_i)}{\partial F(x_i)} f(x_i) \tag{2.4}$$

2.3.2 Difficulty in Distributed Boosting

Although the Boosting algorithm provides good generalization performance and easy implementation, there remain challenges to design the distributed boosting algorithm. In essential, boosting is a sequential algorithm, where in each iteration, the calculation of the current weak learner is based on the previous learners. Besides, to guarantee the convergence, as mentioned in (2.4), calculating the weak learner requires to go through the entire data set, which makes it hard to distribute the computation.

To overcome the difficulty, we could borrow from the idea of stochastic gradient descent. Instead of calculating the weak learner that maximize (2.3), we seek to construct a small data set S that could approximate the whole data set D, such that by solving

$$- \bigtriangledown L(F^{t-1}) \cdot h = -\sum_{i=1}^{|S|} \frac{\partial l(F, x_i)}{\partial F(x_i)} h(x_i) \tag{2.5}$$

h is close to the f in (2.3). Since we only need to transmit the small subset S, it is possible to design the communication efficient distributed boosting.

2.4 Coreset

Coresets are small, weighted summaries of large data sets that has close performance on specific metric comparing to the full data set. Coresets originated in the field of computational geometry and are closely related to the fundamental concepts of ϵ-net and ϵ-approximation. Due to strong composability properties, coresets is suitable for parallel constructions which leads to practical implementations in the context of large data sets.

The coreset construction is similar to the importance sampling. In importance sampling, given a random variable X with known distribution $\mathcal{P}$ and target function $f(x)$, we want to estimate $E[f(X)]$. If we have i.i.d sample generated according to $\mathcal{P}$, then we could calculated the unbiased estimator

$$E[f(x)] = \frac{1}{n}\sum_{i=1}^{n} f(x_i) \tag{2.6}$$

When it is hard to generate the sample according to $\mathcal{P}$, instead, we use another distribution $\mathcal{Q}$ to generate the sample and calculate the expectation as the weighted sum

$$E[f(x)] = \frac{1}{n}\sum_{i=1}^{n} \frac{p(x_i)f(x_i)}{q(x_i)} \tag{2.7}$$

We could prove that (2.7) is the unbiased estimator for $E[f(x)]$. In coreset construction, we carefully select the sample from the whole data set such that the desired metric (gradient, distance, cluster center) on the select sample $\mathcal{S}$ is close enough to that on the whole data set $\mathcal{D}$ for a family of functions $f \in \mathcal{F}$, as described in (2.8),

$$|\frac{1}{|D|}\sum_{i=1}^{|\mathcal{D}|} f(x_i) - \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|} w_i f(x_i)| \leq \epsilon, \quad \forall f \in \mathcal{F} \tag{2.8}$$

with high probability.

While importance sampling is optimized to find the sampling weight to minimize the variance of the estimator for one fixed function f and the distribution of the data is known, the coreset construction requires the sample to have close performance to the whole data set for a family of functions and we only have the access to the data set. The coreset approaches were impractical as the naive construction of coreset requires the computation exponential to the order of data dimension. Recent breakthrough by [79] enabled efficient coreset construction algorithms via random sampling and encouraged the researchers to explore its application on machine learning problem.

Chapter 3

Message Passing Based Distributed Learning for Joint Resource Allocation in Millimeter Wave Heterogeneous Networks

3.1 Introduction

The millimeter wave (mmWave) technology is expected to be the new frontier for 5G communication cellular systems that offers greater bandwidths and faster data rates. However, the unique radio propagation characteristics of mmWave are challenging the design of wireless communication systems. The high space path loss and the blockage effect may require high densities of transmitters, while the latter, along with the highly directional transmission, can cause rapid quality variations. The mmWave channel may change between Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS) in the order of hundreds of milliseconds or even less [113], much faster than comparable technologies such as 4G LTE or IEEE 802.11. The link between base station(BS) and user equipment(UE) may therefore suffer serious instability.

Heterogeneity is expected to be the one of the key features of mmWave networks for the above issues. As demonstrated by recent channel measurements[116], mmWave could be used for outdoor communications over a transmission range of about 150-200 meters. Thus the high-power and low-density macro-cell BSs could be replaced along with denser but lower power small-cell BSs, which could help cover the NLOS region for large-cell BSs and provide load balancing via user association.

The resource allocation in mmWave HetNets is a critical problem. Due to the large density of BSs, the intercell interference requires proper power control for BSs to maximize the system throughput while avoiding substantial interference. Moreover, the rapid switch between LOS and NLOS states demands advanced user association strategy that considers the BS selection cost and decision-making efficiency. Since separate power control and user association may lead to a suboptimal resource allocation [161], the theoretically optimal approach is to solve the problem jointly.

Several recent studies have addressed the related topic in mmWave networks [134]. For user association in mmWave networks, [60] investigates the user association by deriving an optimal and fair cell selection policy and considering the reallocation cost. The authors assumed that a UE is connected to the nearby BSs simultaneously, and the data rates for each link between the BSs and the UE are calculated by solving an optimization problem with a bandwidth constraint. [147] considered the BS selection in mmWave HetNets by modeling a multi-armed bandit problem and developing an online learning policy to connect UEs to the optimal BSs. The distribution of throughput of BS is fixed and the UE is required to estimate the expectation with the minimum samples. An extensive survey about the user association in mmWave is detailed in [89]. For the interference coordinated scheduling in mmWave networks, [40] proposed a generic mathematical framework to analyze the multi-tier mmWave cellular networks. In [30] the spatial-time domain resource allocation is performed, which gives consideration to throughput and fairness. [54] explored the potential gain of ultra-densification for enhancing mmWave communications from a network-level perspective.

The reinforcement learning (RL) approach has been applied in a variety of schemes such as routing, resource allocation and dynamic channel selection in wireless networks [159]. Due to the stochastic nature of the channel, many problem in wireless network that modeled by

stochastic optimization could be considered as a Markov decision process. Reinforcement learning could solve the MDP in an online manner with little prior knowledge. A framework for handover decisions based on MDP for mmWave is proposed in [102]. It models the channel state as a Markov chain and uses dynamic programming to solve the bandwidth allocation problem. However, it does not consider the power control, and the state space is exponential with respect to the number of UEs, which limits its practical application. [130] proposed decentralized procedures for joint interference management and cell association for LTE network. [88] applied reinforcement learning to implement a dynamic channel selection by minimizing external interference. [101] proposed a rigorous and unified framework for simultaneously utilizing both physical-layer-centric and system-level techniques to achieve the minimum possible energy consumption, under delay constraints and [92] applied multi-agent methods for spectrum sensing.

However, since for joint resource allocation problem the number of agents (BSs and UEs) is large and the size of state space is exponential in terms of the number of agents, it is impractical to handle the learning process with traditional RL approach. To reduce the complexity, some previous work that employed RL for the resource allocation problem suffered from two major drawbacks, which prevent their applications in mmWave networks. [88], [101] modeled UEs and BSs as independent agents without collaboration, which fall in the single-agent reinforcement learning (SARL) framework. Previous theoretical analysis revealed that in SARL the agents may change their respective actions frequently, or oscillate between actions, such that the convergence to the optimal solution is not assured [159]. [99] applied the multi-agent method to solve the problem. However, their solution is in a centralized style where the policy could be updated after the BS obtaining all the channel states from all the possible links. As mmWave channel may change in the order of hundreds of milliseconds or less, there is little time to aggregate the channel state information of all UEs. For efficient decision making, we believe such centralized learning method is not suitable in mmWave networks.

In this paper, by employing the tools of RL along with the distributed message passing method, we study the downlink of heterogeneous mmWave cellular networks with the

incorporation of the distinguishing features of mmWave. Our main contributions can be summarized as follows:

- We model the interference coordination and user association problem jointly in HetNets to minimize the time averaged risk-averse rate by considering the reallocation cost, which is later transformed into a multi-agent RL problem.

- A sparse coordination graph is constructed according to the connectivity of the BSs and UEs within the cell. The state/action space is decomposed based on the structure of the graph. BS-centric and UE-centric decomposition are proposed separately to deal with different setups in mmWave HetNets, where the efficiency or power consumption is the priority.

- The distributed message passing method is introduced to solve the multi-agent RL problem based on the sparse coordination graph, which is motivated by the approach of belief propagation in probabilistic graphical models. We use an efficient approximate algorithm for inference with incremental changes in the graphical model.

- We utilize the prior knowledge about the mmWave network, as well as the transition probability of the link state, to generate good exploratory behaviors using planning. The learning process is further accelerated by combining the resulted behavior and environment interaction.

- We collect real-world measurements for the channel statistics, using our mmWave testbed, for simulations. The performance of the proposed framework is presented both in throughput and power consumption.

The remainder of the paper is organized as follows. In Section 3.2, system model is introduced. In Section 3.2.4, we model the wireless network as a coordination graph and introduce the decomposition of the state/action space according to the graph. In Section 3.4, the distributed message passing method is applied to solve the max-sum problem introduced by the decomposition. In Section 3.5, we propose the model based method to accelerate the learning procedure. Numerical results are provided in Section 3.6. Finally, conclusions and future work are provided in Section 3.7.

3.2 System Model

In this section, a model is built for a two-tier heterogeneous downlink mmWave network.

3.2.1 Deployment Model

We assume that a macrocell $\mathcal{B}$ is located at the origin and multiple picocell BSs $\mathcal{P}$ operate in the same frequency band with different transmission powers. The picocell BSs are deployed in the edge region and denote by $\mathcal{M} = \mathcal{P} \cup \mathcal{B}$ all the BSs within the macrocell. We consider a number of UEs distributed uniformly in $\mathcal{R}^2$, according to homogeneous Poisson point processes (PPSs) with density λ_U.

3.2.2 Channel Model

In this paper, we assume each link between UE u_i and BS m_j is characterized by two state variables $\{l_{ij}, G_{ij}\}$.

- $l_{ij} \in \{LOS, NLOS, outage\}$ indicates if there is a direct mmWave link between UE i and BS j. However, according to recent results on mmWave channel modeling [4], an additional outage state should be considered for the link state when no link is established between the BS and the UE due to the blockage. Given the link of length r, define $p_l(r)$, $p_n(r)$, $p_o(r)$ as the probability that the link is LOS, NLOS and outage accordingly. Similar to the 3GPP-based models[95], we approximate the probability function with a ball model. If r_{ij} is within the radius $\mathcal{R}$, the distribution for l_{ij} is $\{p_L^1, p_N^1, p_O^1\}$ and if r_{ij} is larger than $\mathcal{R}$, the corresponding distribution is $\{p_L^2, p_N^2, p_O^2\}$. Based on l_{ij}'s distribution and previous measurements, the transition probability matrix for link state l_{ij}

$$\boldsymbol{\mathcal{P}}_l = \begin{bmatrix} p_{LOS|LOS} & p_{LOS|NLOS} & p_{LOS|Out} \\ p_{NLOS|LOS} & p_{NLOS|NLOS} & p_{NLOS|Out} \\ p_{Out|LOS} & p_{Out|NLOS} & p_{Out|Out} \end{bmatrix} \tag{3.1}$$

could be calculated accordingly.

- $G_{ij} \in \{G_0, G_1, G_2\}$ indicates the beam alignment state for the link. From the realistic point of view, since the set of beam patterns is discrete, perfectly tracking the beam with arbitrary direction may be too costly or even impossible, we assume that the alignment state may switch between multiple discrete states: When both transmitter and receiver are well aligned, $G = G_0$. When either transmitter or receiver is well aligned, $G = G_1$ and when none of transmitter or receiver is well aligned, $G = G_2$. We assume G_{ij} is a random variable and its distribution could be estimated as prior from previous measurements.

In practice UE is able to estimate the LOS/NLOS link states the neighboring BSs between two time slots by channel estimation and acquire the alignment information by checking the tracking error of the beam direction, for example, in [165].

The path loss between UE u_i and BS m_j is

$$L(r_{ij}, l_{ij})(dB) = \rho + \alpha_{l_{ij}} \log_{10}(|r_{ij}|) + \chi_{l_{ij}} \tag{3.2}$$

where $\alpha_{l_{ij}}$ is the path loss component and $\chi_{l_{ij}} \sim \mathcal{N}(0, \xi^2_{l_{ij}})$ is the shading random variable given link state l_{ij}. ρ is the path loss at 1m. Note that LOS and NLOS have different path loss components and fading variables and outage has infinity path loss.

The received signal power for UE u_i from BS m_j is given by

$$P_{ij} = G_{ij} P_j L^{-1}(r_{ij}) \tag{3.3}$$

where P_j is the transmission power from m_j and is could be controlled with multiple discrete levels.

3.2.3 Interference Model

Assume that UE u_i is connected to BS m_j, then the signal-to-inference-noise-ratio (SINR) for u_i is given by

$$SINR_{ij} = \frac{P_{ij}}{\sigma^2 + \sum_{k \neq j} P_{ik}}. \tag{3.4}$$

The corresponding instantaneous rate is given by

$$c_{ij} = \frac{W}{N_j} \log(1 + SINR_{ij}) \tag{3.5}$$

where W is the total available bandwidth, and N_j is the number of UEs connected to BS m_j. It is assumed that the bandwidth is shared equally among all UEs connected to that BS, for simplicity. Define $x_{i,j}$ as the variable indicating the connectivity between UE u_i and BS m_j. $x_{i,j} = 1$ if u_i is connected to m_j. The instantaneous rate for u_i could be

$$R_i = \sum_{m_j \in \mathcal{M}} x_{i,j} c_{ij} \tag{3.6}$$

The time averaged rate to UE u_i is given by

$$\boldsymbol{E}[R_i] = \frac{1}{T} \sum_{t=1}^{T} R_i^t \tag{3.7}$$

In LTE networks, the time averaged rate could be a good metric for system performance since the channel is steady, which is not the case in mmWave networks. A UE may suffer from a short-time poor channel, if the link is NLOS or the beams are not well aligned, even though the overall rate is high. We thus use a time averaged risk averse rate given by

$$AR_i := -\frac{1}{\theta} \log \boldsymbol{E}[\exp(-\theta R_i)] \tag{3.8}$$

when $\theta \to 0$, $AR_i \to \boldsymbol{E}[R_i]$, the time averaged rate. When $\theta \to +\infty$, $AR_i \to \min R_i^t$, the minimum rate. Increasing θ may increase penalty to the short-term drops in rate.

We assume that each UE is connected a single BS at one time slot. In this paper, we want to find the optimal user association policy as well as the transmission power for each BS such that the overall time averaged risk averse rate is maximized. Since frequent handovers may increase the power consumption for UE, we add the penalty term in the objective. Define q_i^t as the variable indicating if the UE u_i's association is changed in time slot t. Then, the optimization problem is formulated as

$$
\begin{aligned}
\max_{P_j, x_{i,j}} \quad & \sum_{u_i \in \mathcal{U}} (AR_i - \gamma \boldsymbol{E}(q_i)) \\
\text{s.t.} \quad & \sum_{m_j \in \mathcal{M}} x_{i,j} = 1, \forall u_i \in \mathcal{U} \\
& x_{i,j} \in \{0, 1\} \\
& P_j \in [0, P_{max}]
\end{aligned}, \tag{3.9}
$$

Solving (5.2) directly is NP-hard. In previous research, to solve such utility maximization problem, the unique association constraint is relaxed to multi-connectivities[161]. However, this may require more overhead to implement and may not be practical. As the non-deterministic transitions between individual link states are Markovian, it is suitable to apply RL approach to solve this problem.

3.2.4 Q-learning

Q-learning [150] is used to find the optimal state-action policy for finite state MDPs. It has been applied to many fields for its guaranteed convergence to the optimal policy. Q-learning problems are characterized by the agent with its state $\mathcal{S}$, the set of action $\mathcal{A}$ per state and the reward $\mathcal{R}$. A policy is the agent's choice of actions for each state. The goal of Q-learning is to find the optimal policy that maximizes the expected value of the total reward over all successive steps, namely

$$
Q(s, a) = E\left\{ \sum_{t=0}^{+\infty} \beta(t) r(s^t, a^t) | s^0, a^0 \right\} \tag{3.10}
$$

where $Q(s, a)$ is the metric of the state-action pair (s, a), $\beta(t) = \beta^t$ and $0 < \beta < 1$ is a discounting factor. r is the reward function. For the system considered in this paper, the basic definition of the agent, state, action and reward functions are as follows.

- **Agent**: UEs and BSs.

- **State**: Each type of agent has its own state space as s_u and s_m. For BS j, the state is its transmission power and the number of connected UEs.

$$s_{m_j} = \{P_j, N_j\} \tag{3.11}$$

The transmission power is assumed to be discrete.

For UE, the state is its link state to the nearby BS, which is given by

$$s_{u_i} = \{l_{i,1}, ...l_{i,n}, G_{i,0}, ...G_{i,n}\} \tag{3.12}$$

where G_{ij} and l_{ij} is the alignment state and link state between u_i and m_j introduced in Section 3.2.2. n is the number of nearby BSs which u_i is within their transmission ranges.

- **Action**:

$$\{a_{m_j}, a_{u_i}\} = \{P_j, (x_{i,1}..., x_{i,n})\} \tag{3.13}$$

where BS could control its transmission power with multiple discrete levels and UE could choose which BS to connect.

- **Reward**: The total rewards at time slot t is

$$r(s_t, a_t) = \sum_{u_i \in \mathcal{U}} \left(\sum_{m_j \in \mathcal{M}} x_{i,j} AR_i - \gamma q_i^t \right) \tag{3.14}$$

The Q-learning problem is solved by updating the Q-function based on the interaction with the environment. The standard algorithm for updating the Q-function at iteration t is given by

$$\begin{aligned} & Q(s^t, a^t) \\ = \; & (1-\alpha)Q(s^t, a^t) + \alpha\big[r(s^t, a^t) + \beta \max_{a^{t+1}} Q(s^{t+1}, a^{t+1})\big] \end{aligned} \tag{3.15}$$

where α is the learning rate. At iteration t, the agent explores the environment by the optimal action and get the reward when entering the next state s^{t+1}. Then it searches the Q-table given s^{t+1} and finds the optimal action a_{opt}^{t+1}. The current Q-function for s^t is updated according to (3.15).

Standard Q-learning considers the system as a single agent. For the HetNet considered in this paper, if we model the whole system as the single agent, the overall state space is given by

$$\mathcal{S} = \boldsymbol{s_{\mathcal{M}}} \times \boldsymbol{s_{\mathcal{U}}} \tag{3.16}$$

where

$$\boldsymbol{s_{\mathcal{M}}} = s_{m_1} \times s_{m_2} ... \times s_{m_{|\mathcal{M}|}} \tag{3.17}$$

and

$$\boldsymbol{s_{\mathcal{U}}} = s_{u_1} \times s_{u_2} ... \times s_{u_{|\mathcal{U}|}} \tag{3.18}$$

.

Similarly the overall action space is given by

$$\mathcal{A} = a_{\mathcal{M}} \times \boldsymbol{a_{\mathcal{U}}} \tag{3.19}$$

The state/action space size increases exponentially with the number of UEs. For a cell with 100 UEs, the overall size for the state space could be more than 10^{30}. Finding the globally optimal action becomes computational intractable as the Q-table is too large to handle. In the next two sections, we will solve the problem in multi-agent reinforcement learning framework and decompose the state/action space according to the topology of the mmWave network.

3.3 State/Action Space Decomposition

In this section, we model the mmWave network as a coordination graph based on the connectivity of the agents and then decompose the large Q-function as the sum of multiple

$$Q(\boldsymbol{s},\boldsymbol{a}) = \sum_{u_i \in \mathcal{U}} Q(s_i, \boldsymbol{s}_{\Gamma(u_i)}, a_i, a_{\Gamma(u_i)}) + \sum_{m_j \in \mathcal{M}} Q(s_j, \boldsymbol{s}_{\Gamma(m_j)}, a_j, a_{\Gamma(m_j)}). \tag{3.20}$$

$$Q(s_i^t, a_i^t) = (1-\alpha)Q(s_i^t, a_i^t) + \alpha[r(s_i^t, a_i^t) + \beta Q(s_i^{t+1}, a_i')] \tag{3.21}$$

$$Q(\boldsymbol{s}^t, \boldsymbol{a}^t) = \alpha \sum_{i \in \mathcal{U} \cup \mathcal{M}} r(s_i^t, a_i^t) + (1-\alpha)Q(\boldsymbol{s}^t, \boldsymbol{a}^t) + \alpha\beta\big[\sum_{u_i \in \mathcal{U}} Q(s_i^{t+1}, a_i') + \sum_{m_j \in \mathcal{M}} Q(s_j^{t+1}, a_j')\big] \tag{3.22}$$

$$Q(\boldsymbol{s},\boldsymbol{a}) = (1-\alpha)Q(\boldsymbol{s},\boldsymbol{a}) + \alpha \sum_{l_{i,j} \in \boldsymbol{E}} \big[r(s_i, s_i, a_{i,j}) + \beta Q(s_i, s_i, a_{i,j}')\big] \tag{3.23}$$

$$Q(s_i, s_j, a_{i,j}) = (1-\alpha)Q(s_i, s_j, a_{i,j}) + \alpha\big[r(s_i, s_i, a_{i,j}) + \beta Q(s_i, s_i, a_{i,j}')\big] \tag{3.24}$$

---

small Q-functions.

$$Q(\boldsymbol{s},\boldsymbol{a}) = \sum_i Q_i(\boldsymbol{s_i}, \boldsymbol{a_i}) \tag{3.25}$$

We call $Q(\boldsymbol{s},\boldsymbol{a})$ **Global** Q-function as it contains the whole state/action space while $Q_i(\boldsymbol{s_i},\boldsymbol{a_i})$ is called **Local** Q-function as it only contains a subset of state/action space. The decomposition is further classified into two categories: the agent based and the edge based decomposition. The former results in a UE-centric pattern which requires more computation cost but provides faster convergence. It is suitable for mobile communications since UE, usually the smart phone, could provide sufficient computational power. The latter decomposition results in the BS centric pattern where the BS collects all the state information from UEs and completes all the computation. This is especially desirable for the application of Internet of Things (IoT) where the UEs are more power limited.

3.3.1 Coordination Graph

Coordination graph [61] exploits the fact that in many multi-agent problems only a few agents depend on each other and thus the large problem could be decomposed to simpler sub-problems. In a coordination graph, each node represents an agent and each edge defines the coordination dependency between the connected nodes[62, 78]. As shown in Fig 3.1, in mmWave networks, UEs and BSs are the nodes in the graph. Each UE's association policy only has the direct dependency on the channel conditions corresponding to their nearby BSs due to the high path loss and blockage in mmWave network. Therefore in coordination graph, the UE nodes are only connected to its nearby dependent BSs. To manage the interference, the power control of the BS also relies on the transmission power of the nearby BSs. Thus we also have the BS-BS edge within the coordination graph. Although we omit most of the UE-BS edges in Fig 3.1 for clear demonstration, the true connection of the graph is still sparse. This is because no edge exists between two UEs as they have no direct dependency. The averaged degree of the node is less than n+1 when we assume that each UE is visible to n nearby BSs. We will utilize the graph's sparsity and design the message passing algorithm to solve the RL problem in a distributed manner in next section. In this section, we first introduce agent-based and edge-based decomposition to handle the large state/action space in mmWave networks and formulate the distributed multi-agent RL problem.

3.3.2 Agent-Based Decomposition

In agent-based decomposition, each node has its own local Q-function $Q(s_i, \boldsymbol{s}_{\Gamma(i)}, a_i, a_{\Gamma(i)})$, which is defined by the state of local agent $\{s_i, a_i\}$, its nearby agents' states as well as their actions $\{\boldsymbol{s}_{\Gamma(i)}, a_{\Gamma(i)}\}$. Here $\Gamma(i)$ are the set of neighboring nodes for node i and $\boldsymbol{s}_{\Gamma(i)}$ is the state space of all the nearby nodes of node i:

$$\boldsymbol{s}_{\Gamma(i)} = \{s_n \times ... \times s_m\}, n, m \in \Gamma(i) \tag{3.26}$$

This is shown in Fig 3.2 for an example of an agent-based decomposition for a 4-agent problem.

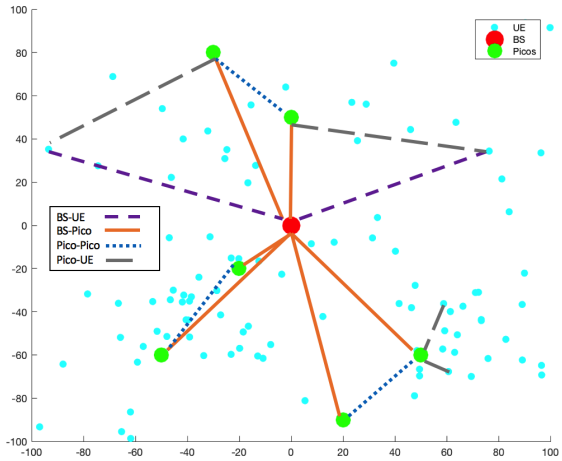


Figure 3.1: Coordination graph for macrocell mmWave network

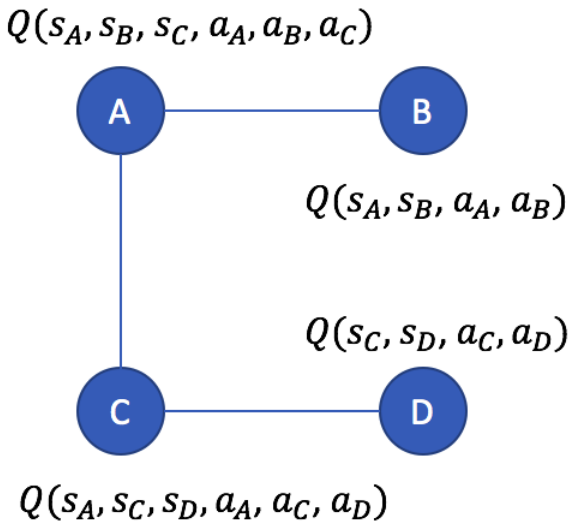


Figure 3.2: Agent-based Decomposition

In the coordination graph considered in this paper, we have two classes of agents: UE and BS. Thus the global Q-fucntion $Q(\boldsymbol{s}, \boldsymbol{a})$ could be written as (3.20), where $\Gamma(u_i)$ is the set of BSs near UE u_i and $\Gamma(m_j)$ is the set of UEs near BS m_j. We denote $(s_i, \boldsymbol{s}_{\Gamma(i)}, a_i, a_{\Gamma(i)})$ by (s_i, a_i) for notational simplicity. Note that in (3.20), the optimal action is obtained based on the Global Q-function.

$$\boldsymbol{a}' = \arg\max_{\boldsymbol{a}} Q(\boldsymbol{s}^{t+1}, \boldsymbol{a}) \tag{3.27}$$

In next section, we prove that it is non trivial to solve (3.27) and design the efficient distributed message passing algorithm to find $\boldsymbol{a}'$. In this section, we assume that each agent has the access to the solution of (3.27).

Then for each local Q-function, the update follows the similar procedure to the classic Q-learning as in (3.21). Since the network gains reward when UE chooses to connect to one of its neighboring BSs, while each agent receives penalty if it switches to a different BS, the reward functions for UE u_i and BS m_j in (3.21) are

$$r(s_j, a_j) = \frac{1}{2} \sum_{u_i \in \Gamma(j)} x_{i,j} AR_i \tag{3.28}$$

and

$$r(s_i, a_i) = \frac{1}{2} \sum_{m_j \in \Gamma(i)} x_{i,j} AR_i - \gamma q_i \tag{3.29}$$

The reward is equally allocated to u_i and m_j and there is a 1/2 factor in (3.28) and (3.29).

Note that in (3.21) the agent needs to collect the current state information s_i, local reward $r(s_i, a_i)$ and the global optimal action a_i'. There is no need for the agent to know the state information from all other nodes. The update of local Q-function is based on local observations; therefore the computation is completely distributed.

Now we prove that given the definition of the reward function in (3.28), (3.29), updating the local Q-function in a distributed manner is equivalent to updating the global Q-function.

Theorem 3.1. *Suppose that each agent in the coordination graph only stores its local Q-function and receives the local reward and states from neighboring agents. All the distributed agents have the access to the globally optimal action* $\boldsymbol{a'}$*. Then the sum of local updating procedure defined in (3.21) is equivalent to updating the global Q-function in (3.15).*

Proof (3.22) is the direct result of (3.20) and (3.21). Then we have

$$\begin{aligned}\sum_{i\in\mathcal{U}\cup\mathcal{M}} r(s_i^t, a_i^t) &= \sum_{i\in\mathcal{U}} r(s_i^t, a_i^t) + \sum_{j\in\mathcal{M}} r(s_j^t, a_j^t) \\ &= \sum_{u_i\in\mathcal{U}} (AR_i - \gamma q_i) \end{aligned} \tag{3.30}$$

is the global reward and

$$\begin{aligned} & \sum_{u_i\in\mathcal{U}} Q(s_i^{t+1}, a_i') + \sum_{m_j\in\mathcal{M}} Q(s_j^{t+1}, a_j') \\ = \; & Q(\boldsymbol{s}^{t+1}, \boldsymbol{a}') \end{aligned} \tag{3.31}$$

Since $\boldsymbol{a}'$ satisfies (3.27), we could prove that (3.22) and (3.15) are equivalent.

Algorithm 2 summarizes the implementation of the proposed RL framework.

Remark: It is not practical for BSs to keep track of all the neighboring UEs' states and rewards, especially when the BS does not transmit data to the UE or the link is not stable. However, as long as the UE is connected to one of the BSs, its state information could be exchanged between BSs through backhaul communication links. On the other hand, since typically the number of neighboring BSs for one UE is 3-6 in urban area [116], we assume that the UE is able to track the link states for all nearby BSs.

The direct advantage of the agent-based decomposition is that agents use the local rewards instead of the global reward to update the local Q-function. In comparison, [92] uses the global reward to update the local Q-function and the agents are unable to distinguish which agents are responsible for the received global reward. The propagation of the globally optimal action $\boldsymbol{a'}$ delivers the global information throughout the graph. Besides, the update of the Q-function is completely distributed. This is suitable for the UE centric control when the UE is capable of computing the Q-function.

Algorithm 2 Distributed Collaborative Q-learning

```
Initialize s_0 for all agents.
∀i, find {j | arg_j max P_{i,j}} and set x_{i,j} = 1.
for t = 1:T do
    for Each UE do
        if Agent-based then
            Obtain (s_i^t, s_{Γ(i)}^t) from neighboring BSs.
        else if Edge-based then
            Transmit s_i^t to the neighboring BSs.
        end if
    end for
    Obtain the globally optimal action a' = MessageQ(s^t).
    if Agent-based then
        UEs and BSs execute a' with ϵ-greedy.
    else if Edge-based then
        BSs inform UEs a'. a' is executed with ϵ-greedy.
    end if
    Obtain reward r^t from the measurement.
    if Agent-based then
        UE and BS calculate the local Q-function based on the reward.
    else if Edge-based then
        BSs calculate the local Q-function based on the reward from UEs.
    end if
    if Use model-based acceleration then
        Q(s, a) = Acce(s, a, M).
    end if
end for
```

3.3.3 Edge-based Decomposition

Although the agent-based decomposition could efficiently reduce the overall state/action space, the computation for updating local Q-function still requires the summation over all possible state/action pairs for each UE. For a typical UE with 3 neighboring BSs, the size for local Q table is $O(|s|^3)$, where s is the size of the state space. This could be easily handled by modern mobile devices. However, for the application of mmWave communication in IoT, the power consumption becomes the critical issue. In this case, it is preferable that the BS could complete all the computation and send the control signal back to UE. We propose an edge-based decomposition RL framework for such a context.

In edge-based decomposition, the global Q-function is the sum of all the local Q-functions defined on the edges of the graph, as shown in Fig 3.3, namely

$$Q(\boldsymbol{s}, \boldsymbol{a}) = \sum_{x_{i,j} \in \boldsymbol{E}} Q(s_i, s_i, a_{i,j}) \tag{3.32}$$

where $\boldsymbol{E}$ is the set of edges in the graph. The updating rule for local Q-function is given by (3.23), where

$$\boldsymbol{a'} = \arg\max_{\boldsymbol{a}} \sum_{x_{i,j} \in \boldsymbol{E}} Q(s_i, s_j, a_{i,j}) \tag{3.33}$$

Here, $a_{i,j} = \{a_i, a_j\}$ is the joint action for agent i and j. Similarly, combining (3.32) and (3.23) we have (3.24)

The reward function is given by

$$r(s_i, s_j, a_{i,j}) = -\gamma \frac{1}{N_m} q_i + AR_i \tag{3.34}$$

where N_m is the number of neighboring BSs for each UE.

The key difference between the agent-based and edge-based decomposition is that the former requires all the nodes (UEs and BSs) to collect the neighboring nodes' state and update the local Q-functions in a distributed manner, while the latter only requires the UEs

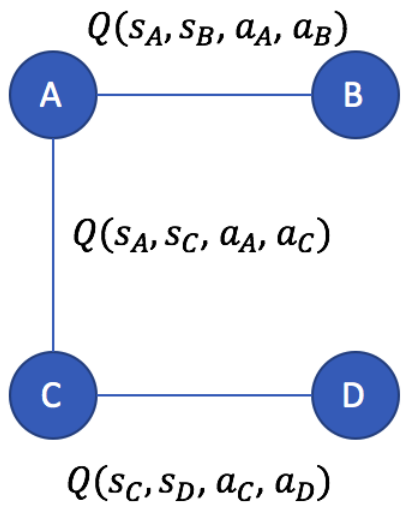


Figure 3.3: Edge-based Decomposition

to send their states and rewards to the associated BSs and the computation is completed by the BSs. BS is less sensitive to the power consumption for computation. The disadvantage for the edge-based decomposition is that the number of local Q-functions is equal to the number of edges in the graph, which could be larger than the number of local Q-functions in the agent-based decomposition. Since the response time is directly related to the computation cost, the edge-based framework provides lower power consumption cost for UEs at the cost of slower response time.

3.4 Distributed Message Passing on Coordination Graph

3.4.1 Max-sum Problem

Although the update of local Q-functions in Section 3.2.4 is based on local reward and state, the globally optimal action $\boldsymbol{a'}$ under the next state $\boldsymbol{s^{t+1}}$ is required, as mentioned in (3.21)

Table 3.1: Comparison between Markov random field and coordination graph

	Markov Random Field	**Coordination Graph**
Problem	Sum-Product	Max-Sum
Node	Random Variable	Agent(BS,UE,etc)
Connection	Between dependent variables	Between linked agents
Sparsity	Yes	Yes

and (3.23). Finding $\boldsymbol{a'}$ is equal to solving the follow max-sum problem:

$$\arg\max_{\boldsymbol{a}} \sum_{k} Q(\boldsymbol{s_k}, \boldsymbol{a_k}) \tag{3.35}$$

where $\boldsymbol{s_k}$ and $\boldsymbol{a_k}$ are the subsets of the overall state/action set. We drop $\boldsymbol{s}$ in the notation since it is fixed when solving the max-sum problem.

Local Q-functions in the coordination graph share joint variables a if they are connected in the graph. It is nontrivial to find the global action $\boldsymbol{a}$ to maximize the global Q-function as $\max_{\boldsymbol{a}} \sum Q(a_i, a_j) \geq \sum \max_{a_i,a_j} Q(a_i, a_j)$. To solve the problem, the straightforward idea is to transfer all the local Q-functions to one BS and solve it in a centralized manner by variable elimination. This is inefficient, since it requires transferring all the local Q-function tables through communications, and the computation cost of variable elimination is exponential in the degree of node.

However, observing that our graph is sparse and inspired by the fact that the coordination graph is similar to a Markov random field, as shown in Table 3.1, we propose a message passing algorithm to solve the max-sum problem in proposed distributed RL framework. This is similar to the belief propagation in marginalizing the joint distribution of Markov random fields, as shown in Fig 3.4.

The message passing method has achieved substantial success in many statistical inference problems, such as LDPC decoding [96] and image denoising [138]. Briefly speaking, the message passing method is carried out by iteratively sending locally optimized messages to neighboring nodes. Although in theory it only guarantees to converge when the graph is free of cycles, its empirical results on graphs with cycles in practical problems are surprisingly excellent.

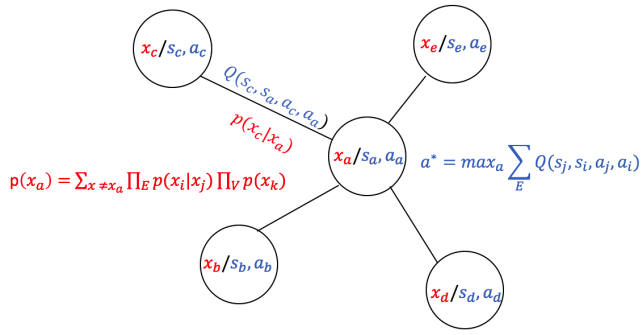


Figure 3.4: Coordination graph and Markov random field

Algorithm 3 *Message Passing Algorithm*

Required: The current state $\boldsymbol{s^t}$
Initialize all the message $\mu_{i,j}(a)$
while Have not converged for all agents **do**
  for Every agent i **do**
    Send message to neighboring agent j according to (3.36)
    if $\mu_{i,j}$ differs from previous less than the threshold **then**
      Agent i converges
    end if
  end for
end while
Output: Compute the optimal action using (3.37)

Since we use Q-table to represent the Q-function, the message from agent i to agent j is a table of the action, which is defined as

$$\mu_{i,j}(a_j) = \max_{a_i} \left\{ Q_{i,j}(a_i, a_j) + \sum_{k \in \Gamma(i)/j} \mu_{k,i}(a_i) \right\} - c_{i,j} \tag{3.36}$$

where $\Gamma(i)/j$ is the neighbor of i except j, and $c_{i,j}$ is the normalizing factor [145] to guarantee the convergence. After the convergence, the globally optimal action for agent i is obtained by locally solving

$$a_i^* = \arg\max_{a_i} \sum_{k \in \Gamma(i)} \mu_{k,i}(a_i) \tag{3.37}$$

With message passing, instead of sending the whole local Q-table, the node communicates with each other by sending the message $\mu_{i,j}(a_i)$, whose size is equal to the number of possible actions for single node. This efficiently reduces the communication cost.

Algorithm 4 *Belief Propagation for Repeated Inference*

```
Required: The graph initially solved by Algorithm 3 and a set of changed nodes Δ
while Δ is not empty do
    for Every node i in Δ do
        Generate message μ_{i,j} and compare with previous
        if mess_i differs from previous larger than threshold then
            Node i sends μ_{i,j} to its neighbors j and add its neighbors Γ(i) to Δ
        end if
    end for
end while
```

3.4.2 Efficient Belief Propagation for Repeated Inference

Assume that each UE is connected to one BS at any time slot and is assigned dedicated uplnk control resources similar to the Physical Uplink Control Channel (PUCCH) in LTE. These resources can be used for UE to periodically report the SNR and other channel characteristics, such as the alignment state G, link state l to BS. As indicates in [71] and our simulation, the belief propagation may takes 80 to 100 iterations to converge given the scale of typical mmWave network.

As in mmWave communication it is expected that the length of subframe is less than 1ms, the total delay caused by the message passing is less than 100 ms. Although the proposed framework is only sensitive to the link state(LOS/NLOS) instead of the exact SNR and thus it is robust to the short-term fading as the link state within 100ms could be considered as fixed[55], it is still possible that some of the links may switch their link states within the inference process. The system is thus suboptimal if the inference is slow. To further speed up the belief propagation, we apply the similar idea in [103] utilizing the fact that when some link states change, they typically only affect a small region of the graph and it is not necessary to update the whole graph.

The algorithm starts after the graph is initially solved by a standard BP. Each time some nodes' states change, it maintains the set of changed node Δ. In the following iterations, only the nodes in Δ send messages. Neighbors of Δ receive the messages and calculate their own messages. If those nodes' messages differ by more than a threshold from the previous messages they sent, they are added to Δ. Although in worst case, all the nodes may be add to Δ and we would recalculate the graph, empirically only a small region of graph would be influenced and the BP would be converged quickly.

The discussion of the convergence analysis is detailed in [103].

3.5 Model-based Acceleration

Q-learning is the model-free RL learning framework. It utilizes no prior knowledge about the environment. Although it guarantees to converge with unchanged model and stable learning parameters, the sample complexity tends to limit its applicability to practical systems. As in standard RL, integrating the prior knowledge about the model has generally been more efficient [87][139]. In this section, we will accelerate the learning procedure by incorporating the prior knowledge about the link state with the Q-learning framework.

We could use the simulated experience in a learned 'model' to supplement real-world on-policy rollouts. The 'model', in the context of RL, is characterized by $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}\}$, where $\mathcal{P}$ is the set of transition probabilities $P(s^{t+1}|s^t, a^t)$ and $\mathcal{R}$ is the corresponding rewards. Given the initial state s^t and action a^t as the input, the model M could generate the sample state

s^{t+1} and corresponding reward $r(s^t, a^t, s^{t+1})$ according to the transition probability. With the known model M, the updating for Q-function could be modified by

$$\begin{aligned} & Q(s^t, a^t) = R(s^t, a^t) \\ + \quad & \gamma \sum_{s^{t+1}} P(s^{t+1}|s^t, a^t) \max_{a^{t+1}} Q(s^{t+1}, a^{t+1}) \end{aligned} \tag{3.38}$$

In Q-learning, the agent will take the action a^t under s^t, and obtain reward r^t and the next state s^{t+1} from the environment. It updates $Q(s^t, a^t)$ by minimizing the temporal difference at time slot t, as described in (3.15). In the model based learning framework, the agent could simulate the environment by generating all the possible future states s^{t+1}, r^t based on 'model' M, namely

$$\{s^{t+1}, r^t\} \leftarrow M(s^t, a^t) \tag{3.39}$$

Algorithm 5 Accelerating Q-Learning

Required: The current state $\boldsymbol{s^t}$, action $\boldsymbol{a^t}$ and the model M
Generate all possible s^{t+1} by $\boldsymbol{s}^{t+1} \leftarrow M(\boldsymbol{s^t}, \boldsymbol{a^t})$
Obtain $r(\boldsymbol{s}^t, \boldsymbol{a}^t)$ by M or historic measurements.
Update $Q(\boldsymbol{s}, \boldsymbol{t})$ by (3.38)
for $j = 1 : N$ **do**
    $bms \leftarrow$ random previously observed state
    $bma \leftarrow$ random action in $\boldsymbol{s}$
    $s', r \leftarrow M(\boldsymbol{s}, \boldsymbol{a})$
    Update $Q(\boldsymbol{s}, \boldsymbol{t})$ by (3.38)
end for

Then the current Q-function is updated in a Value Iteration (VI) way that considers the contribution of all the possible future states instead of the single sample state in Q-learning. This could speed up the convergence and guarantee to converge to the optimal [140]. The procedure of the accelerating framework is in Algorithm 5, which is based on *Dyna Q* [139].

For mmWave networks, the dynamics of the link state is stable and thus the transition probability between the state could be estimated. As mentioned in Section 3.2, the state transition probability could be obtained given the distribution of $\{l_{ij}, G_{ij}\}$.

In agent-based decomposition, for UE node, the transition probability could be calculated as

$$\begin{aligned} & P(s_i^{t+1}, s_{\Gamma(i)}^{t+1} | s_i^t, s_{\Gamma(i)}^t, a_i^t) \\ = \; & P(s_i^{t+1} | s_i^t, a_i^t) \prod_{m_j \in \Gamma(i)} P(s_j^{t+1} | s_j^t, a_j^t) \end{aligned} \tag{3.40}$$

Here, we assume that the state transition is only dependent on the local state/action pair.

Since the action of BS m_j is to control the transmission power and its state s_j is the transmission power itself, then we have

$$P(s_j^{t+1} | s_j^t, a_j^t) = \begin{cases} 1 & a_j^t = s_j^{t+1} \\ 0 & a_j^t \neq s_j^{t+1} \end{cases} \tag{3.41}$$

For the transition probability of the UE u_i, since the UE's state is the product of its alignment and link state, we have

$$P(s_i^{t+1} | s_i^t, a_i^t) = P(s_l^{t+1} | s_l^t, a_i^t) * P(s_G^{t+1} | s_G^t, a_i^t) \tag{3.42}$$

where $P(s_l^{t+1} | s_l^t, a_i^t)$ and $P(s_G^{t+1} | s_G^t, a_i^t)$ are characterized by the mmWave channel as the prior knowledge, as described in Section 3.2. a_i is the UE's choice of the connected BS. Since the link state l is dependent on the relative position, we have

$$P(s_l^{t+1} | s_l^t, a_i^t) = P(s_l^{t+1} | s_l^t) \tag{3.43}$$

which could be calculated from (3.1).

The beamforming alignment state is dependent on UE's choice of BS. If UE u_i decides to connect to BS m_j, where $x_{i,j} = 1$, then with a high probability, the link between them will be well aligned. Therefore, we will estimate the conditional transition matrix $\mathcal{T}|_{x_{i,j}=0}$

and $\mathcal{T}|_{x_{i,j}=1}$ separately, namely

$$\mathcal{T}|_{x_{i,j}=0} = \begin{bmatrix} p_{G_0|G_0,x_{i,j}=0} & p_{G_0|G_1,x_{i,j}=0} & p_{G_0|G_2,x_{i,j}=0} \\ p_{G_1|G_0,x_{i,j}=0} & p_{G_1|G_1,x_{i,j}=0} & p_{G_1|G_2,x_{i,j}=0} \\ p_{G_2|G_0,x_{i,j}=0} & p_{G_2|G_1,x_{i,j}=0} & p_{G_2|G_2,x_{i,j}=0} \end{bmatrix} \tag{3.44}$$

Here G is defined in Section 3.2.

To complete the implementation, we need to compute the expected reward under (s_j, a_j), namely

$$R(s_j^t, a_j^t) = \sum_{s_j^{t+1}} p(s_j^{t+1}|s_j^t, a_j^t) r(s_j^{t+1}, s_j^t, a_j^t) \tag{3.45}$$

We store the value of $r(s_j^{t+1}, s_j^t, a_j^t)$ when system interacts with real environment and transits into the state $\{s_j^t, a_j^t, s_j^{t+1}\}$.

3.6 Experiment and Simulation Result

In this section, we first check the performance for the message passing algorithm introduced in Section 3.4 for the coordination graph. Then we introduce our mmWave hardware platform, from which we obtain the parameters of the channel for the simulation. The numerical results are provided to evaluate the performance of the proposed RL framework.

3.6.1 Max-Sum Result

The results on the simple synthetic graph are summarized in Table 3.2. Q_{ES}, Q_{AMP}, Q_{EMP} are the values of the maximum Q-function calculated by an exhaustive search, agent-based decomposition with message passing and edge-based decomposition with message passing respectively.

We normalize the result by setting the maximum global Q-value as 1. For the synthetic graph, the message passing method can find the optimal Q-value using the exhaustive search method Q_{ES}. For the large graph, since it is impossible to find the exhaustive result, we only

Table 3.2: Max-Sum result

	Synthetic Graph	**Macro Graph**
Number of edge	3	306
Number of node	4	108
$\max Q_{ES}$	1	/
$\max Q_{AMP}$	1	1
$\max Q_{EMP}$	1	0.84

compare the result between the agent-base and edge-based result. As mentioned in Section 3.2, the UE's decision is dependent on the nearby BSs when there exist edges between them. When two UEs are dependent on two identical BSs, there a loop between two UEs in the graph. It is well known that the message passing method can find the exact solution in the tree structure while may not find the optimal solution in the loopy case [162]. However, the exhaustive search method requires the computational cost in the order of $O(A^N)$ while for the message passing method the computational cost is $O(AN)$. Here A is the number of actions for each node and N is the number of nodes. We improve the scalability at the cost of accuracy, whereas we will show in the next subsection that it can find the near-optimal result in the multi-agent reinforcement learning.

In Fig 3.5, the convergence result on a HetNet graph for proposed BP is shown. For vanilla BP (*Agent, Edge*), it takes more than 50 iterations to converge while for Efficient BP (*E-Agent, E-Edge*) proposed in Section 3.4.2, the inference could be converged in less than 20 iterations.

3.6.2 Simulation Parameters

The channel gain for blockage and beamforming is obtained from our own mmWave hardware platform.

Experiment Setup

We first introduce the setup for the experiment. The 56.5 GHz mmWave link is provided by Analog Devices EK1HMC6350 evaluation kit, which includes HMC6300 with the TX module

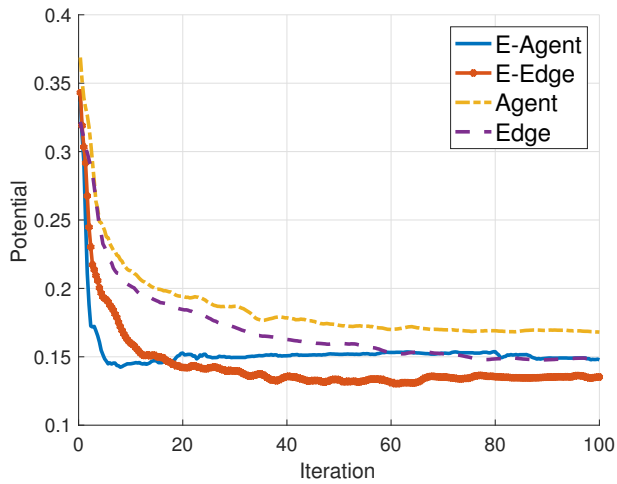


Figure 3.5: Convergeence for BP

and HMC6301 with the RX module. Tektronix TSG4102A, an RF vector signal generator, provides baseband signal. A Tektronix DPO70404C oscilloscope, with 4 GHz bandwidth and 25 GS/s sample rate, captures the waveform for offline analysis. Its four analog channels support differential signaling in quadrature modulation. The antenna is omnidirectional with a gain of 24 dBi. The antennas are rotated in the azimuth plane. The field experiment is shown in Fig 3.6.

Parameters

We first check the pathloss factor τ. The measurement is displayed in Fig. 3.7. By fitting the curve, we obtain $\tau_{LOS} = 1.4$ and $\tau_{NLOS} = 2.2$. Then the overall pathloss is given by

$$L_L(r) = 61.4 + 14\log_{10}(r_{i,j}) + \chi_L \tag{3.46}$$

and

$$L_N(r) = 72.0 + 22\log_{10}(r_{i,j}) + \chi_N \tag{3.47}$$

The gain for beamforming alignment is displayed in Fig 3.8. The misalignment could cause up to 20dB loss when the angle deviation is 40 degrees. Thus in our simulation, we set

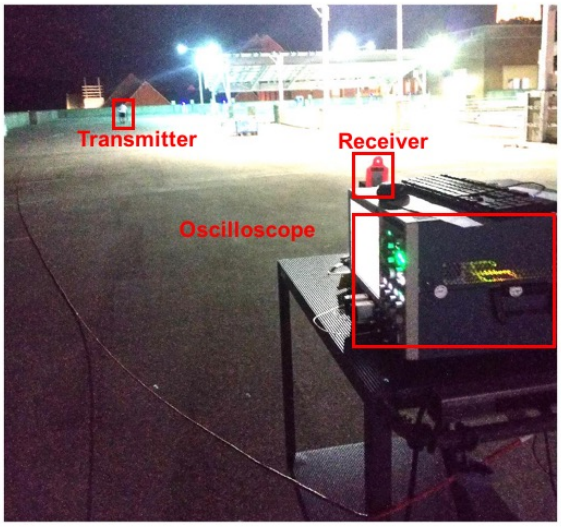


Figure 3.6: Image of outdoor measurement for LOS

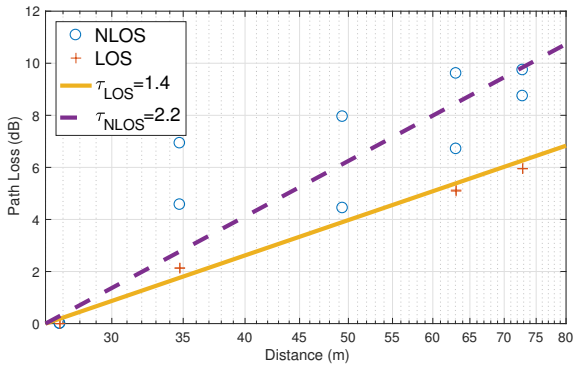


Figure 3.7: Measured path loss values relative to distance of 20m.

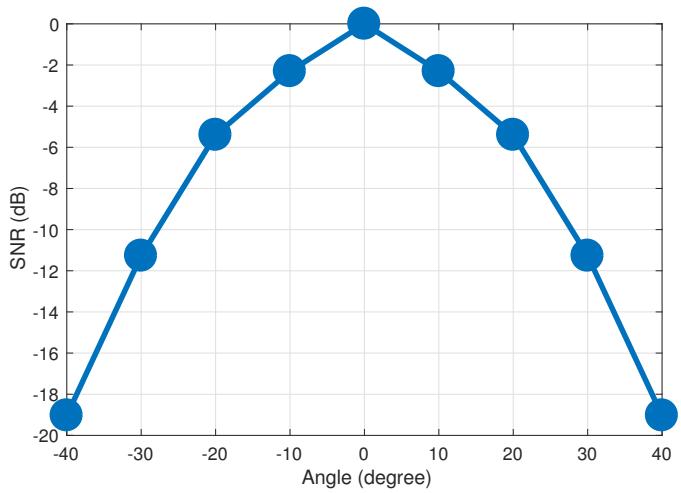


Figure 3.8: Angle Gain

the beamforming gain $G_{i,j} = -20dB$ when the UE and BS are not aligned and $G_{i,j} = -10dB$ when UE and BS are partially aligned.

The Markov transition matrix for link state l is

$$\mathcal{P}_l = \begin{bmatrix} 0.6 & 0.2 & 0.1 \\ 0.3 & 0.6 & 0.1 \\ 0.1 & 0.2 & 0.8 \end{bmatrix} \tag{3.48}$$

and the transition matrix for alignment state G is

$$\mathcal{P}_G = \begin{bmatrix} 0.7 & 0.25 & 0.05 \\ 0.2 & 0.75 & 0.05 \\ 0.05 & 0.45 & 0.5 \end{bmatrix} \tag{3.49}$$

Note that the transition probability is not small between two different states, assuming a high dynamics of the channel.

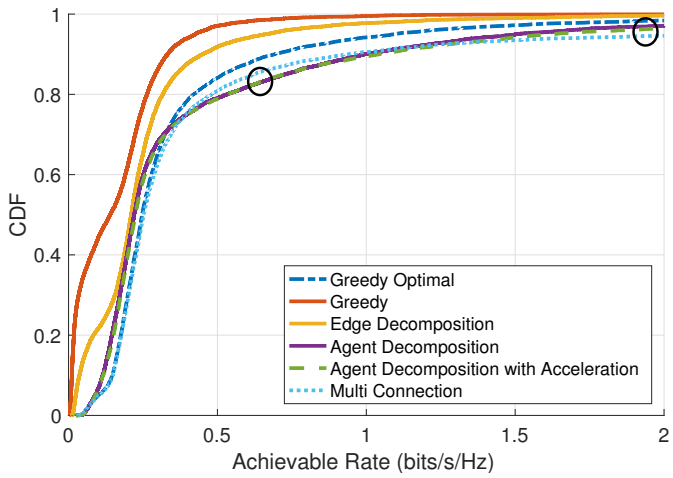


Figure 3.9: Transmission rate distributions for different settings

3.6.3 Simulation Results

For simulation, we randomly generate the UE according to the PPS. One simulation contains 1000 time slots and the environment is simulated according to the Markov transition matrix in Section 3.6.2. We repeat the simulation for 1000 times and get the averaged result.

The remainder of simulation parameters are given in Table 3.3.

Simulation results for the system level performance are presented in Fig 3.9 in terms of the UE rate distribution. We implement the mulit connectivity framework, where each UE is allowed to connect to multiple BSs in one time slot [166]. It could be considered as the upper bound for the proposed problem since connecting to multiple BSs provides more flexibility for UE and thus obtain more resource. Besides, we use some baseline setting for comparison. In the baseline setting, there is no bias for UE and each UE connects to the transmitter with the highest received power. We use the grid search to find the optimal bias factor for each BS. We call them greedy and optimal correspondingly. The transmission power in the baseline setting is unchanged. We implement the agent-based decomposition, agent-based decomposition with acceleration and edge decomposition to find the optimal resource allocation policy.

Fig 3.9 demonstrates that the proposed framework has close performance to the multi-connectivities method. Comparing to the multi-connectivities method, more UEs could obtain more than 0.5 bits/s/Hz in the proposed framework than multi-connectivities method

Table 3.3: Simulation Parameters

Variable	**Value**
Macro cell radius	100m
Carrier Frequency	56.5 GHz
Bandwidth	1 GHz
Number of BS	1
Number of pico transmitters	8
Number of UE	80
BS power	53 dBm
Pico transmitter power	23 dBm
Learning rate α	0.9
Discount factor β	0.9
PPS density λ_U	0.4
Reversed factor θ	1
Switch penality γ	1000

while the latter could help more UEs obtain rate larger than 1.5 bits/s/Hz. This indicates that the proposed framework could improve the fairness of the resource allocation while the multic-onnectivities is more greedy. This is because we add the penalty when UE switch from on BS to another such that when its rate is good, it may not choose to switch. It can also be observed that all the proposed learning frameworks obtain better performance than the purely greedy policy. Here, the performance is approximately 25 percent more significant for agent-based approach compared to the pure greedy policy. The agent-based approach is better than the edge-based approach, which results from the better Q-function approximation via message passing, as shown in Table 3.2. Comparing to the greedy optimal, for the low rate region where the achievable rate is less than 0.4 bits/s/Hz, the UEs obtain less transmission rates from the BS while for the high rate region, the average transmission rate is higher.

Fig 3.10 compares the average achievable rate (normalized by the bandwidth) by different resource allocation schemes with different transmission powers. We could observe that the proposed framework could achieve similar performance to the multi-connectivities method

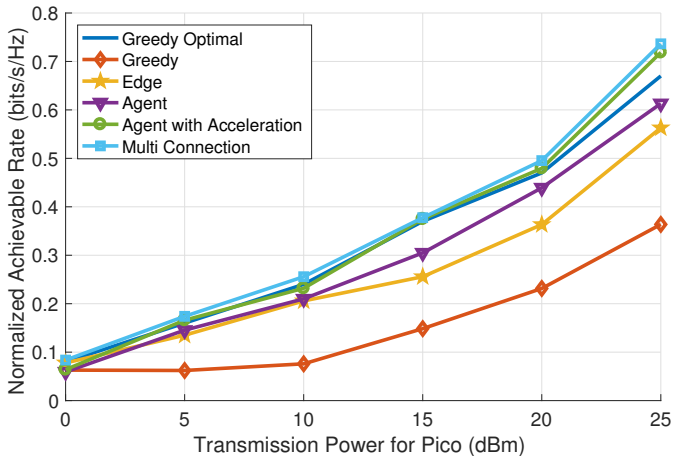


Figure 3.10: Transmission rate versus transmission power

while avoiding frequent switching for UEs. Besides, a remarkable performance gap can be seen in the figure between the proposed scheme and the baseline approach. Notably the gain goes larger as the transmit power of each mmWave BS increases, as could be explained by the fact that the proposed learning framework could manage the interference by flexible user association. We could obtain the similar conclusion by comparing the results of the greedy optimal and the proposed framework. When the transmission power is small, the greedy optimal and the agent-based approaches have similar performances since there is no need to coordinate the interference. With increased transmission power, since the BS in greedy optimal always work on the full transmission power, it becomes more difficult to manage the interference while in the learning framework, the transmission power could be adjusted according to the current state.

We simulate the dynamics of the mmWave networks for a fixed period of time. The states of the links between BS and UE change over time according to the transition probability we defined in Section 3.5. In Fig 3.11, we compare the switching frequency and the power consumption regarding the transmission power in the fixed time period. We normalized the value for a better demonstration. In the greedy optimal scheme, the BSs keep the maximum transmission power, while in the learning framework, the transmission power could change according to the environment. It proves that the proposed scheme could save the power while providing the comparable performance. Besides, it could be observed that in the greedy optimal scheme, the UE may change the connected BS frequently. This is because of

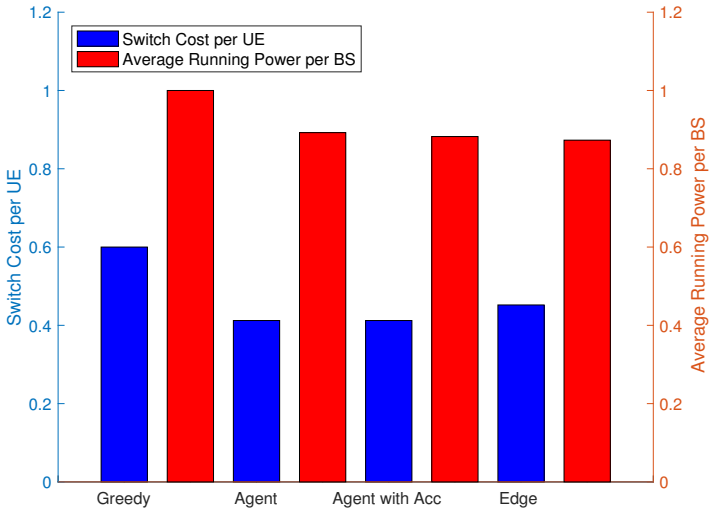


Figure 3.11: Switching cost and power cost

the dynamics of the mmWave network. Since we add a penalty term in the reward function, the agent will reluctantly change the associated BS. This will benefit the system in the power consumption since the switching is frequent for UE.

Finally, we check the performance of the proposed model-based acceleration in Fig 3.12. The computation time consists of two parts: Q-table update and message passing. It could be observed that most computation cost is on the message passing. This is because, in each iteration, the agent is required to update the local Q-function of small size. The length of total running time reflects the number of iterations required to converge. From our simulation, the agent-based decomposition with model acceleration converges 25-percent faster than the agent-based decomposition. The model based acceleration requires more computation to update Q-tables, since in each iteration it calculates the sum of the contributions from all the possible future states.

3.7 Conclusion

We propose a scalable and distributed RL framework for joint power control and user association in mmWave HetNet, where we consider the blockage and beamforming effects and model the link state as an MPD. We formulate the problem by maximizing the overall

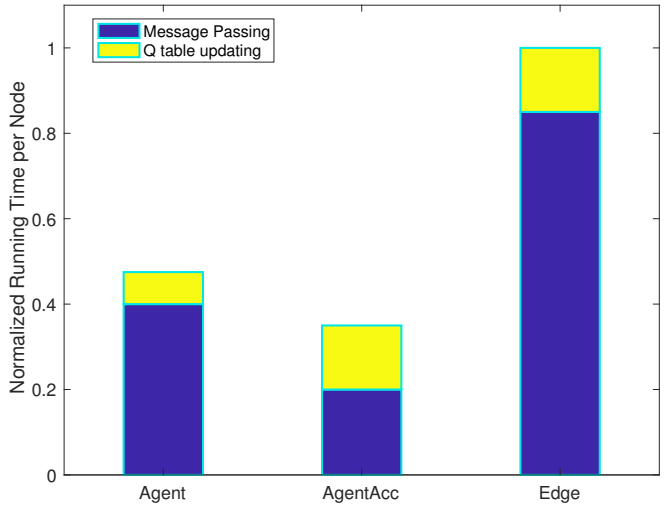


Figure 3.12: Running time for different RL frameworks

system throughput considering the switching cost for each UE. Using coordination graph, we decompose the problem into multiple local sub-problems based the topology of mmWave HetNet. UEs and BSs solve the local problems in a distributed manner while using the message passing algorithm to exchange the local information. We further accelerate the learning process by combining the prior knowledge about the dynamics of mmWave link state. We use statistics of our real world measurements to simulate the proposed framework. The performance of our proposed framework proves that it could increase the system throughput while reducing the overall transmission power comparing to the baseline approach.

Undoubtedly, the proposed framework still requires prior knowledge about the dynamics of the link state. We believe incorporating more features to the state space would further improve the performance. Besides, since we did not consider the fairness of the transmission, the proposed framework is more likely to allocate most the resource to good link. More comprehensive cost models can be taken into consideration, and practical implementation will be considered in the future.

Chapter 4

Distributed Coreset Boosting

4.1 Introduction

The rapid growth in the size and scale of modern data sets has fueled a lot of interest in solving machine learning tasks in a distributed manner. In the distributed setting, the bottleneck is often the communication capacity between computing machines [17]. Some recent researches have studied the communication efficiency in the distributed learning context from multiple aspects, including distributed optimization [44, 172], Probably Approximatedly Correct (PAC) learning [17, 34] and information theory [7, 168].

Meanwhile, in real-world distributed applications, the simplified assumption of independent and identically distribution for all samples breaks down, and labels can have structured, specific character on each distributed node [46]. For example, the model learnt from one mobile user could not be directly applied to another mobile user. When learning a global classifier on such different distributions with limited communication, the efficient convergence could not guarantee [174].

In this paper, we improve the communication efficiency and robustness to distribution in distributed learning by utilizing the redundancy of the data set, which is similar to source coding in communication theory. For a large scale data set, there may be only a small subset of data that is informative to the learning due to redundancy. We construct a coreset [2], namely a small and weighted subset of the data, to approximate the full dataset.

Coreset is widely studied for unsupervised learning, especially for clustering problems [15, 52]. Previous works [28] have provided elegant theoretical bound for coreset when the objective function is unbounded. Recent studies [70] have designed efficient coreset construction algorithms from the Bayesian point of view. It generates the coreset with the similar likelihood to the whole dataset for logistic regression classifiers. The coreset construction in this paper is the generalized framework that bridges the coreset with supervised learning. It shows that, by mining the structure of data using unsupervised learning, the efficiency of supervised learning can be significantly improved regarding the sample complexity.

We design a coreset construction algorithm that approximates the loss of the whole dataset for all concerned base functions $h \in \mathcal{H}$ with high probability. We will show that the proposed algorithm can 'compress' the data set by assigning sampling weight to different samples, similarly to traditional source coding in communication.

Then we will build the connection between the proposed coreset construction algorithm and the traditional Boosting algorithm. We prove coreset is a good choice to generate the 'weak' learner in boosting and its generalization performance bound could help analyze such additive learning model. A smooth coreset boosting algorithm is designed with computational efficiency and robustness to prevent overfitting. We show that the coreset boosting algorithm is easy to be adapted to distributed setting, effective in communication and robust to adversary distribution, which leads to its potential practical applications.

The remainder of this paper is organized as follows. We first formulate the problem and propose the coreset construction algorithm. Based on the coreset, the boosting algorithms in centralized and distributed setting are introduced. Numerical results are provided in the end.

4.2 Problem Setting

Let $\mathcal{D} = \left\{(X_n, Y_n)_{n=1}^{|\mathcal{D}|}\right\}$ be a dataset, where $X_n \in \mathbb{R}^d$ is the d-dimensional feature vector and $Y_n \in \{-1, 1\}$ is the corresponding label. Assume that the feature vector is rescaled to $[0, 1]^d$, which is widely used as a preprocessing. A given function class $\mathcal{F}$, in which each function

maps from $\mathbb{R}^d$ to $\mathbb{R}$, is said to be $\eta-$bounded if, for all $x \in \mathbb{R}^d$ and all $f \in \mathcal{F}$, $|f(x)| \leq \eta$. We denote by $h(x) : \mathbb{R}^d \to \mathbb{R}$ the corresponding base function whose sign predicts the label of sample x. The set of all possible base functions is denoted by $\mathcal{H}$. $l(x)$ is a loss function if it is non-negative and nonincreasing. The overall loss for $h(x)$ on $\mathcal{D}$ is defined as

$$L(h) = \sum_{i=1}^{|\mathcal{D}|} w_i l(Y_i h(X_i)) \tag{4.1}$$

where w_i is the normalized weight for sample (X_i, Y_i) such that $\sum_{i=1}^{|\mathcal{D}|} w_i = 1$. w_i can be considered as a discrete probability distribution over the $|\mathcal{D}|$ samples. In the initialization, we can set $w_i = \frac{1}{|\mathcal{D}|}$. In the subsequent processing, the weights could be updated.

Similarly, the empirical loss for a sample subset M is given by

$$\hat{L}_M(h) = \sum_{i=1}^{|M|} u_i l(Y_i h(X_i)) \tag{4.2}$$

u_i is the weight for (X_i, Y_i) in M, similarly to w_i for $\mathcal{D}$.

The whole dataset $\mathcal{D}$ is clustered into K clusters based on feature X using k-means clustering. Denote by G_k^n the set of samples in cluster k having the same label as that of (X_n, Y_n) excluding (X_n, Y_n) itself. Let G_k^{-n} be the set of samples that have different labels from that of (X_n, Y_n) in cluster k. $W_k^{w,n} = \sum_{(X_j,Y_j)\in G_k^n} w_j$, namely the sum of the sample's distribution who have the same label with (X_n, Y_n). Similarly, we have $W_k^{w,n-} = \sum_{(X_j,Y_j)\in G_k^{-n}} w_j$.

4.3 Generalized Coreset Construction

In this section, an efficient coreset construction algorithm will be proposed, such that with probability $1 - \delta$, we could sample the subset M such that

$$\left| L(h) - \hat{L}_M(h) \right| \leq \epsilon |L(h)|, \quad \forall h \in \mathcal{H}, \tag{4.3}$$

where $\epsilon \in (0, 1)$.

Algorithm 6 Coreset Construction

Input: data $(X, Y) \in \mathcal{D}$, distribution $\{w_i\}_{i=1}^{|\mathcal{D}|}$, K-clustering, tolerance ϵ, failure rate δ
for $n = 1 : |\mathcal{D}|$ **do**
  Calculate m_n using equation (4.5)
end for
for $n = 1 : |\mathcal{D}|$ **do**
  Calculate the sampling probability $p_n = \frac{m_n}{\sum_{n=1}^{|\mathcal{D}|} m_n}$
end for
Sample M from $\mathcal{D}$ using p_n
Output: Subset M

Essentially, the coreset construction algorithm updates the weights of samples in $\mathcal{D}$ and carries out the sampling in a single-round manner. The detailed algorithm is summarized in Algorithm 6. The detailed expressions for the algorithm and the performance are given in the following theorem.

Theorem 4.1. *Assume that the base function $h(x)$ is λ-Lipschitz and η-bounded. The dataset $\mathcal{D}$ is clustered into K clusters. The loss function $l(x)$ is nonincreasing, non-negative, convex and satisfies*

$$\frac{l(a)}{l(b)} \leq e^{|a-b|} \tag{4.4}$$

Given the distribution for each sample (X_n, Y_n) is w_n. Then, if the sampling weight for each point (X_n, Y_n) is $p_n = \frac{m_n}{\sum_{i=1}^{|\mathcal{D}|} m_i}$, where

$$m_n = \left[\frac{1}{w_n + \sum_{k=1}^{K} \left(W_k^{w,n} d_{X_n} + W_k^{w,n-} e^{-4\eta^2} \right)} \right] \tag{4.5}$$

with $d_{X_n} = e^{-\lambda \| \bar{X}_{G,k}^{n,w} - X_n \|^2}$ and

$$\bar{X}_{G,k}^{n,w} = \frac{\sum_{(X_i,Y_i) \in G_k^n} w_i X_i}{\sum_{(X_i,Y_i) \in G_k^n} w_i} \tag{4.6}$$

by sampling the subset M with size

$$|M| = c \frac{\sum_{n=1}^{|\mathcal{D}|} m_n}{|\mathcal{D}| \epsilon^2} \left[dim(\mathcal{H}) \log \left(\frac{\sum_{n=1}^{|\mathcal{D}|} m_n}{|\mathcal{D}|} \right) \right] + \log \frac{1}{\delta}, \tag{4.7}$$

with probability $1-\delta$, the sampled subset M satisfies (4.3).

Remark 1. *A typical choice for the base function $h(x)$ could be linear classifier or decision stumps. Strictly speaking, these functions do not satisfy the Lipschitz continuity assumption in Theorem 4.1 as they contain indicator function to generate the output label. However, the indicator function could be approximated by $tanh(x)$, which satisfies the Lipschitz continuity assumption. We further assume that the loss function $l(x)$ does not change too fast (not faster than the exponential function). This assumption holds for the widely used loss functions such as hinge, quadratic or linear loss.*

Proof We first define the sensitivity $\sigma_n(\mathcal{H})$ similarly in [51]

$$\sigma_n(\mathcal{H}) := \sup_{h\in\mathcal{H}} \frac{l(Y_n h(X_n))}{\sum_{l=1}^{|\mathcal{D}|} w_l l(Y_l h(X_l))} \tag{4.8}$$

[28] provides Theorem 4.2 to construct coreset

Theorem 4.2. *Fix $\beta > 0$. Θ is the parameter space. $\mathcal{F}$ is a set of function and $\forall f \in \mathcal{F}, f(\theta) > 0$. For $n \in [N]$, let $m_n \in \mathbb{R}_+$ be chosen such that*

$$m_n \geq \sigma_n(\Theta)$$

There is a universal constant c such that if M is sample from $\mathcal{F}$ of size

$$|M| \geq \frac{c\sum_{n=1}^{N} m_n}{N\beta^2}\left[dim(\mathcal{F})log(\frac{\sum_{n=1}^{N} m_n}{N})\right] + ln(1/\delta) \tag{4.9}$$

such that the probability each element of M is selected independently from $\mathcal{F}$ with probability $\frac{m_n}{\sum_{n=1}^{N} m_n}$ that $f_n \in \mathcal{F}$ is chosen, then with probability at least $1-\delta$, for all $\theta \in \Theta$,

$$\left|\bar{f}(\theta) - \frac{\bar{m}_N}{|M|}\sum_{f_n\in M}\frac{f_n(\theta)}{m_n}\right| \leq \beta\bar{f}(\theta)$$

In this paper, $f_n(\theta) = l(Y_n h(X_n))$ and the selection of f is equal to sampling (X_n, Y_n) from $\mathcal{D}$. m_n could be chosen arbitrarily large to satisfy the condition in Theorem 4.2 but this will result in a large coreset size according to (4.9). To complete the proof, we need to

find the tight upper bound of $\sigma_n(\theta)$ with Lemma 4.11. The detailed proof of Lemma 4.11 is in the appendix.

Lemma 4.3. *For any k-clustering Q,*

$$\sigma_n(\mathcal{H}) \leq \left[\frac{1}{w_n + \sum_{m=1}^{k}(W_k^{w,n} d_{X_n} + W_k^{w,-n} e^{-4\eta^2})}\right]$$

where $d_{X_n} = e^{-\lambda||\bar{X}_{G,k}^{n,w} - X_n||^2}$.

Combining Lemma 4.11 and Theorem 4.2, with the fact $h(x)$ is η-bounded such that $e^{-|h(E[X])+h(X_n)|^2} \geq e^{-4\eta^2}$, we complete the proof.

The sampling weight p_n could be considered as the 'votes' from all the cluster centers. It is different from the discrete distribution w_n, which is determined by prior knowledge or learning framework. Two main conclusions could be obtained from the expression of m_n in (4.5):

- m_n is large if (X_n, Y_n) is far away from all the clusters.
- m_n is large if (X_n, Y_n) is near the cluster center while its label is inconsistent with most samples in this cluster.

Therefore, those samples that are consistent with the neighbors have smaller sampling weights, while those different from the neighbors have larger weights. The isolated samples are more likely to be sampled into the coreset. This is similar to data compression algorithms where fewer bits are needed to represent the more frequent messages while more bits are required for the rare ones. Regarding the cluster size, if the data is closely clustered, the distance $|X_n - \hat{X}|$ between the sample and its parent cluster center is small, the corresponding $e^{-|X_n - \hat{X}|}$ will dominate the overall sum in the denominator of m_n and the weight for sample will be greatly determined by the number of samples and their labels in the same cluster. When the data is not closely clustered, the weight for each sample will be influenced by the centers of multiple clusters.

In the proof of Lemma 4.11, we assume the average of all samples in each cluster is identical to the cluster center. If the clustering result does not satisfy such condition, we

need to add a small positive term in (4.5) to secure the upper bound, which results in larger coreset.

The proposed coreset construction algorithm is computationally efficient. The clustering can be obtained efficiently via the k-means++ algorithm in $O(|\mathcal{D}|K)$ time [11]. The computation complexity for m_n is also in $O(|\mathcal{D}|K)$. This is desirable for the design of scalable learning system. Only $O(K)$ extra memory is needed to store the number of positive and negative samples, respectively, in each cluster.

We also obtain the following corollary, where the concept of ϵ-approximation can be found in [70].

Corollary 4.4. *Define the logistic likelihood by*

$$P(Y_n|X_n, h) = \frac{1}{1 + \exp(-Y_n h(X_n))}, \tag{4.10}$$

and the log-logistic-likelyhood for $\mathcal{D}$ as $\mathcal{L}(h, D) = \sum_{n=1}^{|\mathcal{D}|} \log P(Y_n|X_n, h)$. By Theorem 4.1, the constructed coreset $\mathcal{M}$ could approximate the log-logistic-likelyhood of $\mathcal{D}$ when loss function in Theorem 4.1 is $l(Y_n h(X_n)) = \log P(Y_n|X_n, h)$.

Therefore, from the Bayesian perspective, the coreset could be considered as a useful approximation for the original samples. Given the base function h generated by the coreset with small size, its performance has a certain assurance on the original dataset $\mathcal{D}$. This is especially desirable in the setting of distributed learning, where communication is the bottleneck, the exchanged messages are limited and we prefer to extract the information from subset using sample as little as possible. In the next section, we will show that the coreset construction algorithm is a natural choice to generate weak learner [122] for boosting, and that by utilizing the property of coreset, high convergence rate and sample efficiency in learning could be achieved.

4.4 Coreset Boosting

In this section, the proposed coreset construction algorithm will be integrated into the Boosting algorithm. The learning process for boosting could be considered as the coordinate

descent in the function space $\mathcal{H}$ [122]. In iteration t, the booster generates a weak learner $h^t(x)$ such that

$$h^t = \arg_{h\in\mathcal{H}} \min L\big(H^{t-1}(X) + \gamma_t h(X)\big) \tag{4.11}$$

In the centralized setting, the base function $h(x)$ is generated based on the whole dataset while in distributed setting, to save the communication cost, we prefer to learn $h^t(x)$ with the subset M in each iteration [17]. Thus we are only able to find $\hat{h}^t(x)$ based on evaluating $\hat{L}_M(h)$. The Hoeffding inequality [68] enables us to bound $L(\hat{h}^t(x))$ with $\hat{L}_M(h)$ for random sampling if loss function $l(x)$ is bounded. The bounded $L(\hat{h}^t(x))$ could guarantee that $\hat{h}^t$ leads to a lower value of loss in the coordinate direction. But this is not the case for most boosting algorithms. For example, in AdaBoost $l(x)$ is the exponential function. When $L(\hat{h}^t(x))$ is unbounded, it's hard to determine if $\hat{h}^t$ could decrease the loss in (4.11). By contrast, the coreset could approximate $L(\hat{h}^t(x))$ with $\hat{L}_M(\hat{h}^t(x))$ for unbounded loss function. In this section, we show the bounded $L(\hat{h}^t(x))$ could lead to the decrease of objective function and accelerate the convergence.

Corollary 4.5. *The coreset constructing algorithm generates subset M such that with probability $1-\delta$,*

$$|L_{Ada}(h) - \hat{L}^M_{Ada}(h)| \leq \beta |L_{Ada}(h)|, \tag{4.12}$$

where $L_{Ada}(h) = \sum_{n=1}^{|\mathcal{D}|} e^{-h(X_n)Y_n}$.

In AdaBoost, the loss function $l_{Ada}(x) = e^{-x}$ satisfies the assumptions in Theorem 4.1 and thus the coreset for AdaBoost could be constructed by replacing the $l(x)$ with $l_{Ada}(x)$ in Theorem 4.1.

Unfortunately, AdaBoost is vulnerable to the outliers and overfitting [41]. The most commonly given explanation is that, in each iteration, AdaBoost assigns too much weight on the outliers. To fix the problem, a smooth loss function $l_{sm}(x)$ is used in this paper that is similar with MadaBoost [43]

$$l_{sm}(x) = \begin{cases} e^{-x}, & x \geq 0 \\ 1-x, & x < 0 \end{cases} \tag{4.13}$$

Algorithm 7 Coreset Boosting

1: **Input:** Dataset (X, Y) divided into K clusters with size N, $H^0(X) = 0$
2: **for** $t = 1 : T$ **do**
3: Construct the coreset (X_s, Y_s) with size $|M|$
4: Solve $h^t = \arg\min_{h \in H} \hat{L}_M(h)$
5: Update

$$w_n^t = -l'(H^{t-1}(X_n)Y_n)$$

and $W^t = \sum w_n^t$. Calculate $\gamma_t = \frac{W^t E^t[h^t(X)Y]}{NK^2}$
6: Update $H^t(X) = H^{t-1}(X) + \gamma_t h^t(X)$
7: **end for**
8: **Output:** $H^T(X) = \sum_{t=1}^{T} \gamma_t h^t(X_n)$

Note that $l_{sm}(x)$ decreases linearly when $x < 0$. Optimizing $l_{sm}(x)$ over $|\mathcal{D}|$ is equal to maximizing the soft margin [149].

By applying the similar proof in Corollary 4.5, it is straightforward to construct the coreset for $l_{sm}(x)$ such that with probability $1 - \delta$

$$|L_{sm}(h) - \hat{L}_{sm}^M(h)| \leq \beta |L_{sm}(h)|, \quad \forall h \in \mathcal{H} \tag{4.14}$$

We first propose the centralized version of coreset boosting in Algorithm 7 and show it could converge efficiently.

Theorem 4.6. *Suppose the feature X is scaled to [0,1]. Assume $h(X)$ is η-bounded and the empirical loss for $h^t(x)$ satisfies $\hat{L}_{sm}^M(h^t) \leq (1+\beta)(1-\alpha)$, then with probability $1-\delta$, the output of Algorithm 7 could achieve error rate $\min_{h \in H} Err(h) + \epsilon$ and converges in $O(\frac{1}{\epsilon^{2-2c}})$ iterations.*

Note that in boosting, the distribution w_n for each sample is updated in each iteration and the sampling probability for generating the coreset m_n is calculated based on w_n.

Proof. The 0-1 loss $Err_{\mathcal{D}(h)}$ is upper bounded by $L_{sm}(h)$ since $l_{sm}(yh(x)) \geq \mathbf{1}_{h(x) \neq y}$. Instead of handling the 0-1 loss directly, we will prove in each iteration, $L_M(h)$ decreases by larger than $O(\epsilon^{2-2c})$ with high probability. First apply Taylor expansion on $l(x)$ with $l''(x) \leq 1$

$$l(x) - l(x + \Delta x) \geq -\Delta x l'(x) - \frac{\Delta x^2}{2}$$

Let x be $\sum_{t=1}^{T-1} \gamma_t Y_n h^t(X_n)$ and Δx be $\gamma_T Y_n h^T(X_n)$,

$$\begin{aligned} & l(H^{T-1}(X_n)Y_n) - l(H^{T-1}(X_n)Y_n + \gamma_T Y_n h^T(X_n)) \\ \geq \; & \gamma_T Y_n h^T(X_n)[-l'(H^{T-1}(X_n)Y_n)] - \frac{\gamma_T^2 (Y_n h^T(X_n))^2}{2} \\ \geq \; & \gamma_T Y_n h^T(X_n) w_n^T - \frac{\gamma_T^2 \eta^2}{2} \end{aligned}$$

This is the direct result of the definition for w_n^T in Algorithm 2 and the fact that $Y_n h^T(X_n) \leq \eta$. Take the expectation of both sides and assume the initial distribution for each sample is $\frac{1}{|\mathcal{D}|}$. Then we have

$$\begin{aligned} & L_{sm}(H^{T-1}) - L_{sm}(H^T) \\ = \; & E_0\big[l(H^{T-1}(X_n)Y_n)\big] - E_0\big[l(H^T(X_n)Y_n)\big] \\ \geq \; & \frac{\gamma_T W^T E^T[Y h^T(X)]}{|\mathcal{D}|} - \frac{\gamma_T^2 \eta^2}{2} \end{aligned}$$

By choosing $\gamma_T = \frac{W^T E^T[Y h^T(X)]}{N\eta^2}$, the maximum value for the right side of the equation could be achieved.

To complete the proof, we need to verify that given the assumption h^t is generated on the coreset, $2\frac{(W^T E^T[Y h^t(X)])^2}{|\mathcal{D}|^2 \eta^2}$ is in the order of $O(\epsilon^{2-2c})$.

Lemma 4.7. *Assume in each iteration t we could always find a base function h^t based on the coreset such that the corresponding smooth loss $\hat{L}_{sm}^M(h^t) \leq (1+\beta)(1-\alpha)$. Then with probability $1-\delta$,*

$$W^t E^t[h^t(X)Y] \geq |\mathcal{D}|\alpha(\min_{h \in H} Err(h) + \epsilon)^{1-c} \tag{4.15}$$

Proof. Combining the property of coreset in Theorem 1 and the assumption, we have

$$L_{sm}(h^t) \leq 1 - \alpha \tag{4.16}$$

with probability $1-\delta$.

For simplicity, denote $Z_n^h = h^t(X_n)Y_n$. p_n is the distribution. Then we have the classified/misclassified set of points as $Z^+ = \{(X_n, Y_n) | Z_n^h > 0\}$ and $Z^- = \{(X_n, Y_n) | Z_n^h <$

0}. We first consider $E^t[Z^h]$.

$$E^t[Z^h] = \sum_{(X_n,Y_n)\in Z^-} p_n Z_n^h + \sum_{(X_n,Y_n)\in Z^+} p_n Z_n^h$$

Since

$$\begin{aligned} L_{sm}(h^t) &= \sum_{(X_n,Y_n)\in Z^-} p_n l(Z_n^h) + \sum_{(X_n,Y_n)\in Z^+} p_n l(Z_n^h) \\ &= \sum_{(X_n,Y_n)\in Z^-} p_n(1 - Z_n^h) + \sum_{(X_n,Y_n)\in Z^+} p_n e^{-Z_n^h} \\ &= -E^t[h^t(X)Y] + \sum_{(X_n,Y_n)\in Z^-} p_n \\ &+ \sum_{(X_n,Y_n)\in Z^+} p_n(e^{-Z_n^{h^t}} + Z_n^{h^t}) \end{aligned}$$

Applying the inequality $x + e^{-x} > 1$ and (4.16),

$$-E^t[h^t(X)Y] + \sum_{(X_n,Y_n)\in Z^-} p_n + \sum_{(X_n,Y_n)\in Z^+} p_n \leq 1 - \alpha$$

As p_n is the distribution for each sample such that

$$\sum_{(X_n,Y_n)\in Z^-} p_n + \sum_{(X_n,Y_n)\in Z^+} p_n = 1$$

we have $E^t[h^t(X)Y] \geq \alpha$.

Consider $W^t = \sum_{Z_n\in Z^-} w_n + \sum_{Z_n\in Z^+} w_n$. In iteration t, H^{t-1} is not good enough which implies $Err(H^{t-1}) > \min_{h\in H} Err(h) + \epsilon$. According to the weighting function in Algorithm 1, for $(X_n, Y_n) \in Z^-$, $w_n = 1$, $(X_n, Y_n) \in Z^+$, $w_n \geq 0$, total weight is upper bounded by

$$\begin{aligned} W^t &\geq |\mathcal{D}|(\min_{h\in H} Err(h) + \epsilon) + \sum_{(X_n,Y_n)\in Z^+} w_n \\ &\geq |\mathcal{D}|(\min_{h\in H} Err(h) + \epsilon)^{1-c} \end{aligned}$$

where c is the positive constant factor. The second term is lower bounded as we assume the error rate for H^{t-1} is less than 0.5 and therefore implies

$$1 - (\min_{h \in H} Err(h) + \epsilon) \geq (\min_{h \in H} Err(h) + \epsilon)$$

As we initialize $H^0(X) = 0$, $L_{sm}(H^0) = 1$. Given in each iteration the loss function $L_{sm}(H)$ decreases at least by $\frac{(\min_{h \in H} Err(h)+\epsilon)^{2-2c}\alpha^2}{\eta^2}$, we could conclude the algorithm converges in $O(\frac{1}{\epsilon^{2-2c}\alpha^2})$ iterations. This convergence rate is better than the previous boosting algorithm's $O(\frac{1}{\epsilon^2\gamma^2})$ [76], when they have the access to γ weak learner. Observe that the convergence rate depends on the correlation of base function $E[h^t(X)Y]$. This indicates the larger correlation $h^t(x)$ has, more useful information it brings to the booster and sequently the algorithm will converge faster.

A major concern for the proposed boosting algorithm is computational efficiency. In boosting, it takes $O(|\mathcal{D}|)$ to update the weight, $O(|\mathcal{D}|)$ for constructing the coreset and extra $O(|M|^a)$ for generating the weak base function h^t in each iteration. $|M|$ is small comparing to $|\mathcal{D}|$. The base function $h(x)$ is not necessary to be accurate, which implies $a \leq 2$ [10]. Therefore, the overall computation in each iteration is linear to the size of dataset and the overall computation cost is $O(\frac{|\mathcal{D}|}{\epsilon^{2-c}\alpha^2})$. As mentioned in previous section, the computation cost for clustering is $O(|\mathcal{D}|)$. Although both of them are linear to the data size, empirically, the boosting requires much more computation than clustering. In the next section, we will demonstrate that the benefit of the clustering, which makes the learning framework communication efficient and robust to distribution in distributed setting.

4.5 Distributed Coreset Boosting

Learning in distributed excels at processing large scale data while the communication cost for the shared information may limit the overall performance. Our coreset boosting algorithm could be adapted to distributed setting with small communication cost. Assume there are r clients over which the data is randomly partitioned, with D_i the set of index of data points on client i and $n_i = |D_i|$. From the observation in [50], we have

Algorithm 8 Distributed Coreset Boosting

Input: data (X, Y), r worker nodes each with dataset size n_i and master node.
Distributedly cluster the data set.
for $t = 1 : T$ **do**
  Worker nodes locally construct and send the coreset M_i to master node.
  Master node finds h^t using the received coreset and broadcasts.
  Worker node i locally updates $w_n^{t,i}$ and sends to the master node.
  Master node calculates and broadcasts γ_t and W^t.
  Update $H^t(X) = H^{t-1}(X) + \gamma_t h^t(X)$.
end for
Output: $H^T(X)$.

- If M_i is the coreset for $\mathcal{D}_i$, then $\cup M_i$ is the coreset for $\cup \mathcal{D}_i$.

Therefore, the master node could construct the global coreset by collecting and merging the local coresets generated by distributed nodes. Since there is no assumption that the sample in each node is $i.i.d.$ in coreset framework, which is always the assumption in other sampling methods, the proposed coreset constructing algorithm is robust to adversary distribution where D_i could be extreme different to each other. The cost for robustness is $r - 1$ extra small coresets. We will prove in Theorem 4.8 that the extra $r - 1$ coresets are small. The distributed coreset boosting is described in Algorithm 9

The overall communication in Algorithm 9 contains two parts.

- Clustering. k-means clustering is needed as preprocessing. The typical communication cost for efficient distributed k-means clustering is $O(rm)$, where r is the number of distributed nodes and m is the number of connections between the nodes[18].

- Learning. Specifically, in each iteration, the coreset has to be transmitted through the communication channel from the worker nodes. After that, the master node broadcasts the classifier $h^t(x)$ back to the worker. There is extra communication for transmitting γ_t and W_i^T, which is the ignorable overhead.

The size of coreset in each iteration is $\sum_i^k |M_i|$ in (4.7), where $|\mathcal{D}|$ is replaced by $|\mathcal{D}_i|$. Notice that m_n in Theorem 4.1 depends on the data size $|\mathcal{D}_i|$. We will prove that when the $|\mathcal{D}_i|$ is large, m_n is upper bounded by the factor that is independent of $|\mathcal{D}_i|$.

Theorem 4.8. *Suppose the base function $h(x)$ is λ-Lipschitz and η-bounded. The distribution for each sample is $\frac{1}{|\mathcal{D}|}$. If $|\mathcal{D}|$ is large enough such that $|\mathcal{D}| > e^{\max(2\lambda, 4\eta^2)}$, then m_n is upper bounded by $O(e^{\max(2\lambda, 4\eta^2)})$.*

Proof From Theorem 1, we have

$$m_n = \left[\frac{1}{\frac{1}{|\mathcal{D}|} + \sum_{k=1}^{K}(W_k^{w,n} d_{X_n} + W_k^{w,n-} e^{-4\eta^2})}\right]$$

where $d_{X_n} = e^{-\lambda||\bar{X}_{G,k}^{w,n} - X_n||^2}$.

For normalized vector X_n, we have $||\bar{X}_{G,i} - X_n|| \leq 2$. As

$$\sum_{k=1}^{K}(W_k^{w,n} + W_k^{w,n-}|) = 1 - \frac{1}{|\mathcal{D}|}$$

m_n is further upper bounded by

$$m_n \leq \frac{1}{\frac{1}{|\mathcal{D}|} + e^{-\max(4\lambda, 4\eta^2)}}$$

Given the assumption $|\mathcal{D}| > e^{\max(4\lambda, 4\eta^2)}$, we have the desired upper bound for m_n as $O(e^{\max(4\lambda, 4\eta^2)})$.

The upper bound for the coreset size depends on $h(x)$'s lipschitz constant λ and maximum value η (If the data is not the normalized vector, then it is also related to the dimension d for X). Generally speaking, if $h(x)$ has broader range and sharper derivative, which capture the complexity of $h(x)$, then larger sample size $|M|$ is required. If the base function is too complicated that exceeds the descriptive capacity of $\mathcal{D}$, the coreset will approach $\mathcal{D}$ itself.

Insert the upper bound of m_n into (4.7), the upper bound of the coreset size is in the order of $\hat{O}(\frac{e^{\max(4\lambda, 4\eta^2)}}{\beta^2} dim(\mathcal{H})r)$. Here $dim(\mathcal{H})$ could be considered as the VC dimension of the weak classifier $h(x)$. The communication cost for transmitting $h^t(x)$ is $O(dim(\mathcal{H}))$. Since we prove in Theorem 4.6 that the algorithm converges in $O(\frac{1}{\epsilon^{2-2c}})$ iterations, the total communication for the distributed coreset boosting is $\hat{O}(\frac{e^{\max(4\lambda, 4\eta^2)}}{\beta^2 \epsilon^{2-2c}} dim(\mathcal{H}))$.

The proposed algorithm contains sampling in each iteration. It is possible that some samples are selected for multiple times throughout the learning. We set cache in both

Table 4.1: Classification accuracy on various data sets using subset.

Data set	WebSpam	CovType	Yahoo!
SmoothBoost Coreset Sampling			
Acc_{tr} %	**91.54 (0.2)**	**75.45 (0.4)**	**62.90 (0.3)**
Acc_{te} %	**90.19 (0.3)**	**75.15 (0.3)**	**62.35 (0.2)**
Time	104.1s	200.9s	1200.3 s
Clustering	14.1s	30.2s	198.3 s
Boosting	90 s	170.7s	1002.0 s
SmoothBoost Random Sampling			
$Accu_{tr}$ %	89.49 (0.2)	73.06 (0.3)	60.11 (0.4)
$Accu_{te}$ %	88.75 (0.2)	72.90 (0.5)	60.01 (0.2)
Time	**82.1s**	84.1s	903.5s
AgnosticBoost with Subset			
$Accu_{tr}$ %	90.16 (0.2)	74.32 (0.2)	61.14 (0.5)
$Accu_{te}$ %	90.00 (0.1)	73.09 (0.4)	61.01 (0.4)
Time	93.1s	210.1s	1223.5s
AdaBoost with Random Sampling			
$Accu_{tr}$ %	89.38 (0.1)	73.32 (0.3)	59.14 (0.5)
$Accu_{te}$ %	88.97 (0.1)	71.09 (0.4)	58.11 (0.4)
Time	84.1s	**75.3**s	**870.2**s

worker nodes and master node. We stored the index of transmitted sample in work node. If such samples are repeated sampled to construct the coreset, work node will inform the master and they are no longer transmitted. Since in boosting most distribution is assigned to the outliers throughout learning, those points would be repeatedly chosen in high frequency. We expect the cache mechanism could reduce the communication cost efficiently.

Synchronization is critical in the distributed learning. The proposed framework could mitigate the straggler problem since the worker nodes are only responsible for updating the weight and sampling. Learning the base function is the computational expensive part, which could be well handled by the proposed framework as the sample size $|\mathcal{M}|$ for learning base function is small. We further manange the straggler problem by implementing the proposed algorithm with MapReduce framework.

Table 4.2: Classification accuracy on various data sets using all the training set.

Data set	WebSpam	CovType	Yahoo!
Dimension	127	54	10
Size	350000	581012	5811883
SmoothBoost with whole data set			
$Accu_{tr}$ %	92.46 (0.2)	75.46 (0.1)	63.00 (0.2)
$Accu_{te}$ %	89.99 (0.1)	75.25 (0.2)	62.58 (0.2)
AdaBoost with whole data set			
$Accu_{tr}$ %	92.14 (0.2)	73.31 (0.1)	62.90 (0.2)
$Accu_{te}$ %	89.71 (0.1)	72.91 (0.1)	62.80 (0.1)
AgnosticBoost with whole data set			
$Accu_{tr}$ %	91.00 (0.1)	72.31 (0.2)	61.90 (0.3)
$Accu_{te}$ %	89.78 (0.2)	71.91 (0.2)	61.80 (0.2)

4.6 Result

In this section, we evaluated the empirical performance of the proposed coreset boosting algorithm on 2 middle size datasets of varying type, Web, CovType[1] and one large dataset Yahoo! [35], as summarized in Table 1. All the features are rescaled to $[0, 1]$. The lipschitz constant λ is set to 10 as we approximate the step function with $tanh(x)$ and $\eta = 2$. The hyperparameters for sampling, β and δ, are set as 0.08 and 0.05 respectively. The cluster number k is 16 and the dataset is distributed to 16 workers.

4.6.1 Approximation Quality

Since the ultimate goal is to use the coresets to approximate the true loss of the dataset, we first check the performance on the approximation factor

$$\epsilon = \frac{|L(h) - \hat{L}(h)|}{|L(h)|}$$

The approximation factor measure the relative difference between the loss on the subset and the loss on the whole dataset. We randomly generate 5000 η-bouned linear base functions

[1]The Web dataset could be required from https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html and the CovType could be required from https://archive.ics.uci.edu/ml/datasets/Covertype

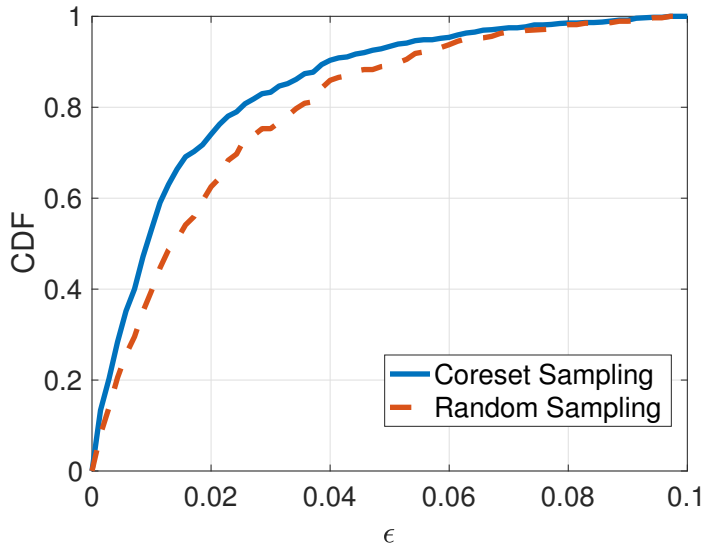


Figure 4.1: Cumulative distribution for ϵ

$h(x)$ and evaluate the approximation factor ϵ. The experiment is repeated for 100 times. Fig 4.1 shows our result on *Web* data. The size for the coreset and the random subset is 250. The distribution of approximation factor ϵ for coreset is centered largely around the origin while the distribution for random sampling has the relatively fat tail. We have $Var(\epsilon_{core}) = 0.0979$ and $Var(\epsilon_{random}) = 0.1308$. The result suggests we are able to construct coreset whose loss is close to the whole dataset for most possible clasiifiers we concerned. Besides, the coreset outperformas the random sampling regrading the approximation quality

4.6.2 Learning Quality

We use decision stumps as our weak learners. The simulations are repeated for 20 times and we showed the standard deviation of the accuracy for the randomness of sampling. 70 percent of the data is assigned for training. We compare the performance of proposed SmoothBoost to the classic AdaBoost and the AgnBoost introduced in [34]. We first check the centralized version while we have the access to all the training data. The result is shown in Table 4.2. These three algorithms achieve similar result when trained on all the training data. Then we compare the performance in the distributed setting, where in each iteration, we only sample a subset of the training data with the same size such that the communication cost for each algorithm is the same. The results are shown in Table 4.1. The bold entries

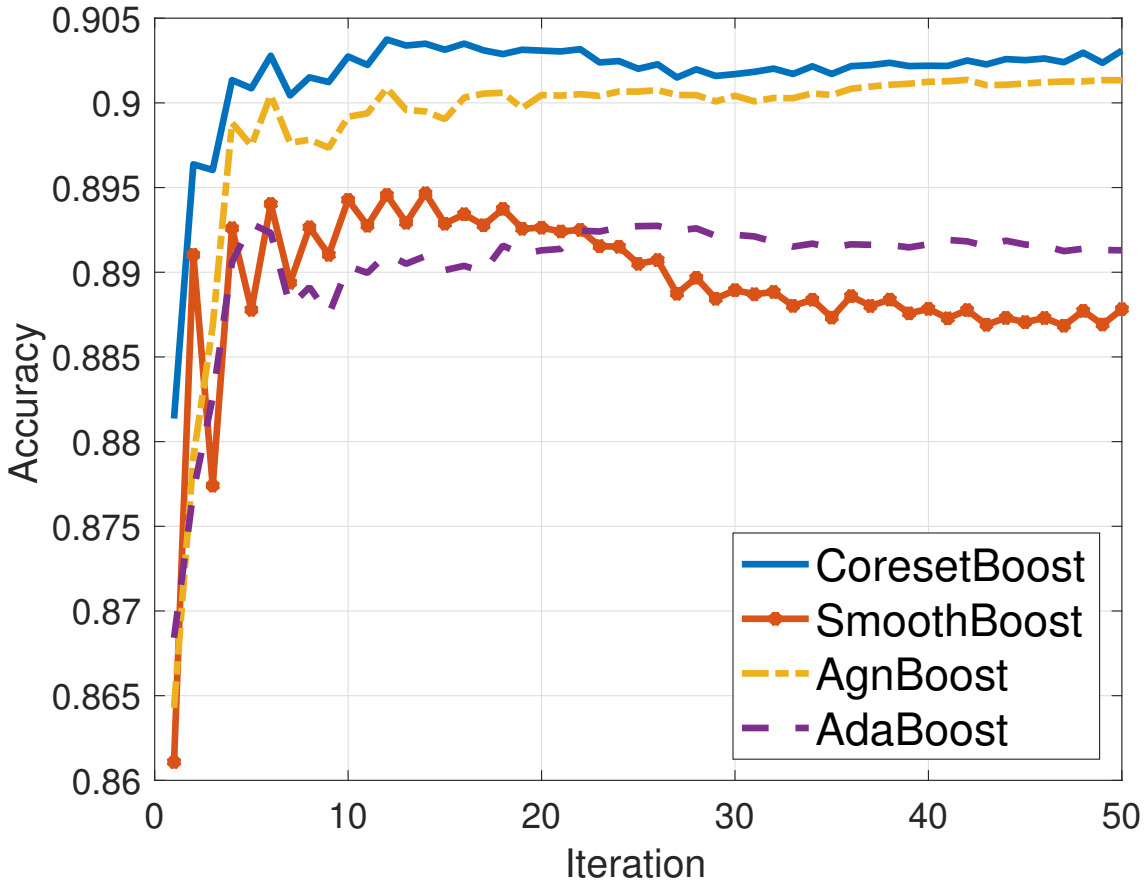


Figure 4.2: WebSpam training process

indicates the best error rate. One can see that the proposed CoresetBoosting outperforms the SmoothBooster who has the access to random sampling subset. Meanwhile, it has the competitive performance with the SmoothBooster who has the access to the whole dataset. Besides, the proposed algorithm is scalable as the running time is near linear to the overall dataset size. Fig 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 demonstrate the learning process during the boosting. The coreset boosting has better generalized performance than random sampling version and it is more efficient at reducing the loss $L_{sm}(H^T)$. As we mentioned, the clustering will not introduce too much extra compuation (about 15 percent) while the performance is improved.

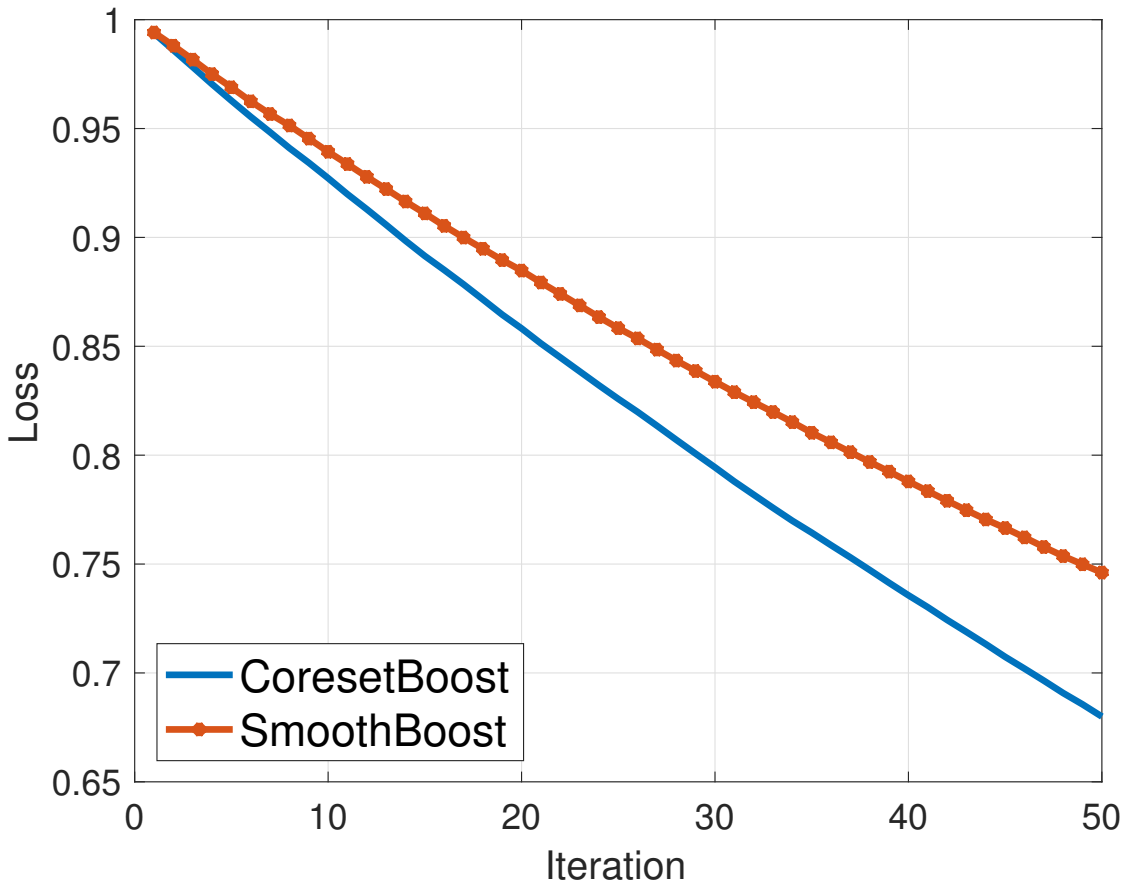


Figure 4.3: WebSpam training loss

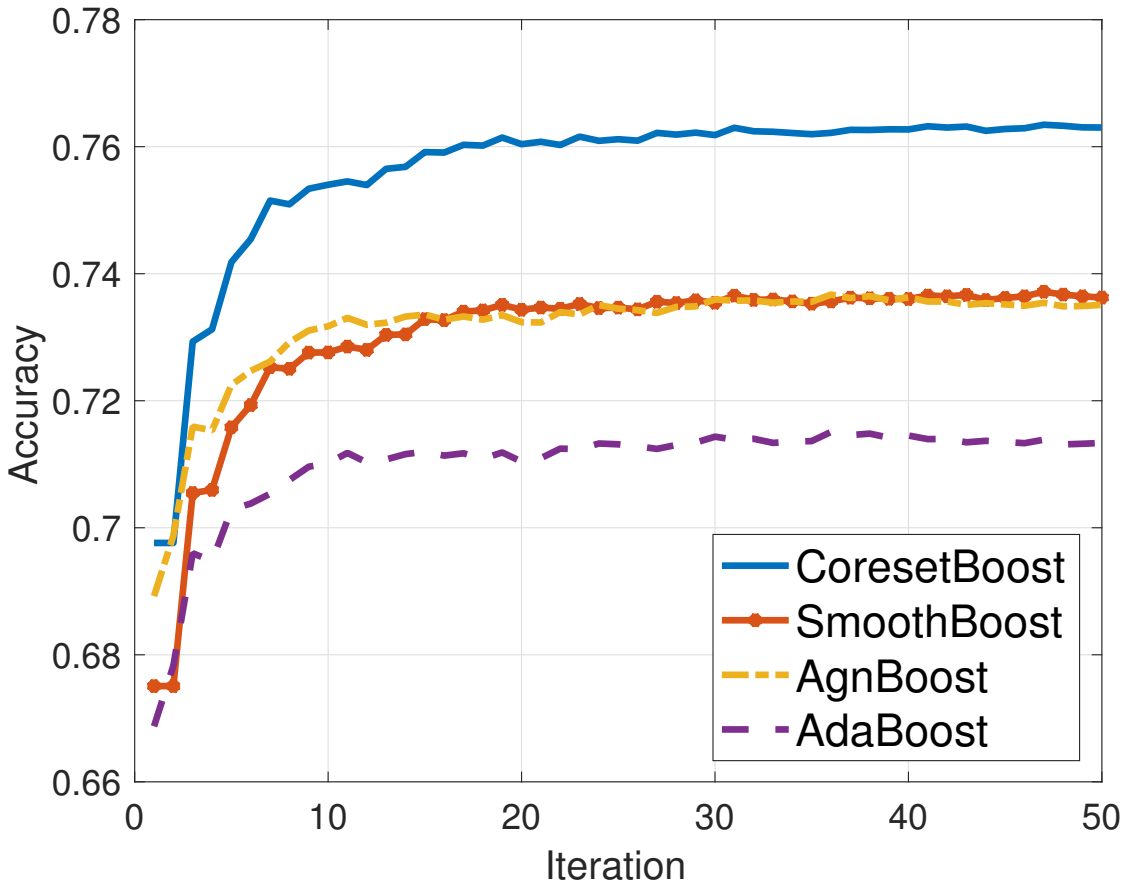


Figure 4.4: CovType training process

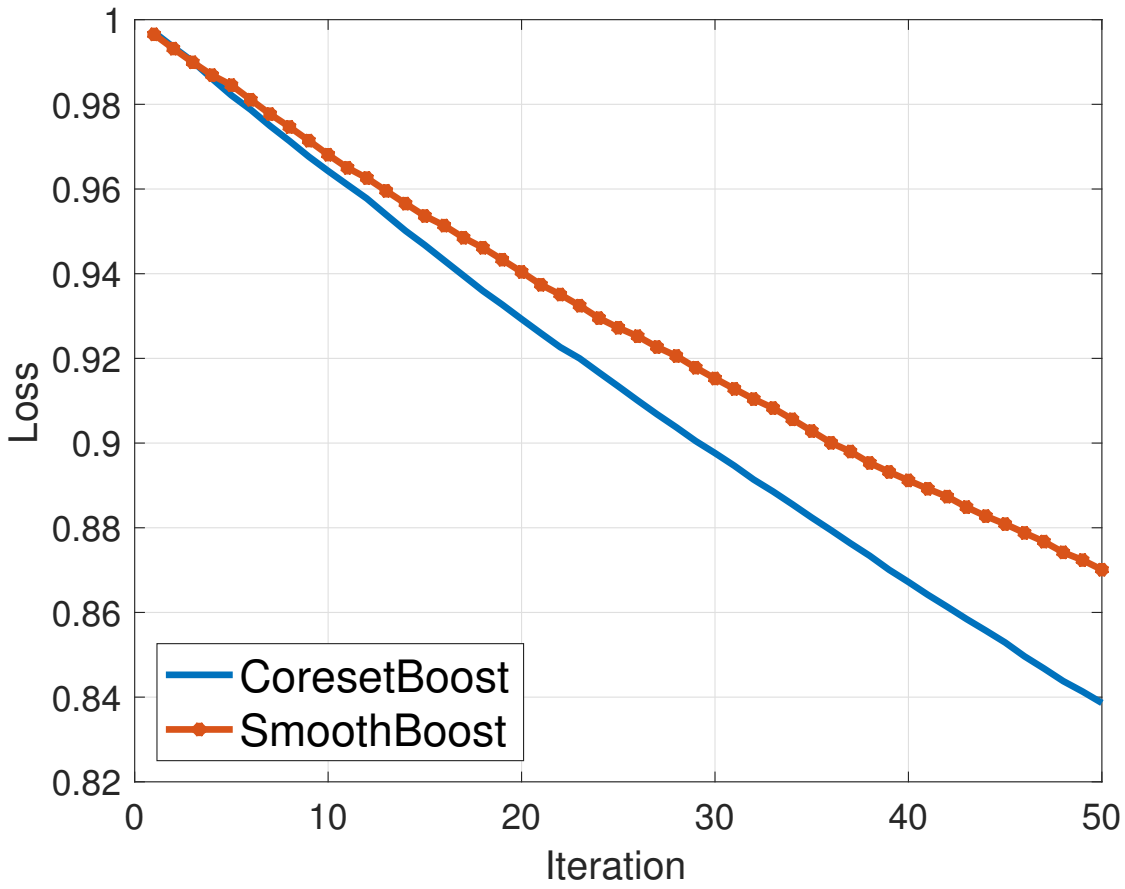


Figure 4.5: CovType training loss

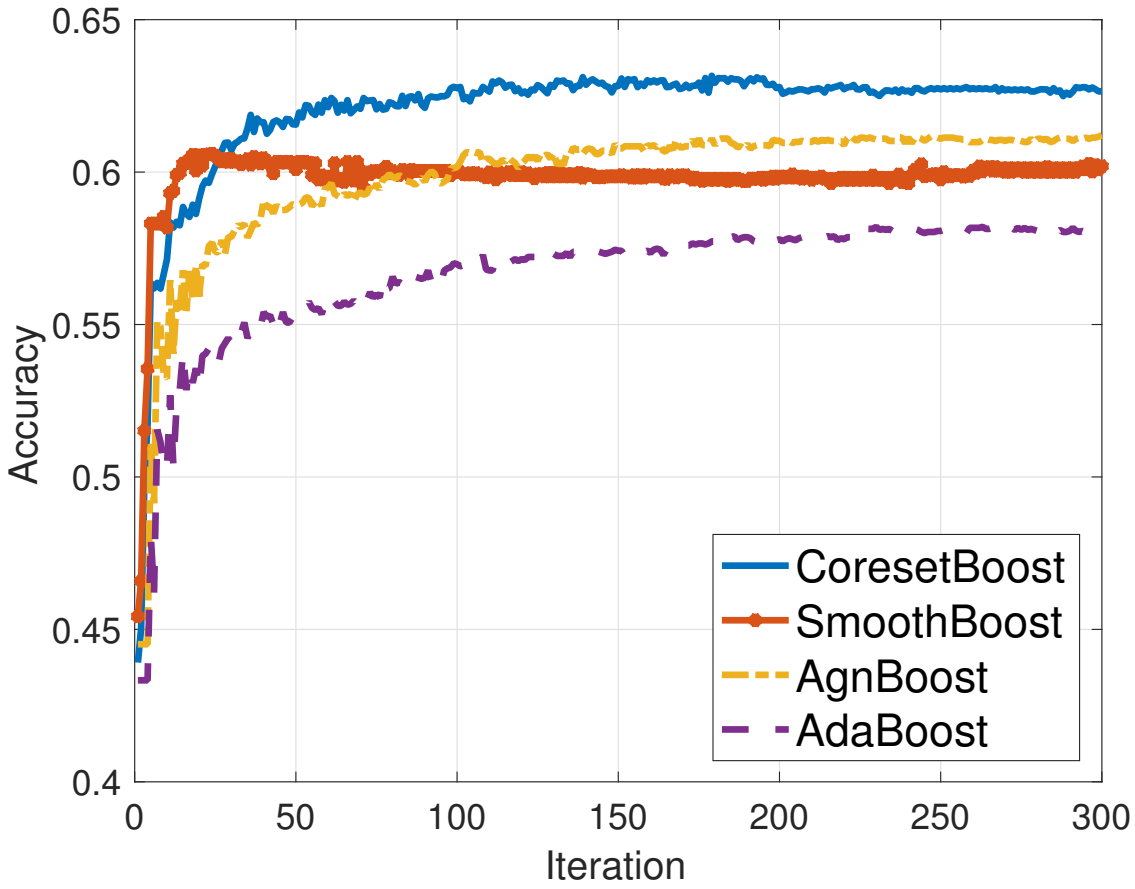


Figure 4.6: Yahoo training process

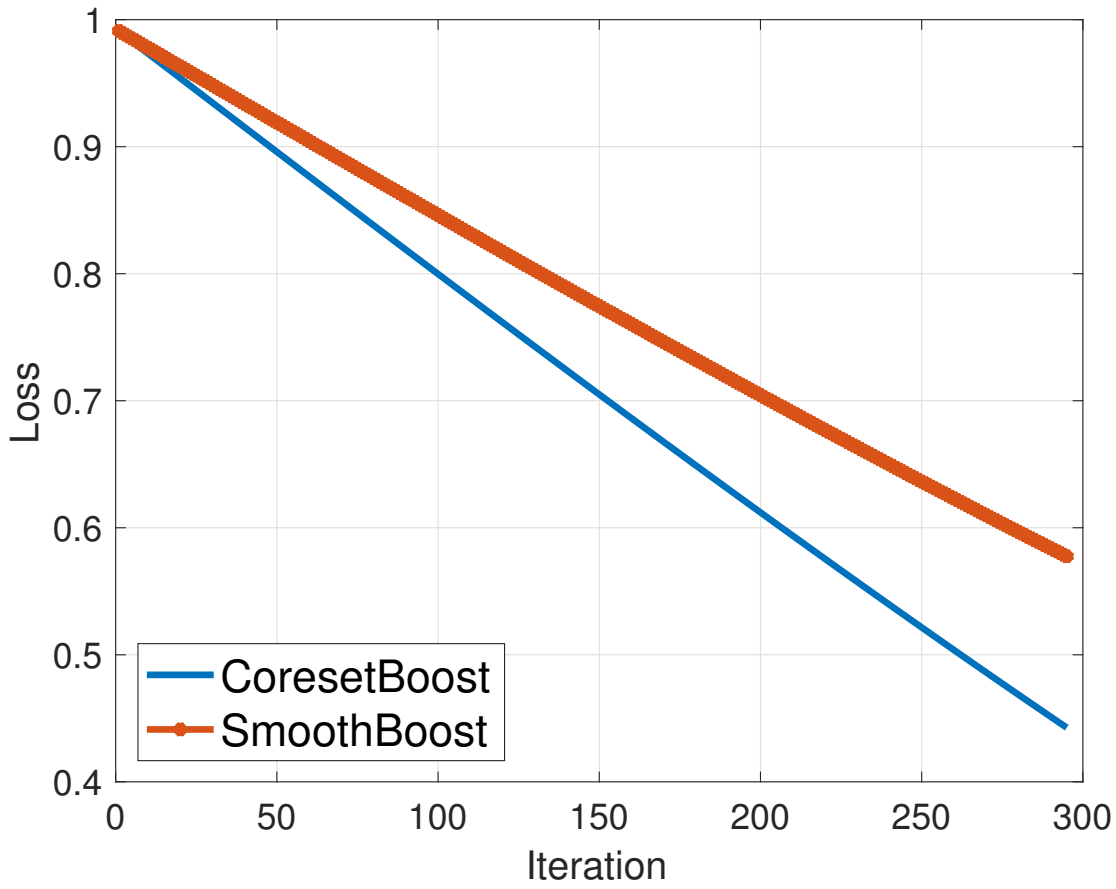


Figure 4.7: Yahoo training loss

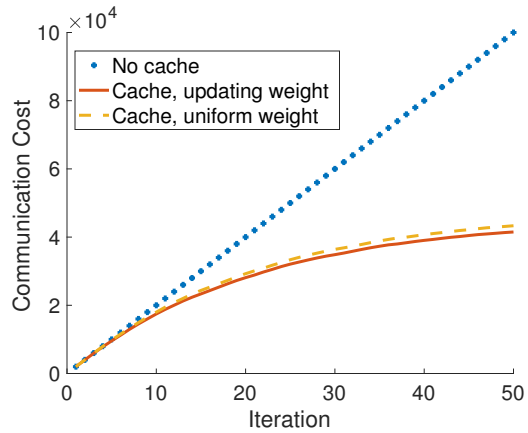


Figure 4.8: Communication cost

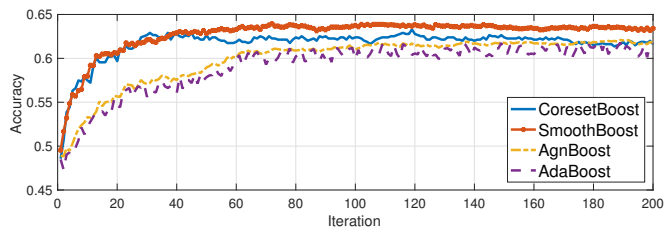


Figure 4.9: Peformance on adversary distribution

4.6.3 Communication Cost

Fig 4.8 shows the communication cost for the proposed distributed coreset boosting regarding the number of transmitted samples throughout the learning process on *Web*. The dataset is randomly split into four parts. In each iteration, the master node asks for the coreset with $|M| = 2000$ from the distributed nodes. As expected, the total communication cost is reduced by cache and the improvement is strengthened in boosting comparing to random sampling.

4.6.4 Robustness

At last we check the robustness of the proposed algorithm when the distribution of the sample is adversary. We sort the samples based on their first feature and split the data accordingly. This is the extreme case where the distribution on each node is totally different. As shown in Fig 4.9, the proposed learning framework could still achieve the high accuracy while the

other three baseline approach suffer large performance loss. Besides, the convergence rate for CoresetBoost is larger than the other three methods.

4.7 Proof

Here, the sensitivity $\sigma_n(\mathcal{H})$ is defined similarly in [51]

$$\sigma_n(\mathcal{H}) := \sup_{h \in \mathcal{H}} \frac{l(Y_n h(X_n))}{\sum_{l=1}^{|\mathcal{D}|} w_l l(Y_l h(X_l))} \tag{4.17}$$

Lemma 4.9. *Define $F_1(\boldsymbol{x}) = l(h(\boldsymbol{x}))$ and $F_2(\boldsymbol{x}) = l(-h(\boldsymbol{x}))$. $F_1(\boldsymbol{x})$ and $F_2(\boldsymbol{x})$ are convex.*

Proof.

$$\begin{aligned}
& F_1(\boldsymbol{x}) - F_1(\boldsymbol{y}) = l(h(\boldsymbol{x})) - l(h(\boldsymbol{y})) \\
\geq\ & l'(h(\boldsymbol{y}))[h(\boldsymbol{x}) - h(\boldsymbol{y})] + l'(h(\boldsymbol{y}))h'(\boldsymbol{y})^T|\boldsymbol{x} - \boldsymbol{y}| \\
-\ & l'(h(\boldsymbol{y}))h'(\boldsymbol{y})^T|\boldsymbol{x} - \boldsymbol{y}| && (4.18) \\
\geq\ & -\lambda l'(h(\boldsymbol{y}))||\boldsymbol{x} - \boldsymbol{y}||^2 + l'(h(\boldsymbol{y}))h'(\boldsymbol{y})^T|\boldsymbol{x} - \boldsymbol{y}| \\
-\ & l'(h(\boldsymbol{y}))h'(\boldsymbol{y})^T|\boldsymbol{x} - \boldsymbol{y}| && (4.19) \\
\geq\ & l'(h(\boldsymbol{y}))h'(\boldsymbol{y})^T|\boldsymbol{x} - \boldsymbol{y}| \\
-\ & l'(h(\boldsymbol{y}))[\lambda|\boldsymbol{x} - \boldsymbol{y}| + h'(\boldsymbol{y})]^T|\boldsymbol{x} - \boldsymbol{y}| && (4.20) \\
\geq\ & l'(h(\boldsymbol{y}))h'(\boldsymbol{y})^T|\boldsymbol{x} - \boldsymbol{y}| \\
\geq\ & F_1'(\boldsymbol{y})^T(\boldsymbol{x} - \boldsymbol{y}) && (4.21)
\end{aligned}$$

(4.18) is obtained from the convexity of $l(x)$. (4.19) is because $h(\boldsymbol{x})$ is λ-Lipschitz such that

$$h(\boldsymbol{x}) - h(\boldsymbol{y}) \geq -\lambda||\boldsymbol{x} - \boldsymbol{y}||^2$$

As $l(x)$ is nonincreasing and $h'(\boldsymbol{x}) \geq -\lambda|\boldsymbol{x} - \boldsymbol{y}|$, the second term in (4.20) is smaller than 0.

By similar manipulation, we could prove that $F_2(x)$ is convex.

Lemma 4.10. *For all a random vector $Z \in \mathbb{R}^D$ with finite mean $\bar{Z} = E[Z]$ and a fixed vectors V,*

$$\inf_{h \in \mathcal{H}} E_Z \left[\frac{l(h(Z))}{l(h(V))} \right] \geq e^{-\lambda ||\bar{Z} - V||^2} \tag{4.22}$$

where $h(x)$ has the Lipschitz constant λ.

Proof.

First since $l(h(x))$ is convex, using Jensen's inequality,

$$\inf_{h \in \mathcal{H}} E_Z \left[\frac{l(h(Z))}{l(h(V))} \right] \geq \inf_{h \in \mathcal{H}} \frac{l(h(E[Z]))}{l(h(V))}$$

Insert $\frac{l(a)}{l(b)} \geq e^{-|a-b|}, \forall a, b \in \mathbb{R}$, we have

$$\begin{aligned} \inf_{h \in \mathcal{H}} \frac{l(h(E[Z]))}{l(h(V))} &\geq \inf_{h \in \mathcal{H}} e^{-|h(E[Z]) - h(V)|} \\ &\geq e^{-\lambda ||E[Z] - V||^2} \end{aligned}$$

Lemma 4.11. *For any k-clustering Q,*

$$\sigma_n(\mathcal{H}) \leq \left[\frac{1}{w_n + \sum_{m=1}^{k} (W_k^{w,n} e^{-\lambda ||\bar{X}_{G,k}^{n,w} - X_n||^2} + W_k^{w,-n} e^{-4\eta^2})} \right]$$

Proof.

$$\begin{aligned} & \sigma_n(\mathcal{H})^{-1} \\ = & \inf_{h \in \mathcal{H}} \sum_{j=1}^{\mathcal{D}} \frac{w_j l(Y_j h(X_j))}{l(Y_n h(X_n))} \\ = & \inf_{h \in \mathcal{H}} \left[w_n + \sum_{j \neq n} \frac{w_j l(Y_j h(X_j))}{l(Y_n h(X_n))} \right] \\ = & \inf_{h \in \mathcal{H}} \left[w_n + \sum_{Y_j = 1} \frac{w_j l(h(X_j))}{l(Y_n h(X_n))} + \sum_{Y_j = -1} \frac{w_j l(-h(X_j))}{l(Y_n h(X_n))} \right] \end{aligned} \tag{4.23}$$

Algorithm 9 Coreset Sampling

1: **Input:** data $(X_i, Y_i) \in D$, the discrete distribution $\boldsymbol{p}$ such that $\sum_{i=1}^{|D|} p_i = 1$ calculated by coreset construction and the required number of samples $|M|$
2: **for** $t = 1 : |M|$ **do**
3: Independently sample one point according to the distribution $\boldsymbol{p}$
4: **end for**
5: **Output:** The coreset M

For all $(X_j, Y_i = 1)$ within cluster k, given $|G_k^+|$ is sufficiently large and using Lemma 4.10 with Jensen's inequality

$$\begin{aligned}\sum_{(X_j,Y_i)\in G_k^+} w_j l(h(X_j)) &= |W_k^{w,+}| E[l(h(X))] \\ &\geq |W_k^{w,+}| l(h(E[X])) \\ &\geq |W_k^{w,+}| l(h(\bar{X}_{G,k}^{n,w}))\end{aligned}$$

Similarly we have

$$\sum_{(X_j,Y_i)\in G_k^-} w_j l(-h(X_j)) \geq |W_k^{w,-}| l(-h(E[\bar{X}_{G,k}^{n-,w}]))$$

Here

$$W_k^{w,+} = \sum_{(X_i,Y_i)\in G_k^+} w_j \tag{4.24}$$

Insert into (4.23)

$$\begin{aligned}&\sigma_n(\mathcal{H})^{-1} \\ \geq\ & \inf_{h\in\mathcal{H}} \Bigg[w_n + \sum_{m=1}^{k} |W_k^{w,+}| \frac{l(h(\bar{X}_{G,m}))}{l(Y_n h(X_n))} \\ +\ & \sum_{m=1}^{k} |W_k^{w,-}| \frac{l(-h(\bar{X}_{G,m}))}{l(Y_n h(X_n))} \Bigg]\end{aligned}$$

4.7.1 Proof for Theorem 2

The 0-1 loss $Err_{\mathcal{D}(h)}$ is upper bounded by $L_{sm}(h)$ since $l_{sm}(yh(x)) \geq \mathbf{1}_{h(x)\neq y}$. Instead of handling the 0-1 loss directly, we will prove in each iteration, $L_M(h)$ decreases by larger than $O(\epsilon^{2-2c})$ with high probability. First apply Taylor expansion on $l(x)$ with $l''(x) \leq 1$

$$l(x) - l(x + \Delta x) \geq -\Delta x l'(x) - \frac{\Delta x^2}{2}$$

Let x be $\sum_{t=1}^{T-1} \gamma_t Y_n h^t(X_n)$ and Δx be $\gamma_T Y_n h^T(X_n)$,

$$\begin{aligned} & l(H^{T-1}(X_n)Y_n) - l(H^{T-1}(X_n)Y_n + \gamma_T Y_n h^T(X_n)) \\ \geq\ & \gamma_T Y_n h^T(X_n)[-l'(H^{T-1}(X_n)Y_n)] - \frac{\gamma_T^2 (Y_n h^T(X_n))^2}{2} \\ \geq\ & \gamma_T Y_n h^T(X_n) w_n^T - \frac{\gamma_T^2 \eta^2}{2} \end{aligned}$$

This is the direct result of the definition for w_n^T in Algorithm 2 and the fact that $Y_n h^T(X_n) \leq \eta$. Take the expectation of both sides and assume the initial distribution for each sample is $\frac{1}{|\mathcal{D}|}$. Then we have

$$\begin{aligned} & L_{sm}(H^{T-1}) - L_{sm}(H^T) \\ =\ & E_0\big[l(H^{T-1}(X_n)Y_n)\big] - E_0\big[l(H^T(X_n)Y_n)\big] \\ \geq\ & \frac{\gamma_T W^T E^T[Y h^T(X)]}{|\mathcal{D}|} - \frac{\gamma_T^2 \eta^2}{2} \end{aligned}$$

By choosing $\gamma_T = \frac{W^T E^T[Y h^T(X)]}{N\eta^2}$, the maximum value for the right side of the equation could be achieved.

To complete the proof, we need to verify that given the assumption h^t is generated on the coreset, $2\frac{(W^T E^T[Y h^t(X)])^2}{|\mathcal{D}|^2 \eta^2}$ is in the order of $O(\epsilon^{2-2c})$.

Lemma 4.12. *Assume in each iteration t we could always find a base function h^t based on the coreset such that the corresponding smooth loss $\hat{L}_{sm}^M(h^t) \leq (1+\beta)(1-\alpha)$. Then with probability $1-\delta$,*

$$W^t E^t[h^t(X)Y] \geq |\mathcal{D}|\alpha(\min_{h\in H} Err(h) + \epsilon)^{1-c} \tag{4.25}$$

Remark. h^t is K-bounded. The typical value for K is 1 as weak classifier and the assumption in Lemma 4.12 holds in this case. For larger K, it is not hard to find the required $h^t(x)$ by solving $\arg\min_{h \in H} \hat{L}^M_{sm}(h)$.

4.7.2 Definition of ϵ-Approximation

This is defined in [70]. Suppose that $D = \left\{(X_n, Y_n)_{n=1}^{|D|}\right\}$ is a dataset. Let H be the set of classifiers. We first define the likelihood of observation (X_n, Y_n) given the classifier $h \in H$ as

$$p(Y_n|X_n; h) = \frac{1}{1 + \exp(-Y_n h(X_n))} \tag{4.26}$$

Then the likelihood of the whole datset D given classifier h(x) could be calculated as

$$L(h, D) = \prod_{n=1}^{|D|} p(Y_n|X_n; h) \tag{4.27}$$

and the log-likelihood as

$$\mathcal{L}(h) = \sum_{n=1}^{|D|} \log p(Y_n|X_n; h(X_n)). \tag{4.28}$$

The target of coreset is to construct the subset of the whole dataset M such that

$$|\hat{\mathcal{L}}(h) - \mathcal{L}(h)| \leq \epsilon, \forall h \in H \tag{4.29}$$

with high probability, where

$$\hat{\mathcal{L}}(h) = \sum_{(X_n, Y_n) \in M} r_n \log p(Y_n|X_n; h) \tag{4.30}$$

We say that M is an ϵ-approximation of D given the classifier space H.

4.7.3 Proof of Corollary 1

It is straightforward to prove $h(X)$ satisfies the assumption. Since $\|\theta\| \leq r$ and $\|X\| = 1$, we have $h(X) \leq r$.

Then we have

$$\begin{aligned} \|h(X_1) - h(X_2)\| &= |\theta(X_1 - X_2)| \\ &\leq \|\theta\|\|(X_1 - X_2)\| \\ &\leq r\|(X_1 - X_2)\| \end{aligned} \tag{4.31}$$

Therefore we prove $h(x)$ is r-Lipschit and r-bounded. Now we move to prove that $\frac{l(x)}{l(y)} \leq e^{|x-y|}$. Let $\rho(x) = \frac{l(a)}{l(a+\Delta)}$. We have

$$\rho'(x) = \frac{(1+e^x)log(1+e^{-x}) - (1+e^{x+\Delta})\log(1+e^{-x-\Delta})}{(1+e^x)(1+e^{x+\Delta})log^2(1+e^{-x-\Delta})} \tag{4.32}$$

We see that $\text{sgn}(\rho'(x)) = \text{sgn}(\Delta)$. For $\Delta > 0$

$$\begin{aligned} \sup_x \frac{l(x)}{l(x+\Delta)} &= \lim_{x\to+\infty} \frac{l(x)}{l(x+\Delta)} \\ &= \lim_{x\to+\infty} \frac{l'(x)}{l'(x+\Delta)} \\ &= \lim_{x\to+\infty} \frac{e^{-x}}{1+e^{-x}} \frac{1+e^{-x-\Delta}}{e^{-x-\Delta}} \\ &= e^{\Delta} \end{aligned} \tag{4.33}$$

Similarly for $\Delta < 0$,

$$\begin{aligned} \sup_x \frac{l(x)}{l(x+\Delta)} &= \lim_{x\to-\infty} \frac{l(x)}{l(x+\Delta)} \\ &= \lim_{x\to-\infty} \frac{l'(x)}{l'(x+\Delta)} \\ &= \lim_{x\to-\infty} \frac{e^{-x}}{1+e^{-x}} \frac{1+e^{-x-\Delta}}{e^{-x-\Delta}} \\ &= 1 \end{aligned} \tag{4.34}$$

Then we prove that $\frac{l(x)}{l(y)} \leq e^{|x-y|}$.

Chapter 5

Selective Sampling Based Efficient Classifier Representation in Distributed Learning

5.1 Introduction

Modern machine learning system aims to solve practical problems with large data size and high dimension. However, using a single machine to store and compute such large scale data set becomes prohibitively difficult. Distributed learning has recently attracted substantial interest. Some distributed learning frameworks ([85][168][22][111]) have been proposed to speed up the learning process by parallelizing the computation among the data distributed across different locations or entities. In such a setting, the communication within the network becomes a bottleneck for the design of stable and efficient distributed learning. Usually these frameworks apply the techniques of distributed learning, which decentralizes the traditional algorithms and finally obtains the optimal classifier at a master node.

In our paper, we consider a novel approach: each node storing a portion of the data sends out a message indicating the scope of locally good classifiers based on the local data, such that the set of globally good classifiers is obtained by the intersections of sets of locally

good classifiers. Such a scheme has the following two advantages when compared with the traditional approaches:

- It provides an intuitive and straightforward framework for distributed learning.
- Such a scheme can provide a set of good classifiers, instead of a single one, similarly to the list decoding in channel coding.

The challenge here is how to efficiently represent the sets of locally good classifiers, which can essentially be viewed as a source coding problem in communication systems. The task is difficult in the setting since the classifiers are usually functions (e.g., the linear classifiers are the linear functions), thus requiring the representation in the function space. One approach is to parameterize the classifiers and describe the sets of locally good classifiers in the real parameter space. However, it could be prohibitively difficult to describe a complicate region in real spaces.

In this paper, we carry out the source coding using selective sampling of local data, namely for any node a subset of samples in the local database is selected and then sent to other nodes. A classifier is locally good if it performs well over these selected samples. If we consider the hypothesis space (namely the space of the classifiers) and the sample space as being mutually dual spaces, essentially we are using a dual space to describe the space under study. Such a dual space based description has widely been used in mathematics (e.g., the definition of weak convergence in probability theory).

Under this selective sample based source coding framework, the main challenge is how to efficiently select the samples, in order to achieve a good tradeoff between the communication requirement (since more samples require more communication resources) and the description distortion (since less samples bring more errors to the set of locally good classifiers). Our approach is to formulate the sample selection as an optimization problem, and then simplify it by approximations such that a simple greedy algorithm can be applied. Note that once these samples are selected, it is possible to further consider the compression of these samples. The possible compression is not considered in this paper, but will be studied in our future research.

In more details, our main contributions to the framework of source coding include

- We define an innovative probability to measure the quality of the selected sample in classification.
- A weight based optimization problem is formulated to minimize the upper bound of the proposed probability and the optimal weight could be used to select the informative samples. An efficient algorithm is proposed to find the optimal weight. With high confidence, the learning result from the sampled data is close to the optimal result.
- Detailed performance analysis, implementation concerns and simulation results on synthetic and real world data are provided.

The remainder of the paper is organized as follows. Studies related to this paper is introduced in Section 5.2. The system model is briefed in Section 5.3. Then, the proposed framework of source coding for distributed learning is detailed in Section 5.4. The numerical results and conclusions are provided in Sections 5.5 and 5.6, respectively.

5.2 Related Work

Balcan et al. [17] are the first to consider communication as one of the fundamental resources in distributed learning, and they applied a theoretical analysis on the communication complexity based on Probably Approximately Correct (PAC) theory [77]. [34] extends [17] to design a noise-tolerant distributed boosting algorithm with communication complexity $O(\log \frac{1}{\epsilon})$, which shares the similar idea with our work, since in each iteration the algorithm adaptively changes the weight of each sample according to its importance. [109] proposed a screening algorithm for support vector machine (SVM) to eliminate the non-support point before learning, and to efficiently decrease the the training sample size. However, their algorithm utilizes the structure of SVM and could not be applied to other classifiers. In the field of database, to improve the efficiency of each query and maintain a fixed sampling budget, similar sampling algorithms are proposed. [75] introduces the Horvitz-Thompson (HT) estimator and formulates an optimization problem to allocate distribution to the available data. The data center selects a subset of samples according to the distribution for the future coming queries. Although their work is similar to our as both formulate a

weight based optimization problem, the concerns and the target functions are substantially different in these two papers.

5.3 System Model

In this section, we introduce the system model in this paper.

5.3.1 Classification

In the classification setting, we consider learning with respect to a distribution over $X \times Y$ and assume X to be countable and $Y = \{-1, 1\}$. The learning is with respect to some class of functions, H, where each $h \in H$ is a binary classifier $h : X \to \{-1, 1\}$. Generically speaking, the goal is to find the optimal classifier h_{opt} with the least error $R(h)$

$$h_{opt} = \min_{h \in H} E_{(x,y)}[\mathbf{1}_{\boldsymbol{h(x) \neq y}}] = \min_{h \in H} R(h). \tag{5.1}$$

Usually the true distribution of X and Y is unknown and we are able to access the data D generated by the distribution. To solve the classification problem, typically, an optimization problem based on the available data is formulated as

$$h_{em} = \min_{h \in H} \sum_{(x_i, y_i) \in D} L(x_i, y_i), \tag{5.2}$$

where $L(x, y)$ is the cost function depending on the model of the classifier. (5.2) could be solved by some standard gradient descent method like [26]. The computation cost increases with the size of the training data $|D|$.

5.3.2 Network

In the distributed learning setting, the training data is not stored at a centralized location. Communication between the distributed nodes then becomes the bottleneck for the learning problem. For simplicity, we assume that there are only two separated nodes storing samples, $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ (hence $\boldsymbol{D} = \boldsymbol{D}_1 \cup \boldsymbol{D}_2$), generated by the same distribution and the same mapping

from x to y. The principle adopted in this paper can be extended to more generic networks having more nodes and arbitrary network topology. Since there are only two nodes, we can consider a single round of transmission from node 1 to node 2; then, node 2 can feed back its learning result, based on both the message from node 1 and its local data, back to node 1, which results in very light communications.

5.3.3 Division of Classifier Space

In typical classification problems, the space of classifier is continuous, since the parameters of classifiers are usually real numbers. For example, the linear classifiers with n weights form an n-dimensional vector space; if normalization of the weights are taken into consideration, the classifier space is the n-dimensional sphere.

Although the classifiers could be continuous functions, the classifier space can be considered as being countable due to the limited number of samples; i.e., the classifier space can be partitioned to finitely many subsets $\{H_n\}_{n=1,...,N_1}$ such that

$$h(x_1) = h(x_2), \qquad \forall x_1, x_2 \in \boldsymbol{D}_1, h \in H_n, n = 1, ..., N_1, \tag{5.3}$$

namely the classifiers within the same subset generate the same classification results for all the samples. Hence, the classifiers within the same subset H_n can be considered as being equivalent. The corresponding error probability of classification is then denoted by $\mathcal{E}_n$ for H_n. For simplicity, we assume

$$\mathcal{E}_1 \leq ... \leq \mathcal{E}_{K_1} < \alpha < \mathcal{E}_{K_1+1} \leq ... \leq \mathcal{E}_{N_1}. \tag{5.4}$$

This assumption is similar with the assumption of finite Vapnik-Chervonenkis (VC) dimension [144] for the classifier. Under this assumption and H is countable, we introduce a discrete probability space $(\{H_n\}_{n=1,...,N_1}, P)$, where P is the prior probability of the equivalence classes of classifiers. Due to the finiteness of H, the probability is well defined. We further assume a uniform prior distribution for each subset such that $P(H_n) = \frac{1}{N_1}$ due to the lack of prior information on the classifiers.

5.3.4 Learning Goal: Good Classifiers

Slightly different from many traditional algorithms, we change the goal of finding the optimal classifier to looking for a set of good classifiers. We say that a classifier is **globally good** if its error probability $R(h)$ over the whole data set $\boldsymbol{D}$ is smaller than threshold α. Suppose node 1 selects a subset of samples $\boldsymbol{S} \subset \boldsymbol{D}_1$. The classifier is **locally good** on the selected sample $\boldsymbol{S}$ if its error $R_S(h)$ is smaller than threshold α, where $R_S(h) = \frac{1}{|S|}\sum_{(x_i,y_i)\in S} \mathbf{1}_{\boldsymbol{h(x_i)\neq y_i}}$. Then subsets $\hat{\mathcal{H}}_0(S)$, $\hat{\mathcal{H}}_1(S)$, $\mathcal{H}_0$ and $\mathcal{H}_1$ are defined to represent globally/locally good classifier set and bad classifier set, or more precisely

$$\begin{cases} \mathcal{H}_0 = \{h \in H | R(h) \leq \alpha\} \\ \mathcal{H}_1 = \{h \in H | R(h) > \alpha\} \\ \hat{\mathcal{H}}_0(\boldsymbol{D}_1) = \{h \in H | R_{\boldsymbol{D}_1}(h) \leq \alpha\} \\ \hat{\mathcal{H}}_1(\boldsymbol{D}_1) = \{h \in H | R_{\boldsymbol{D}_1}(h) > \alpha\} \\ \hat{\mathcal{H}}_0(\boldsymbol{D}_2) = \{h \in H | R_{\boldsymbol{D}_2}(h) \leq \alpha\} \\ \hat{\mathcal{H}}_1(\boldsymbol{D}_2) = \{h \in H | R_{\boldsymbol{D}_2}(h) > \alpha\} \\ \hat{\mathcal{H}}_0(\boldsymbol{S}) = \{h \in H | R_S(h) \leq \alpha\} \\ \hat{\mathcal{H}}_1(\boldsymbol{S}) = \{h \in H | R_S(h) > \alpha\} \end{cases} . \tag{5.5}$$

To measure the quality of the selected sample S, we use the probability measure mentioned in 5.3.3 and define the error probability of the classification of classifiers as

$$\begin{aligned} P_{err}(S) &= P(h \in \hat{\mathcal{H}}_0(\boldsymbol{D}_1), h \in \hat{\mathcal{H}}_1(S)) \\ &+ P(h \in \hat{\mathcal{H}}_1(\boldsymbol{D}_1), h \in \hat{\mathcal{H}}_0(S)). \end{aligned} \tag{5.6}$$

Note that this error rate is different from the classification errors: P_{err} is over the classifiers h and R is over the data (x, y).

Intuitively, this probability indicates the confidence of the classifier h generated by the sample S. A large $P_{err}(S)$ indicates that if the classifier h has a good performance on S, with high confidence, it has good performance on D and vice versa. We will use $P_{err}(S)$ to measure the quality of the selected samples.

5.3.5 Learning and Communication Model

Based on the above definitions, the procedure of learning is: node 1 chooses a subset of samples $\boldsymbol{S} \in \boldsymbol{D}_1$ such that $P_{err}(\boldsymbol{S})$ can well approximate $R_{\boldsymbol{D}_1}$ (or equivalently $\hat{\mathcal{H}}_0(\boldsymbol{S})$ is very similar to $\hat{\mathcal{H}}_0(\boldsymbol{D}_1)$; then node 2 estimates the set of globally optimal classifiers using $\hat{\mathcal{H}}_0(\boldsymbol{S}) \cap \hat{\mathcal{H}}_0(\boldsymbol{D}_2)$.

5.4 Classifier Representation Via Sample Selection

In this section, we formulate the sample selection based Classifier Representation into an optimization problem. We focus on node 1 which has the data set $\boldsymbol{D}_1$.

5.4.1 Formulation and Simplification

Our goal is finding a subset of samples in $\boldsymbol{D}_1$ that minimizes the proposed error probability P_{err}

$$D_{opt} = \arg\min_{\boldsymbol{S} \in \boldsymbol{D}_1} P_{err}(\boldsymbol{S}). \tag{5.7}$$

However, the optimization problem (5.7) is difficult to solve since it is essentially a discrete optimization due to the limited number of samples in $\boldsymbol{D}_1$. Hence, we change to find a mathematically and algorithmically tractable upper bound and then optimize the upper bound. To that goal, as $\mathcal{H}_0 = \cap_{n=1}^{K_1} H_n$ and $\mathcal{H}_1 = \cap_{n=K_1+1}^{N_1} H_n$, we rewrite the expression of P_{err} as

$$\begin{aligned} P_{err}(\boldsymbol{S}) &= \sum_{n=1}^{K_1} P(h \in \hat{\mathcal{H}}_1(S) | h \in H_n) P(H_n) \\ &+ \sum_{n=K_1+1}^{N_1} P(h \in \hat{\mathcal{H}}_0(S) | h \in H_n) P(H_n), \end{aligned} \tag{5.8}$$

where $P(H_n) = P(h \in H_n)$ is the prior probability of the classifier.

Next, we handle the first probability in (5.8). As $h \in \mathcal{H}_0$, we have

$$\mathcal{E}(h, \boldsymbol{D}_1) < \alpha, \forall h \in \mathcal{H}_0. \tag{5.9}$$

Here we define the gap $K_S(h)$ between the classification error rates over the entire data $\boldsymbol{D}_1$ and sampled data $\boldsymbol{S}$ as

$$\begin{aligned} K_S(h) &= R_S(h) - R(h) \\ &\geq \alpha - \mathcal{E}(h, \boldsymbol{D}_1). \end{aligned} \tag{5.10}$$

We consider $K_S(h)$ as a random variable whose randomness stems from the random selection of h. Fix S and suppose $h \in H_n$, then $\mathcal{E}(h, \boldsymbol{D}_1) = \mathcal{E}_n$. Hence, the conditional expectation of $K_S(h)$ is given by

$$E[K_S(h)|h \in H_n] = E\left[\frac{1}{|\boldsymbol{S}|}\sum_{(x_i,y_i)\in\boldsymbol{S}} \mathbf{1}_{h(x_i)\neq y_i}\right] - \mathcal{E}_n, \tag{5.11}$$

while its variance is bounded by

$$\begin{aligned} V[K_S(h)|h \in H_n] &\leq E[K_S^2(h)|h \in H_n] \\ &\leq \left(\max\left\{\frac{|\boldsymbol{D}_1|}{|\boldsymbol{S}|} - 1, 1\right\}\mathcal{E}_n\right)^2, \end{aligned} \tag{5.12}$$

where the last inequality is due to

$$\begin{aligned} & \left(\frac{1}{|\boldsymbol{S}|}\sum_{(x_i,y_i)\in\boldsymbol{S}} \mathbf{1}_{h(x_i)\neq y_i} - \frac{1}{|\boldsymbol{D}_1|}\sum_{(x_i,y_i)\in\boldsymbol{D}_1} \mathbf{1}_{h(x_i)\neq y_i}\right)^2 \\ = & \left((\frac{1}{|\boldsymbol{S}|} - \frac{1}{|\boldsymbol{D}_1|})\sum_{(x,y)\in\boldsymbol{S}} \mathbf{1}_{h(x)\neq y} - \frac{1}{|\boldsymbol{D}_1|}\sum_{(x_i,y_i)\notin\boldsymbol{S}} \mathbf{1}_{h(x_i)\neq y_i}\right)^2 \\ \leq & \left(\max\left\{\frac{1}{|\boldsymbol{S}_1|} - \frac{1}{|\boldsymbol{D}_1|}, \frac{1}{|\boldsymbol{D}_1|}\right\}\sum_{(x_i,y_i)\in\boldsymbol{D}_1} \mathbf{1}_{h(x_i)\neq y_i}\right)^2 \\ = & \left(\max\left\{\frac{1}{|\boldsymbol{S}|} - \frac{1}{|\boldsymbol{D}_1|}, \frac{1}{|\boldsymbol{D}_1|}\right\}|\boldsymbol{D}_1|\mathcal{E}_n\right)^2 \\ = & \left(\max\left\{\frac{|\boldsymbol{D}_1|}{|\boldsymbol{S}|} - 1, 1\right\}\mathcal{E}_n\right)^2. \end{aligned} \tag{5.13}$$

We believe that the uniform distribution is reasonable for $P(H_n)$ and thus it has no impact on our choice of sample. Then we focus on the first part of the equation. By applying the

Chebyshev's Inequality, we have

$$\begin{aligned} & P\left(R_S(h) \geq \alpha | h \in H_n\right) \\ = \; & P(K_S(h) \geq \alpha - \mathcal{E}_n | h \in H_n) \\ \leq \; & \frac{V[K_S(h)|h \in H_n]}{\left(\alpha - \mathcal{E}_n - (E[K_S(h)|h \in H_n])\right)^2}. \end{aligned} \tag{5.14}$$

We denote $\frac{V[K_S(h)|h \in H_n]}{(\alpha - \mathcal{E}_n - (E[K_S(h)|h \in H_n]))^2}$ by $M_S(h)$, which is given by

$$M_S(h) = \frac{V[K_S(h)|h \in H_n]}{(\alpha - \mathcal{E}_n)^2} \frac{1}{(1 - \frac{E[K_S(h)|h \in H_n])}{\alpha - \mathcal{E}_n})^2}. \tag{5.15}$$

The upper bound for $M_S(h)$ is obtained in the following lemma.

Lemma 5.1. *If $\mathcal{E}_n \in [\mathcal{E}_{min}, \frac{\alpha + \mathcal{E}_{min}}{2}]$, $n = 1, 2, \ldots K_1$, where $\mathcal{E}_{min}$ is the error rate of the optimal classifier on the selected sample, then the value of $M_S(h)$ is upper bounded by*

$$M_S(h) \leq \frac{2E[K_S(h)|h \in H_n])}{(\alpha - \mathcal{E}_n)^3}. \tag{5.16}$$

Proof. As indicated in (5.15), given a fixed set of classifier H_n, the first part is bounded by a constant provided by (5.12). For the second part, according to the definition of $K_S(h)$ in (5.10), we have

$$\begin{aligned} & E[K_S(h)|h \in H_n] \\ = \; & E[R_S(h)|h \in H_n] - E[R(h)|h \in H_n] \\ = \; & E[R_S(h)|h \in H_n] - \mathcal{E}_n. \end{aligned} \tag{5.17}$$

Given (5.9) and the upper bound in Lemma 5.1, $\frac{E[K_S(h)|h \in H_n])}{\alpha - \mathcal{E}_n} \in (-1, 1)$ for all $E[R_S(h)|h \in H_n] \in [\mathcal{E}_{min}, 1)$. Applying the Taylor's expansion to the second term of (5.15), we have

$$M_S(h) = \frac{V[K_S(h)|h \in H_n]}{(\alpha - \mathcal{E}_n)^2}$$

$$
\begin{aligned}
&+ \quad \frac{2V[K_S(h)|h \in H_n]E[K_S(h)|h \in H_n])}{(\alpha - \mathcal{E}_n)^3} \\
&+ \quad \sum_{i=2}^{\infty} O\left(\frac{E^i[K_S(h)|h \in H_n])}{(\alpha - \mathcal{E}_n)^{2+i}}\right) \\
&\leq \quad \left(\max\left\{\frac{|\boldsymbol{D}_1|}{|\boldsymbol{S}|} - 1, 1\right\}\mathcal{E}_n\right)^2 \\
&\times \quad \left(\frac{1}{(\alpha - \mathcal{E}_n)^2} + \frac{2E[K_S(h)|h \in H_n])}{(\alpha - \mathcal{E}_n)^3}\right) \\
&+ \quad \sum_{i=2}^{\infty} O\left(\frac{E^i[K_S(h)|h \in H_n])}{(\alpha - \mathcal{E}_n)^{2+i}}\right). \qquad (5.18)
\end{aligned}
$$

As the the other part of (5.18) is predefined when the sample size is fixed, the selection of sample only influences $\frac{2E[K_S(h)|h \in H_n]}{(\alpha - \mathcal{E}_n)^3}$ which consequently controls the upper bound of $P(h \in \hat{\mathcal{H}}_1(S)|h \in H_n)$. □

The upper bounder of the first part in (5.6) is obtained as follows:

$$
\begin{aligned}
& P(h \in \mathcal{H}_0, h \in \hat{\mathcal{H}}_1(S)) \\
\leq \quad & C + 2\left(\max\left\{\frac{|\boldsymbol{D}_1|}{|\boldsymbol{S}|} - 1, 1\right\}\right)^2 \\
\times \quad & \sum_{n=1}^{K_1} \frac{\mathcal{E}_n^2 P(H_n)}{(\alpha - \mathcal{E}_n)^3} E\left[\frac{1}{|\boldsymbol{S}|}\sum_{(x,y)\in S} \mathbf{1}_{h(x)\neq y}|h \in H_n\right] \\
+ \quad & \sum_{n=1}^{K_1}\sum_{i=2}^{\infty} O\left(\frac{E^i[K_S(h)|h \in H_n])}{(\alpha - \mathcal{E}_n)^{2+i}}\right), \qquad (5.19)
\end{aligned}
$$

With similar manipulation and the assumption $\mathcal{E}_n \in [\frac{\alpha+1-\mathcal{E}_{min}}{2}, 1 - \mathcal{E}_{min}]$ on the second part of P_{err}, we can also prove

$$
\begin{aligned}
& P(h \in \mathcal{H}_1, h \in \hat{\mathcal{H}}_0(S)) \\
\leq \quad & C' - 2\left(\max\left\{\frac{|\boldsymbol{D}_1|}{|\boldsymbol{S}|} - 1, 1\right\}\right)^2 \\
\times \quad & \sum_{n=K_1}^{N_1} \frac{\mathcal{E}_n^2 P(H_n)}{(\mathcal{E}_n - \alpha)^3} E\left[\frac{1}{|\boldsymbol{S}|}\sum_{(x,y)\in \boldsymbol{S}} \mathbf{1}_{h(x)\neq y}|h \in H_n\right] \\
+ \quad & \sum_{n=K_1}^{N_1}\sum_{i=2}^{\infty} O\left(\frac{E_S^i[D(h)|h \in H_n]}{(\mathcal{E}_n - \alpha)^{2+i}}\right), \qquad (5.20)
\end{aligned}
$$

Remark 2. *In the proposed framework, the classifier sets with error rate* $\mathcal{E}_n \in (\frac{\alpha+\mathcal{E}_{min}}{2}, \frac{1+\alpha-\mathcal{E}_{min}}{2})$ *are not considered. This assumption also promises the Taylor approximation in (5.18). As the purpose of sampling is to find the subset of data that well describes the classifier space and to use these samples from distributed nodes to find the globally optimal classifier, this is acceptable if we set* α *to be large enough. This is because the discrimination of classifier between* $\{h \in H_n | \mathcal{E}_n \in (\frac{\alpha+\mathcal{E}_{min}}{2}, \alpha)\}$*, which could be considered as good classifier, and* $\{h \in H_n | \mathcal{E}_n \in (\alpha, \frac{1+\alpha-\mathcal{E}_{min}}{2})\}$*, which could be considered as bad classifier, is less related to the optimal classifier, and the corresponding samples are less informative therefore.*

Based on the above analysis, we obtain an upper bound for P_{err} in the following theorem, which is more mathematically tractable.

Theorem 5.2. *For the setup in this paper, the error rate of classifying the classifiers is upper bounded by*

$$\begin{aligned} P_{err} &\leq C'' + \sum_{n=1}^{N_1} c_n E\left[\frac{1}{|\boldsymbol{S}|}\sum_{(x,y)\in\boldsymbol{S}} \mathbf{1}_{h(x)\neq y} | h \in H_n\right] \\ &+ \sum_{n=1}^{N_1}\sum_{i=2}^{\infty} O\left(\left|\frac{E^i[K_S(h)|h\in H_n])}{(\mathcal{E}_n-\alpha)^{2+i}}\right|\right), \end{aligned} \tag{5.21}$$

where

$$c_n = \begin{cases} \min(\beta(\mathcal{E}_n), \frac{\mathcal{E}_n^2 P(H_n)}{(\alpha-\mathcal{E}_n)^3}), & n \leq K_1 \\ -\min(\beta(\mathcal{E}_n), \frac{\mathcal{E}_n^2 P(H_n)}{(\mathcal{E}_n-\alpha)^3}), & n > K_1 \end{cases}, \tag{5.22}$$

Remark 3. *Here we introduce a hard constraint on* c_n*. This is because the upper bound of (5.18) is no longer tight when* $\mathcal{E}_n$ *and* α *are close. Instead, we apply the Hoeffdings inequality [125] to (5.18) and obtain*

$$\begin{aligned} P(R_S(h) \geq \alpha) &= P(R_S(h) - R(h) \geq \alpha - \mathcal{E}_n) \\ &\leq e^{-2|S|(\alpha-\mathcal{E}_n)^2}. \end{aligned} \tag{5.23}$$

This upper bound is independent of the selection of h and S and only dependent on $\mathcal{E}_n$, which could be estimated by $\frac{1}{|D|}\sum_{(x_i,y_i)\in D}\mathbf{1}_{h(x_i)\neq y_i}$.

Since the first term is a constant independent of the selection of S while the remaining higher order term is hard to handle, we can choose to minimize the linear term; i.e.,

$$D^* = \arg\min_{\boldsymbol{S}\subset\boldsymbol{D}_1}\sum_{n=1}^{N_1} c_n E\left[\frac{1}{|\boldsymbol{S}|}\sum_{(x_i,y_i)\in\boldsymbol{S}}\mathbf{1}_{h(x_i)\neq y_i}|h\in H_n\right]. \tag{5.24}$$

Note that the coefficients c_n is positive when $n \leq K_1$ and negative when $n > K_1$. Hence, the selection of samples desires to shrink the classification error for good classifiers while enlarge it for bad classifiers. Another point we notice is that $|c_n|$ is large when $\mathcal{E}_n$ is close to α; however, as $\mathcal{E}_n \in (\frac{\alpha+\mathcal{E}_{min}}{2}, \frac{1+\alpha-\mathcal{E}_{min}}{2})$ is not considered in the proposed algorithm, we avoid handling the case when $c_n \to \infty$.

5.4.2 Algorithm of Sample Selection

The advantage of the new metric is due to the following equivalent form:

$$\begin{aligned}
&\sum_{n=1}^{N_1} c_n E\left[\frac{1}{|\boldsymbol{S}|}\sum_{(x_i,y_i)\in\boldsymbol{S}}\mathbf{1}_{h(x_i)\neq y_i}|h\in H_n\right] \\
=\ &\frac{1}{|\boldsymbol{S}|}\sum_{(x_i,y_i)\in\boldsymbol{S}}\sum_{n=1}^{N_1} c_n E[h(x_i)\neq y_i|h\in H_n] \\
=\ &\frac{1}{|\boldsymbol{S}|}\sum_{(x_i,y_i)\in\boldsymbol{S}} w(x_i,y_i),
\end{aligned} \tag{5.25}$$

where $w(x_i, y_i)$ is the weight of sample (x_i, y_i):

$$w(x,y) = \sum_{n=1}^{N_1} c_n E[h(x)\neq y|h\in H_n]. \tag{5.26}$$

Then, the problem becomes choosing $|\boldsymbol{S}|$ samples in $\boldsymbol{D}_1$ such that the sum of weights corresponding to the selected samples is minimized. This can be achieved by the greedy algorithm, namely simply selecting the samples having first $|\boldsymbol{S}|$ smallest weights.

Note that the weights concern the conditional expectations which are difficult to evaluate, due to the lack of mathematically tractable distributions. We can use empirical processes to approximate the conditional expectation. That is, we randomly choose samples in the classifier space and use the average to approximate $w(x_i, y_i)$ for each $(x_i, y_i) \in D$. The detailed procedure can be found in Algorithm 10.

Algorithm 10 Procedure of Sample Section for Representing Good Classifiers

```
Choose a large number M and randomly select M samples in H.
for Each sample of classifier h do
    Evaluate the classification error E(h, D_1) over the sample set D_1.
    Calculate the coefficient c(h) using (5.22).
end for
for Each sample (x, y) ∈ D_1 do
    Initialize the weight w(x, y) = 0.
    for Each classifier sample h do
        if h(x) ≠ y then
            w(x, y) = w(x, y) + c(h).
        end if
    end for
end for
Sort all samples with the ascending order of their weights {w(x, y)}.
Choose the |S| samples having the smallest weights.
```

Remark 4. *α is used to define what a good classifier is. This is a model specific parameter. The choice of α determines which subset of classifiers could not be considered. In our numerical simulations, α could be 1.3 to 1.5 times of $\mathcal{E}_{min}$ such that the sets of locally good classifiers do have an intersection.*

5.5 Numerical Results

In this section, we provide numerical simulation results to demonstrate the performance of the proposed framework and algorithm.

5.5.1 Synthetic Data

We first test our algorithm on a synthetic data set. Consider two different classes of sets C_1 and C_2 with labels 1 and -1. C_1 and C_2 are equiprobable and consist of random vectors drawn

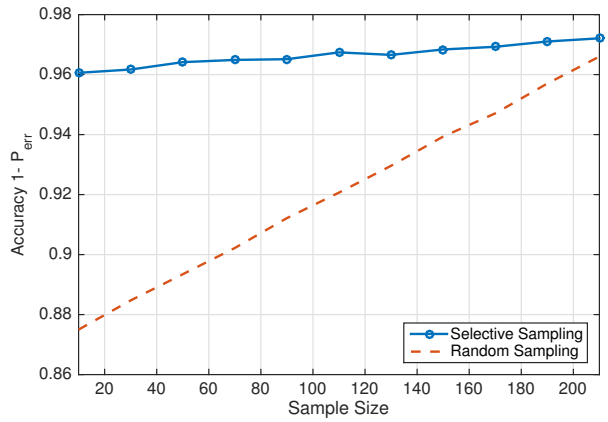


Figure 5.1: Accuracy comparison between the proposed sampling algorithm and random sampling on synthetic data

from a two-dimensional Gaussian distribution with mean vector $m_1 = [0, 0]$, $m_2 = [2, 3]$ and covariance matrix $\Sigma_1 = [3, 0; 0, 4]$ and $\Sigma_2 = [3, 0; 0, 2]$. The total size of the data set $|D|$ is 8000. We implement the proposed algorithm using various sample sizes $|\boldsymbol{S}|$. The linear SVM is considered as the target classifier. The minimum value of $|\boldsymbol{S}|$ is estimated from the VC dimension of the classifier. We calculate the accuracy rate $1 - P_{err}$ and the result is averaged over 30 independent simulations with random generated data set and classifier samples. For comparison, we randomly sample data with the same size and check its accuracy of representing the classifier space.

As displayed in Fig. 5.1, for the synthetic data, the proposed algorithm (Selective Sampling) outperforms the random sampling when the sample size is small. The performance of random sampling becomes similar to the proposed algorithm when the sample size becomes large. Fig. 5.2 illustrates the distribution of the sampled data, from which we observe that most selected samples are close to the boundary and thus provide more critical information for the classification.

Note that, as shown in Table 5.1, even if sample size is small ($|\boldsymbol{S}| = 30$) compared to the total data size $|\boldsymbol{D}_1| = 8000$, we can still use these small amount of samples to describe the set of good classifiers with a high accuracy. Furthermore, the optimal classifier found from the selected samples achieves a performance close to the globally optimal one. This implies that a simple strategy for distributed nodes to learn a globally optimal classifier is to collect

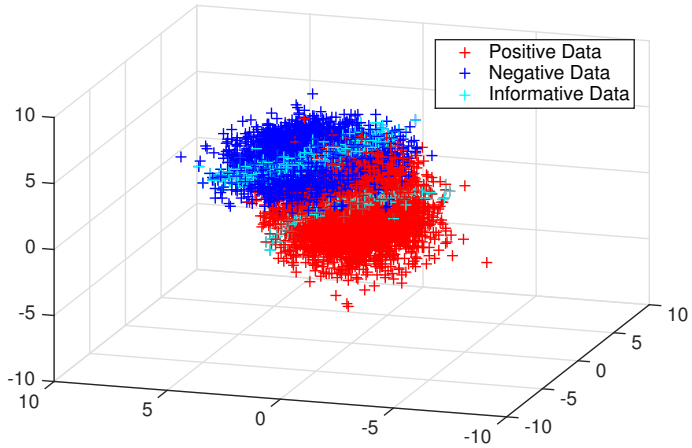


Figure 5.2: Illustration of informative data close to the boundary

Table 5.1: Learning result for the synthetic data

	Selective Sampling		Random Sampling	
	$R_S(h)$	$R(h)$	$R_S(h)$	$R(h)$
Synthetic	0.1 $\pm$0.017	0.0913 $\pm$0.015	0.1 $\pm$0.021	0.0935$\pm$0.017
Magic	0.23 $\pm$0.034	0.21 $\pm$0.031	0.22 $\pm$0.043	0.20 $\pm$0.041

a small number of samples to a center in a single communication round and let the center learn using these samples.

5.5.2 Real World Data

Here the proposed algorithm is tested on the MAGIC Gamma Telescope Data Set from UCI Machine Learning Repository [13]. The data is generated to simulate the registration of high energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope using the imaging technique. There are 10 attributes describing the image of each sample and one label indicating whether it is signal or background noise. The overall data size $|\boldsymbol{D}_1|$ is 19020. For the convenience of generating sample classifiers, we normalize the feature of each sample.

Fig 5.3 shows the performance comparison between the proposed approach and random sampling, along with two other baseline approaches, namely the distributed boosting [34] and HT estimator [75]. The proposed sampling algorithm outperforms these baseline approaches as expected. The reasons include

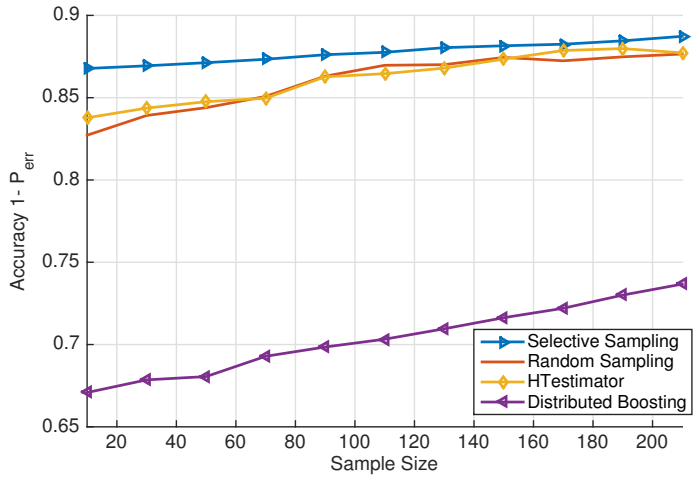


Figure 5.3: Accuracy comparison on real world data

- The distributed boosting selects the samples that are misclassified by most good classifiers with high probability. In this case, $P(h \in \mathcal{H}_0, h \in \hat{\mathcal{H}}_1(S))$ is large since most classifiers have bad performance on these noisy samples while some of them could have globally good performance.

- In the HT estimator, the sampling algorithm is proposed to minimize the geometric distance of $R_S(h)$ and $R(h)$, where the target function could be described as

$$\min_{\boldsymbol{S} \in \boldsymbol{D}} \sum_{h \in H} (R_S(h) - R(h))^2, \tag{5.27}$$

  with some other constraints on $\boldsymbol{S}$. When the sample size is small, the resolution of error rate is low and even the difference between $R_S(h)$ and $R(h)$ is minimized. It is possible that they fall in different categories. Besides, (5.27) requires solving a quadratic optimization problem, which requires substantial computation cost compared to the sampling algorithm proposed in this paper.

5.5.3 Computational Cost

The proposed algorithm is robust on the simple synthetic data where the dimension of each sample is small. However, when applying it to data set with high dimensions, there are some practical issues degrading its efficiency and accuracy. With the increasing dimension,

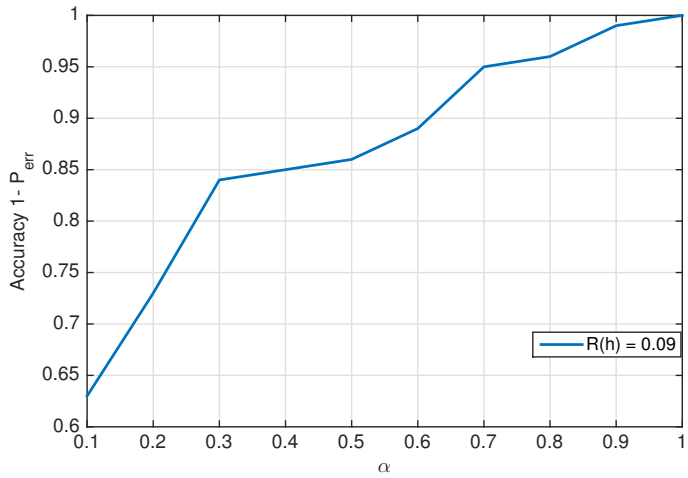


Figure 5.4: Influence of threshold α on the learning performance

it becomes more difficult to generate sufficient classifiers since a large number of h_n's are required to describe the classifier space H. An alternative solution is to apply the dimension reduction algorithm such as Principle Component Analysis (PCA) to find the main feature of data before learning.

5.5.4 Selection of α

As for the only tuning parameter in the proposed framework, we claim that with the moderate assumption that α is not close to $R(h_{opt})$, the learning result is not controlled by the selection of α. As shown in Fig. 5.4, when $R(h_{opt}) = 0.09$, the accuracy is satisfying once α is larger than 0.2.

5.5.5 Performance of Classification of Classifiers

For further analysis, we demonstrate the performance of the proposed algorithm on the real world data with respect to the ingredient of $P_{err}(S)$. In Fig. 5.5, the *Type 1* error indicates the number of good classifiers that perform bad on the selected sample. Meanwhile, the *Type 2* error indicates the number of bad classifiers have poor local performance. We can observe from the figure that, when α is small, most errors are of *Type 1* as a result of high threshold for good classifiers. With the increase of α, the number of *Type 1* errors decreases.

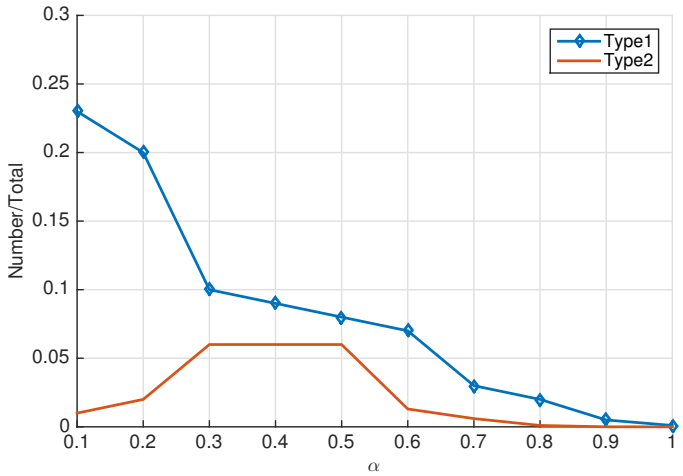


Figure 5.5: Analysis of $P_{err}(S)$ with respect to different α's.

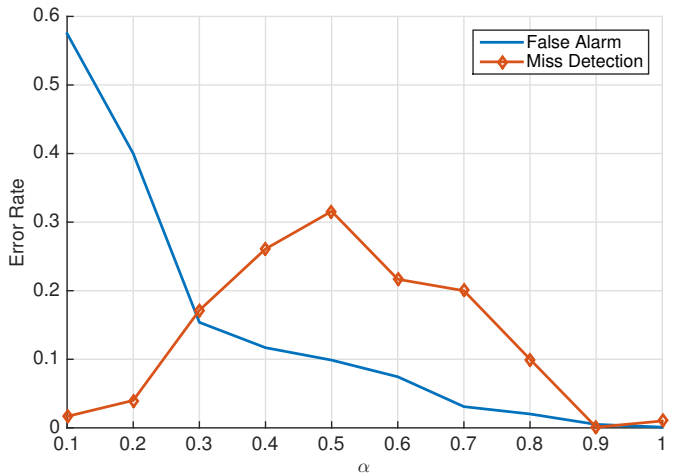


Figure 5.6: False alarm and missed detection rates

Meanwhile, the *Type 2* error rate is always small compared to the *Type 1* error, and achieves its maximum when the threshold α is moderate. The false alarm and miss detection rates of the proposed algorithm are shown in Fig 5.6, Since $\alpha \geq 0.5$ provides little information about the connection between the sample space and the classifier space, we conclude that 0.3 is a good choice for α.

5.6 Conclusion

The paper has proposed a sample selection based source coding scheme in the context of distributed learning. An efficient sampling algorithm has been implemented to optimize the upper bound of the proposed metric. The simulation results have indicated its advantage over random sampling and other related sampling methods, either in accuracy or efficiency. With the reduced sample size, the communication and computation cost is reduced with slight degradation of the machine learning performance.

Chapter 6

Open Problems and Future Work

The goal of this chapter is to discuss advantages and weaknesses of the algorithms developed in this dissertation and highlight several promising avenues for future research.

Reinforcement learning in resource allocation

Our proposed framework solved the user association and power control in HetNets jointly by applying the Q learning approach. To acchieve the satisfying running time performance, we need the prior knowledge about the channel dynamics. When the channel dynamics is fixed over time, the proposed is efficient since the training process is required for once. However, if the channle dynamics changes over time, the proposed framework requires the system to train a new Q table, which is not practical. Besides, we assume that the number of users in the cell is fixed. This assumption is not valid for practical system as the number of users in the cell could change frequently. In the future, it is possible to introduce the learning Q learning approach to handle the problem and consider the number of the user as the system state. With the development of modern machine learning, there are more efficient deep learning frameworks and the computation efficiency would no longer be the contraint for the problem that requiring low latency.

Communication efficient distributed learning The proposed coreset construction algorithm is scalable to the size of the data set. However, since the computation complexity of coreset construntion is linear to the dimension of the data set, it is hard to apply the proposed algorithm to the modern machine learning problem, such as language processing or image classification proble,]m, where the dimension is high. It is worthy of further investigation on

the design of computational efficient algorithm that is scalable to the dimension of the data set.

Bibliography

[1] Agarwal, A. and Duchi, J. C. (2011). Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 873–881. 16

[2] Agarwal, P. K., Har-Peled, S., and Varadarajan, K. R. (2005). Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30. 66

[3] Aji, A. F. and Heafield, K. (2017). Sparse communication for distributed gradient descent. *arXiv preprint arXiv:1704.05021*. 17

[4] Akdeniz, M. R., Liu, Y., Samimi, M. K., Sun, S., Rangan, S., Rappaport, T. S., and Erkip, E. (2014). Millimeter wave channel modeling and cellular capacity evaluation. *IEEE Journal on Selected Areas in Communications*, 32(6):1164–1179. 36

[5] Al-Rawi, H. A., Ng, M. A., and Yau, K.-L. A. (2015). Application of reinforcement learning to routing in distributed wireless networks: a review. *Artificial Intelligence Review*, 43(3):381–416. 2

[6] Alliance, N. (2012). Small cell backhaul requirements. *white paper, June*. 24

[7] Amari, S. et al. (1998). Statistical inference under multiterminal data compression. *IEEE Transactions on Information Theory*, 44(6):2300–2324. 66

[8] Andrews, J. G. (2013). Seven ways that hetnets are a cellular paradigm shift. *IEEE Communications Magazine*, 51(3):136–144. 5

[9] Andrews, J. G., Singh, S., Ye, Q., Lin, X., and Dhillon, H. S. (2014). An overview of load balancing in hetnets: Old myths and open problems. *IEEE Wireless Communications*, 21(2):18–25. 5

[10] Arora, S. and Barak, B. (2009). *Computational complexity: a modern approach*. Cambridge University Press. 77

[11] Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics. 72

[12] Aryafar, E., Keshavarz-Haddad, A., Wang, M., and Chiang, M. (2013). Rat selection games in hetnets. In *INFOCOM, 2013 Proceedings IEEE*, pages 998–1006. IEEE. 9

[13] Asuncion, A. and Newman, D. (2007). Uci machine learning repository. 111

[14] Athanasiou, G., Weeraddana, P. C., and Fischione, C. (2013). Auction-based resource allocation in millimeterwave wireless access networks. *IEEE Communications Letters*, 17(11):2108–2111. 7

[15] Bachem, O., Lucic, M., and Krause, A. (2015). Coresets for nonparametric estimation-the case of dp-means. In *International Conference on Machine Learning*, pages 209–217. 67

[16] Bai, T. and Heath, R. W. (2015). Coverage and rate analysis for millimeter-wave cellular networks. *IEEE Transactions on Wireless Communications*, 14(2):1100–1114. 10

[17] Balcan, M.-F., Blum, A., Fine, S., and Mansour, Y. (2012). Unsupervised svms: On the complexity of the furthest hyperplane problem. In *COLT 2012 - The 25th Annual Conference on Learning Theory*, pages 26.1–26.22. 19, 66, 73, 99

[18] Balcan, M.-F. F., Ehrlich, S., and Liang, Y. (2013). Distributed k-means and k-median clustering on general topologies. In *Advances in Neural Information Processing Systems*, pages 1995–2003. 78

[19] Bao, W. and Liang, B. (2014). Structured spectrum allocation and user association in heterogeneous cellular networks. In *INFOCOM, 2014 Proceedings IEEE*, pages 1069–1077. IEEE. 11

[20] Bartlett, P. L. and Mendelson, S. (2002). Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482. 14

[21] Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202. 14

[22] Bellet, A., Liang, Y., Garakani, A. B., Balcan, M.-F., and Sha, F. (2015). A distributed frank-wolfe algorithm for communication-efficient sparse learning. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 478–486. SIAM. 18, 97

[23] Ben-Nun, T. and Hoefler, T. (2018). Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. *arXiv preprint arXiv:1802.09941*. 18

[24] Bethanabhotla, D., Bursalioglu, O. Y., Papadopoulos, H. C., and Caire, G. (2016). Optimal user-cell association for massive mimo wireless networks. *IEEE Transactions on Wireless Communications*, 15(3):1835–1850. 9

[25] Bose, J. C. et al. (1927). *Collected physical papers*. Longmans, Green and Co., London. 24

[26] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer. 14, 100

[27] Boutsidis, C., Drineas, P., and Magdon-Ismail, M. (2013). Near-optimal coresets for least-squares regression. *IEEE transactions on information theory*, 59(10):6880–6892. 20

[28] Braverman, V., Feldman, D., and Lang, H. (2016). New frameworks for offline and streaming coreset constructions. *arXiv preprint arXiv:1612.00889*. 67, 70

[29] Canini, K., Chandra, T., Ie, E., McFadden, J., Goldman, K., Gunter, M., Harmsen, J., LeFevre, K., Lepikhin, D., Llinares, T., et al. (2012). Sibyl: A system for large scale supervised machine learning. *Technical Talk*. 3

[30] Chen, C.-Y., Chen, Y.-Y., and Wei, H.-Y. (2016). Multi-cell interference coordinated scheduling in mmwave 5g cellular systems. In *Ubiquitous and Future Networks (ICUFN), 2016 Eighth International Conference on*, pages 912–917. IEEE. 33

[31] Chen, C.-Y., Choi, J., Brand, D., Agrawal, A., Zhang, W., and Gopalakrishnan, K. (2017). Adacomp: Adaptive residual gradient compression for data-parallel distributed training. *arXiv preprint arXiv:1712.02679*. 17

[32] Chen, L. and Li, H. (2016). An mdp-based vertical handoff decision algorithm for heterogeneous wireless networks. In *Wireless Communications and Networking Conference (WCNC), 2016 IEEE*, pages 1–6. IEEE. 13

[33] Chen, L., Wang, B., Chen, X., Zhang, X., and Yang, D. (2011). Utility-based resource allocation for mixed traffic in wireless networks. In *Computer communications workshops (INFOCOM WKSHPS), 2011 IEEE conference on*, pages 91–96. IEEE. 6

[34] Chen, S.-T., Balcan, M.-F., and Chau, D. H. (2015). Communication efficient distributed agnostic boosting. *arXiv preprint arXiv:1506.06318*. 66, 82, 99, 111

[35] Chu, W., Park, S.-T., Beaupre, T., Motgi, N., Phadke, A., Chakraborty, S., and Zachariah, J. (2009). A case study of behavior-driven conjoint analysis on yahoo!: front page today module. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1097–1104. ACM. 81

[36] Corroy, S., Falconetti, L., and Mathar, R. (2012). Dynamic cell association for downlink sum rate maximization in multi-cell heterogeneous networks. In *Communications (ICC), 2012 IEEE International Conference on*, pages 2457–2461. IEEE. 7

[37] Coucheney, P., Hyon, E., and Kelif, J.-M. (2013). Mobile association problem in heterogenous wireless networks with mobility. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, pages 3129–3133. IEEE. 8

[38] Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., et al. (2012). Large scale distributed deep networks. In *Advances in neural information processing systems*, pages 1223–1231. 3

[39] Dhillon, H. S., Ganti, R. K., Baccelli, F., and Andrews, J. G. (2012). Modeling and analysis of k-tier downlink heterogeneous cellular networks. *IEEE Journal on Selected Areas in Communications*, 30(3):550–560. 10

[40] Di Renzo, M. (2015). Stochastic geometry modeling and analysis of multi-tier millimeter wave cellular networks. *IEEE Transactions on Wireless Communications*, 14(9):5038–5057. 33

[41] Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157. 73

[42] Docomo, N. (2010). Performance of eicic with control channel coverage limitation. *R1-103264, 3GPP Std., Montreal, Canada*. 5, 27

[43] Domingo, C. and Watanabe, O. (2000). Madaboost: A modification of adaboost. In *COLT*, pages 180–189. 73

[44] Duchi, J. C., Agarwal, A., and Wainwright, M. J. (2012). Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3):592–606. 66

[45] Duchi, J. C., Jordan, M. I., Wainwright, M. J., and Zhang, Y. (2014). Optimality guarantees for distributed statistical estimation. *arXiv preprint arXiv:1405.0782*. 16

[46] Dundar, M., Krishnapuram, B., Bi, J., and Rao, R. B. (2007). Learning classifiers when the training data is not iid. In *IJCAI*, pages 756–761. 66

[47] El-Alfy, E.-S., Yao, Y.-D., and Heffes, H. (2006). A learning approach for prioritized handoff channel allocation in mobile multimedia networks. *IEEE transactions on wireless communications*, 5(7):1651–1660. 13

[48] Elayoubi, S. E., Altman, E., Haddad, M., and Altman, Z. (2010). A hybrid decision approach for the association problem in heterogeneous networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–5. IEEE. 13

[49] ElSawy, H., Hossain, E., and Haenggi, M. (2013). Stochastic geometry for modeling, analysis, and design of multi-tier and cognitive cellular wireless networks: A survey. *IEEE Communications Surveys & Tutorials*, 15(3):996–1019. 10

[50] Feldman, D., Faulkner, M., and Krause, A. (2011). Scalable training of mixture models via coresets. In *Advances in Neural Information Processing Systems*, pages 2142–2150. 77

[51] Feldman, D. and Langberg, M. (2011). A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 569–578. ACM. 70, 90

[52] Feldman, D., Schmidt, M., and Sohler, C. (2013). Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1434–1453. SIAM. 67

[53] Feng, M., Jiang, T., Chen, D., and Mao, S. (2014). Cooperative small cell networks: High capacity for hotspots with interference mitigation. *IEEE Wireless Communications*, 21(6):108–116. 7

[54] Feng, W., Wang, Y., Lin, D., Ge, N., Lu, J., and Li, S. (2017). When mmwave communications meet network densification: A scalable interference coordination perspective. *IEEE Journal on Selected Areas in Communications*. 33

[55] Ford, R., Rangan, S., Mellios, E., Kong, D., and Nix, A. (2017). Markov channel-based performance analysis for millimeter wave mobile networks. In *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*, pages 1–6. IEEE. 53

[56] Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139. 28

[57] Garg, R. and Khandekar, R. (2009). Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 337–344. ACM. 17

[58] Ge, R., Huang, F., Jin, C., and Yuan, Y. (2015). Escaping from saddle pointsonline stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842. 14

[59] Ghosh, A., Mangalvedhe, N., Ratasuk, R., Mondal, B., Cudak, M., Visotsky, E., Thomas, T. A., Andrews, J. G., Xia, P., Jo, H. S., et al. (2012). Heterogeneous cellular networks: From theory to practice. *IEEE communications magazine*, 50(6). 5

[60] Goyal, S., Mezzavilla, M., Rangan, S., Panwar, S., and Zorzi, M. (2017). User association in 5g mmwave networks. In *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*, pages 1–6. IEEE. 33

[61] Guestrin, C., Koller, D., and Parr, R. (2001). Multiagent planning with factored mdps. In *NIPS*, volume 1, pages 1523–1530. 43

[62] Guestrin, C., Lagoudakis, M., and Parr, R. (2002). Coordinated reinforcement learning. In *ICML*, volume 2, pages 227–234. 43

[63] Ha, V. N. and Le, L. B. (2014). Distributed base station association and power control for heterogeneous cellular networks. *IEEE Transactions on Vehicular Technology*, 63(1):282–296. 9

[64] Hansryd, J., Edstam, J., Olsson, B.-E., and Larsson, C. (2013). Non-line-of-sight microwave backhaul for small cells. *Ericsson Review*, 3:2–8. 24

[65] Hardt, M., Recht, B., and Singer, Y. (2016). Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234. 14

[66] Harsanyi, J. C., Selten, R., et al. (1988). A general theory of equilibrium selection in games. *MIT Press Books*, 1. 9

[67] Hashemi, S. H., Jyothi, S. A., and Campbell, R. H. (2018). Communication scheduling as a first-class citizen in distributed machine learning systems. *arXiv preprint arXiv:1803.03288*. 18

[68] Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30. 73

[69] Hsieh, K., Harlap, A., Vijaykumar, N., Konomis, D., Ganger, G. R., Gibbons, P. B., and Mutlu, O. (2017). Gaia: Geo-distributed machine learning approaching lan speeds. In *NSDI*, pages 629–647. 21

[70] Huggins, J., Campbell, T., and Broderick, T. (2016). Coresets for scalable bayesian logistic regression. In *Advances In Neural Information Processing Systems*, pages 4080–4088. 67, 72, 94

[71] Ihler, A. T., John III, W. F., and Willsky, A. S. (2005). Loopy belief propagation: Convergence and effects of message errors. *Journal of Machine Learning Research*, 6(May):905–936. 52

[72] Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. (2017). How to escape saddle points efficiently. In *International Conference on Machine Learning*, pages 1724–1732. 14

[73] Jo, H.-S., Sang, Y. J., Xia, P., and Andrews, J. G. (2012). Heterogeneous cellular networks with flexible cell association: A comprehensive downlink sinr analysis. *IEEE Transactions on Wireless Communications*, 11(10):3484–3495. 11

[74] Johansson, B., Rabi, M., and Johansson, M. (2009). A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20(3):1157–1170. 17

[75] K. Lang, E. L. and Shmakov, K. (2016). Stratified sampling meets machine learning. *Preprint*. 99, 111

[76] Kanade, V. and Kalai, A. (2009). Potential-based agnostic boosting. In *Advances in Neural Information Processing Systems*, pages 880–888. 77

[77] Kearns, M. J. and Vazirani, U. V. (1994). *An introduction to computational learning theory*. MIT press. 99

[78] Kok, J. R. and Vlassis, N. (2006). Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research*, 7(Sep):1789–1828. 43

[79] Langberg, M. and Schulman, L. J. (2010). Universal ε-approximators for integrals. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 598–607. SIAM. 31

[80] Lasaulce, S. and Tembine, H. (2011). *Game theory and learning for wireless networks: fundamentals and applications*. Academic Press. 9

[81] Lee, K., Lam, M., Pedarsani, R., Papailiopoulos, D., and Ramchandran, K. (2018). Speeding up distributed machine learning using codes. *IEEE Transactions on Information Theory*, 64(3):1514–1529. 1

[82] Levorato, M., Firouzabadi, S., and Goldsmith, A. (2012). A learning framework for cognitive interference networks with partial and noisy observations. *IEEE Transactions on Wireless Communications*, 11(9):3101–3111. 1, 13

[83] Li, M., Andersen, D. G., Park, J. W., Smola, A. J., Ahmed, A., Josifovski, V., Long, J., Shekita, E. J., and Su, B.-Y. (2014a). Scaling distributed machine learning with the parameter server. In *OSDI*, volume 14, pages 583–598. 16, 21

[84] Li, M., Andersen, D. G., Smola, A. J., and Yu, K. (2014b). Communication efficient distributed machine learning with the parameter server. In *Advances in Neural Information Processing Systems*, pages 19–27. 21

[85] Li, M., Zhou, L., Yang, Z., Li, A., Xia, F., Andersen, D. G., and Smola, A. (2013). Parameter server for distributed machine learning. In *Big Learning NIPS Workshop*, volume 1. 97

[86] Li, R., Zhao, Z., Chen, X., Palicot, J., and Zhang, H. (2014c). Tact: A transfer actor-critic learning framework for energy saving in cellular radio access networks. *IEEE transactions on wireless communications*, 13(4):2000–2011. 2

[87] Li, W. and Todorov, E. (2004). Iterative linear quadratic regulator design for nonlinear biological movement systems. In *ICINCO (1)*, pages 222–229. 53

[88] Li, Y., Ji, H., Li, X., and Leung, V. (2012). Dynamic channel selection with reinforcement learning for cognitive wlan over fiber. *International Journal of Communication Systems*, 25(8):1077–1090. 34

[89] Liu, D., Wang, L., Chen, Y., Elkashlan, M., Wong, K.-K., Schober, R., and Hanzo, L. (2016). User association in 5g networks: A survey and an outlook. *IEEE Communications Surveys & Tutorials*, 18(2):1018–1044. 8, 33

[90] Liu, S. and Du, J. (2016). Poster: Mobiear-building an environment-independent acoustic sensing platform for the deaf using deep learning. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services Companion*, pages 50–50. ACM. 2

[91] Liu, Y.-J., Tang, L., Tong, S., Chen, C. P., and Li, D.-J. (2015). Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time mimo systems. *IEEE Transactions on Neural Networks and Learning Systems*, 26(1):165–176. 2

[92] Lo, B. F. and Akyildiz, I. F. (2010). Reinforcement learning-based cooperative sensing in cognitive radio ad hoc networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*, pages 2244–2249. IEEE. 34, 46

[93] Lopez-Perez, D., Guvenc, I., De la Roche, G., Kountouris, M., Quek, T. Q., and Zhang, J. (2011). Enhanced intercell interference coordination challenges in heterogeneous networks. *IEEE Wireless Communications*, 18(3). 27

[94] Lu, J. S., Steinbach, D., Cabrol, P., and Pietraski, P. (2012). Modeling human blockers in millimeter wave radio links. *ZTE Communications*, 10(4):23–28. 26

[95] Lu, W. and Di Renzo, M. (2015). Stochastic geometry modeling of cellular networks: Analysis, simulation and experimental validation. In *Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 179–188. ACM. 36

[96] Luby, M. G., Mitzenmacher, M., Shokrollahi, M. A., and Spielman, D. A. (2001). Improved low-density parity-check codes using irregular graphs. *IEEE Transactions on Information Theory*, 47(2):585–598. 50

[97] Lucic, M., Bachem, O., and Krause, A. (2015). Strong coresets for hard and soft bregman clustering with applications to exponential family mixtures. *arXiv preprint arXiv:1508.05243*. 20

[98] Lucic, M., Faulkner, M., Krause, A., and Feldman, D. (2017). Training mixture models at scale via coresets. *arXiv preprint arXiv:1703.08110*. 20

[99] Lundén, J., Kulkarni, S. R., Koivunen, V., and Poor, H. V. (2013). Multiagent reinforcement learning based spectrum sensing policies for cognitive radio networks. *IEEE Journal of Selected Topics in Signal Processing*, 7(5):858–868. 34

[100] Madan, R., Borran, J., Sampath, A., Bhushan, N., Khandekar, A., and Ji, T. (2010). Cell association and interference coordination in heterogeneous lte-a cellular networks. *IEEE Journal on selected areas in communications*, 28(9):1479–1489. 7

[101] Mastronarde, N. and van der Schaar, M. (2011). Fast reinforcement learning for energy-efficient wireless communication. *IEEE Transactions on Signal Processing*, 59(12):6262–6266. 34

[102] Mezzavilla, M., Goyal, S., Panwar, S., Rangan, S., and Zorzi, M. (2016). An mdp model for optimal handover decisions in mmwave cellular networks. In *Networks and Communications (EuCNC), 2016 European Conference on*, pages 100–105. IEEE. 34

[103] Nath, A. and Domingos, P. M. (2010). Efficient belief propagation for utility maximization and repeated inference. In *AAAI*, volume 4, page 3. 53

[104] Nedic, A. and Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61. 17

[105] Neelakantan, A., Le, Q. V., and Sutskever, I. (2015). Neural programmer: Inducing latent programs with gradient descent. *arXiv preprint arXiv:1511.04834*. 14

[106] Nesterov, Y. (2013a). Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161. 14

[107] Nesterov, Y. (2013b). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media. 14

[108] Niyato, D. and Hossain, E. (2009). Dynamics of network selection in heterogeneous wireless networks: An evolutionary game approach. *IEEE transactions on vehicular technology*, 58(4):2008–2017. 9

[109] Ogawa, K., Suzuki, Y., and Takeuchi, I. (2013). Safe screening of non-support vectors in pathwise svm computation. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1382–1390. 99

[110] Pi, Z. and Khan, F. (2011). An introduction to millimeter-wave mobile broadband systems. *IEEE communications magazine*, 49(6). 25, 26

[111] R. Bekkerman, M. B. and Langford, J. (2012). *Scaling Up Machine Learning: Parallel and Distributed Approaches*. Cambridge University Press. 97

[112] Ram, S. S., Nedić, A., and Veeravalli, V. V. (2010). Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of optimization theory and applications*, 147(3):516–545. 16

[113] Rangan, S., Rappaport, T. S., and Erkip, E. (2014). Millimeter-wave cellular wireless networks: Potentials and challenges. *Proceedings of the IEEE*, 102(3):366–385. 6, 24, 32

[114] Rao, J. B. and Fapojuwo, A. O. (2014). A survey of energy efficient resource management techniques for multicell cellular networks. *IEEE Communications Surveys & Tutorials*, 16(1):154–180. 5

[115] Rappaport, T. S. et al. (1996). *Wireless communications: principles and practice*, volume 2. prentice hall PTR New Jersey. 26

[116] Rappaport, T. S., Sun, S., Mayzus, R., Zhao, H., Azar, Y., Wang, K., Wong, G. N., Schulz, J. K., Samimi, M., and Gutierrez, F. (2013). Millimeter wave mobile communications for 5g cellular: It will work! *IEEE access*, 1:335–349. 24, 33, 46

[117] Recht, B., Re, C., Wright, S., and Niu, F. (2011). Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in neural information processing systems*, pages 693–701. 16

[118] Reddi, S. J., Póczos, B., and Smola, A. J. (2015). Communication efficient coresets for empirical loss minimization. In *UAI*, pages 752–761. 20

[119] Roddy, D. (2006). *Satellite Communications, (Professional Engineering)*. McGraw-Hill Professional: New York. 24

[120] Sadr, S. and Adve, R. S. (2014). Tier association probability and spectrum partitioning for maximum rate coverage in multi-tier heterogeneous networks. *IEEE Communications Letters*, 18(10):1791–1794. 11

[121] Sadr, S. and Adve, R. S. (2015). Handoff rate and coverage analysis in multi-tier heterogeneous networks. *IEEE Transactions on Wireless Communications*, 14(5):2626–2638. 8

[122] Schapire, R. E. and Freund, Y. (2012). *Boosting: Foundations and algorithms*. MIT press. 72, 73

[123] Schmidt, M., Roux, N. L., and Bach, F. R. (2011). Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in neural information processing systems*, pages 1458–1466. 14

[124] Semiari, O., Saad, W., Valentin, S., Bennis, M., and Maham, B. (2014). Matching theory for priority-based cell association in the downlink of wireless small cell networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 444–448. IEEE. 9

[125] Serfling, R. J. (1974). Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, pages 39–48. 107

[126] Shalev-Shwartz, S. and Zhang, T. (2013). Accelerated mini-batch stochastic dual coordinate ascent. In *Advances in Neural Information Processing Systems*, pages 378–385. 18

[127] Shamir, O., Srebro, N., and Zhang, T. (2014). Communication-efficient distributed optimization using an approximate newton-type method. In *International conference on machine learning*, pages 1000–1008. 18

[128] Shen, K. and Yu, W. (2014). Distributed pricing-based user association for downlink heterogeneous cellular networks. *IEEE Journal on Selected Areas in Communications*, 32(6):1100–1113. 6

[129] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489. 13

[130] Simsek, M., Bennis, M., and Czylwik, A. (2012). Dynamic inter-cell interference coordination in hetnets: A reinforcement learning approach. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 5446–5450. IEEE. 34

[131] Singh, S. and Andrews, J. G. (2014). Joint resource partitioning and offloading in heterogeneous cellular networks. *IEEE Transactions on Wireless Communications*, 13(2):888–901. 11

[132] Singh, S., Dhillon, H. S., and Andrews, J. G. (2013). Offloading in heterogeneous networks: Modeling, analysis, and design insights. *IEEE Transactions on Wireless Communications*, 12(5):2484–2497. 10

[133] Singh, S., Kulkarni, M. N., Ghosh, A., and Andrews, J. G. (2015). Tractable model for rate in self-backhauled millimeter wave cellular networks. *IEEE Journal on Selected Areas in Communications*, 33(10):2196–2211. 10

[134] Stephen, R. G. and Zhang, R. (2017). Joint millimeter-wave fronthaul and ofdma resource allocation in ultra-dense cran. *IEEE Transactions on Communications*, 65(3):1411–1423. 33

[135] Stevens-Navarro, E., Lin, Y., and Wong, V. W. (2008). An mdp-based vertical handoff decision algorithm for heterogeneous wireless networks. *IEEE Transactions on Vehicular Technology*, 57(2):1243–1254. 13

[136] Su, L., Yang, C., Xu, Z., and Molisch, A. F. (2013). Energy-efficient downlink transmission with base station closing in small cell networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 4784–4788. IEEE. 8

[137] Su, X., Zhang, D., Li, W., and Zhao, K. (2016). A deep learning approach to android malware feature learning and detection. In *Trustcom/BigDataSE/I SPA, 2016 IEEE*, pages 244–251. IEEE. 2

[138] Sudderth, E. B., Ihler, A. T., Isard, M., Freeman, W. T., and Willsky, A. S. (2010). Nonparametric belief propagation. *Communications of the ACM*, 53(10):95–103. 50

[139] Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the seventh international conference on machine learning*, pages 216–224. 53, 54

[140] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge. 54

[141] Tam, K., Feizollah, A., Anuar, N. B., Salleh, R., and Cavallaro, L. (2017). The evolution of android malware and android analysis techniques. *ACM Computing Surveys (CSUR)*, 49(4):76. 2

[142] Tishby, N. and Zaslavsky, N. (2015). Deep learning and the information bottleneck principle. In *Information Theory Workshop (ITW), 2015 IEEE*, pages 1–5. IEEE. 1

[143] Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media. 14

[144] Vapnik, V. N. and Vapnik, V. (1998). *Statistical learning theory*, volume 1. Wiley New York. 101

[145] Wainwright, M., Jaakkola, T., and Willsky, A. (2004). Tree consistency and bounds on the performance of the max-product algorithm and its generalizations. *Statistics and computing*, 14(2):143–166. 52

[146] Wang, L. and Kuo, G.-S. G. (2013). Mathematical modeling for network selection in heterogeneous wireless networksa tutorial. *IEEE Communications Surveys & Tutorials*, 15(1):271–292. 6

[147] Wang, M., Dutta, A., Buccapatnam, S., and Chiang, M. (2016). Smart exploration in hetnets: Minimizing total regret with mmwave. In *Proc. IEEE International Conference on Sensing, Communication and Networking*. 33

[148] Wangni, J., Wang, J., Liu, J., and Zhang, T. (2017). Gradient sparsification for communication-efficient distributed optimization. *arXiv preprint arXiv:1710.09854*. 17

[149] Warmuth, M. K., Glocer, K. A., and Rätsch, G. (2007). Boosting algorithms for maximizing the soft margin. In *Advances in Neural Information Processing Systems*, pages 1585–1592. 74

[150] Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4):279–292. 39

[151] Wen, W., Xu, C., Yan, F., Wu, C., Wang, Y., Chen, Y., and Li, H. (2017). Terngrad: Ternary gradients to reduce communication in distributed deep learning. In *Advances in Neural Information Processing Systems*, pages 1508–1518. 17

[152] Woodruff, D. P. and Zhang, Q. (2017). When distributed computation is communication expensive. *Distributed Computing*, 30(5):309–323. 19

[153] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*. 3

[154] Xiao, L. (2010). Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596. 14

[155] Xu, B., Chen, Y., Elkashlan, M., Zhang, T., and Wong, K.-K. (2016a). User association in massive mimo and mmwave enabled hetnets powered by renewable energy. In *Wireless Communications and Networking Conference (WCNC), 2016 IEEE*, pages 1–6. IEEE. 7

[156] Xu, Y., Athanasiou, G., Fischione, C., and Tassiulas, L. (2016b). Distributed association control and relaying in millimeter wave wireless networks. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–6. IEEE. 7

[157] Xu, Y., Hu, R. Q., Wei, L., and Wu, G. (2014). Qoe-aware mobile association and resource allocation over wireless heterogeneous networks. In *Global Communications Conference (GLOBECOM), 2014 IEEE*, pages 4695–4701. IEEE. 6

[158] Xu, Y. and Mao, S. (2017). User association in massive mimo hetnets. *IEEE Systems Journal*, 11(1):7–19. 9

[159] Yau, K.-L. A., Komisarczuk, P., and Teal, P. D. (2012). Reinforcement learning for context awareness and intelligence in wireless networks: Review, new features and open issues. *Journal of Network and Computer Applications*, 35(1):253–267. 13, 33, 34

[160] Ye, M. and Abbe, E. (2018). Communication-computation efficient gradient coding. *arXiv preprint arXiv:1802.03475*. 17

[161] Ye, Q., Rong, B., Chen, Y., Al-Shalash, M., Caramanis, C., and Andrews, J. G. (2013). User association for load balancing in heterogeneous cellular networks. *IEEE Transactions on Wireless Communications*, 12(6):2706–2716. 5, 33, 39

[162] Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2003). Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239. 16, 57

[163] Yi, Y., Zhang, J., Zhang, Q., and Jiang, T. (2011). Spectrum leasing to multiple cooperating secondary cellular networks. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE. 6

[164] Yin, D., Chen, Y., Ramchandran, K., and Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. *arXiv preprint arXiv:1803.01498*. 19

[165] Zhang, C., Guo, D., and Fan, P. (2016). Tracking angles of departure and arrival in a mobile millimeter wave channel. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–6. IEEE. 37

[166] Zhang, H., Huang, S., Jiang, C., Long, K., Leung, V. C., and Poor, H. V. (2017a). Energy efficient user association and power allocation in millimeter-wave-based ultra dense networks with energy harvesting base stations. *IEEE Journal on Selected Areas in Communications*, 35(9):1936–1947. 61

[167] Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 116. ACM. 14

[168] Zhang, Y., Duchi, J., Jordan, M. I., and Wainwright, M. J. (2013). Information-theoretic lower bounds for distributed statistical estimation with communication constraints. In *Advances in Neural Information Processing Systems*, pages 2328–2336. 16, 66, 97

[169] Zhang, Y. and Jordan, M. I. (2015). Splash: User-friendly programming interface for parallelizing stochastic algorithms. *arXiv preprint arXiv:1506.07552*. 22

[170] Zhang, Y., Liang, P., and Charikar, M. (2017b). A hitting time analysis of stochastic gradient langevin dynamics. *arXiv preprint arXiv:1702.05575*. 1, 14

[171] Zhang, Y. and Lin, X. (2015). Disco: Distributed optimization for self-concordant empirical loss. In *International conference on machine learning*, pages 362–370. 18

[172] Zhang, Y., Wainwright, M. J., and Duchi, J. C. (2012). Communication-efficient algorithms for statistical optimization. In *Advances in Neural Information Processing Systems*, pages 1502–1510. 66

[173] Zhou, Z.-H. (2014). Boosting 25 years. CCL. 28

[174] Zhou, Z.-H., Sun, Y.-Y., and Li, Y.-F. (2009). Multi-instance learning by treating instances as non-iid samples. In *Proceedings of the 26th annual international conference on machine learning*, pages 1249–1256. ACM. 66

Vita

Yawen Fan was born on October 15, 1990. He graduated from Fudan University, China in June 2013, with a B.S. in Electrical Engineering. In August 2013 he entered the doctoral program in Electrical Engineering at The University of Tennessee, Knoxville. He worked as software engineer in Google from September to December in 2018. His paper got best paper award of IEEE Globecom 2017.