Two New Applications of Tensors to Machine Learning for Wireless Communications

by

Keerthana Bhogi

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Harpreet S. Dhillon, Chair
R. Michael Buehrer
Timothy Talty

August 9, 2021
Blacksburg, Virginia

Keywords: Tensor, machine learning (ML), Grassmann manifold (GM), full-dimension multiple-input multiple-output (FD-MIMO), federated learning (FL), neural network (NN).

Two New Applications of Tensors to Machine Learning for Wireless Communications

Keerthana Bhogi

ABSTRACT

With the increasing number of wireless devices and the phenomenal amount of data that is being generated by them, there is a growing interest in the wireless communications community to complement the traditional model-driven design approaches with data-driven machine learning (ML)-based solutions. However, managing the large-scale multi-dimensional data to maintain the efficiency and scalability of the ML algorithms has obviously been a challenge. Tensors provide a useful framework to represent multi-dimensional data in an integrated manner by preserving relationships in data across different dimensions. This thesis studies two new applications of tensors to ML for wireless communications where the tensor structure of the concerned data is exploited in novel ways.

The first contribution of this thesis is a tensor learning-based low-complexity precoder codebook design technique for a full-dimension multiple-input multiple-output (FD-MIMO) system with a uniform planar antenna (UPA) array at the transmitter (Tx) whose channel distribution is available through a dataset. Represented as a tensor, the FD-MIMO channel is further decomposed using a tensor decomposition technique to obtain an optimal precoder which is a function of Kronecker-Product (KP) of two low-dimensional precoders, each corresponding to the horizontal and vertical dimensions of the FD-MIMO channel. From the design perspective, we have made contributions in deriving a criterion for optimal product precoder codebooks using the obtained low-dimensional precoders. We show that this product codebook design problem is an unsupervised clustering problem on a Cartesian Product Grassmann Manifold (CPM), where the optimal cluster centroids form the desired codebook. We further simplify this clustering problem to a K-means algorithm on the low-dimensional factor Grassmann manifolds (GMs) of the CPM which correspond to the horizontal and vertical dimensions of the UPA, thus significantly reducing the complexity of precoder codebook construction when compared to the existing codebook learning techniques.

The second contribution of this thesis is a tensor-based bandwidth-efficient gradient communication technique for federated learning (FL) with convolutional neural networks (CNNs). Concisely, FL is a decentralized ML approach that allows to jointly train an ML model at the server using the data generated by the distributed users coordinated by a server, by sharing only the local gradients with the server and not the raw data. Here, we focus on efficient compression and reconstruction of convolutional gradients at the users and the server, respectively. To reduce the gradient communication overhead, we compress the sparse gradients at the users to obtain their low-dimensional estimates using compressive sensing (CS)-based technique and transmit to the server for joint training of the CNN. We exploit a natural tensor structure offered by the convolutional gradients to demonstrate the correlation of a gradient element with its neighbors. We propose a novel prior for the convolutional gradients that captures the described spatial consistency along with its sparse nature in an appropriate way. We further propose a novel Bayesian reconstruction algorithm based on the Generalized Approximate Message Passing (GAMP) framework that exploits this prior information about the gradients. Through the numerical simulations, we demonstrate that the developed gradient reconstruction method improves the convergence of the CNN model.

Two New Applications of Tensors to Machine Learning for Wireless Communications

Keerthana Bhogi

GENERAL AUDIENCE ABSTRACT

The increase in the number of wireless and mobile devices have led to the generation of massive amounts of multi-modal data at the users in various real-world applications including wireless communications. This has led to an increasing interest in machine learning (ML)-based data-driven techniques for communication system design. The native setting of ML is *centralized* where all the data is available on a single device. However, the distributed nature of the users and their data has also motivated the development of distributed ML techniques. Since the success of ML techniques is grounded in their data-based nature, there is a need to maintain the efficiency and scalability of the algorithms to manage the large-scale data. Tensors are multi-dimensional arrays that provide an integrated way of representing multi-modal data. Tensor algebra and tensor decompositions have enabled the extension of several classical ML techniques to tensors-based ML techniques in various application domains such as computer vision, data-mining, image processing, and wireless communications. Tensors-based ML techniques have shown to improve the performance of the ML models because of their ability to leverage the underlying structural information in the data.

In this thesis, we present two new applications of tensors to ML for wireless applications and show how the tensor structure of the concerned data can be exploited and incorporated in different ways. The first contribution is a tensor learning-based precoder codebook design technique for full-dimension multiple-input multiple-output (FD-MIMO) systems where we develop a scheme for designing low-complexity product precoder codebooks by identifying and leveraging a tensor representation of the FD-MIMO channel. The second contribution is a tensor-based gradient communication scheme for a decentralized ML technique known as federated learning (FL) with convolutional neural networks (CNNs), where we design a novel bandwidth-efficient gradient compression-reconstruction algorithm that leverages a tensor structure of the convolutional gradients. The numerical simulations in both applications demonstrate that exploiting the underlying tensor structure in the data provides significant gains in their respective performance criteria.

To my parents, brother, and amazing friends

Acknowledgments

First and foremost, I would like to thank my advisor Dr. Harpreet S. Dhillon for giving me the opportunity to work with him during my MS. Thank you for believing in me even before I started MS at VT and bearing for the initial few months until I understood research. I am extremely grateful to Dr. Chiranjib Saha for being an amazing mentor and a friend. I could not imagine completing this thesis without your guidance and support. Thank you for keeping up with my mistakes through these two years and teaching me life-skills very patiently. I hope to continue to work with you in the future.

A big shout-out to my family at VT: Shagun, Richa, Vaibhav, Sanjana, Balvansh, Ayush. Thank you for being my survival kit for the last two years. I am going to miss all the laughter, dinners, trips, and cribbing on Bollywood movies. I wish a very good luck to all of you for your future ventures. Shagun, Sanjana, and Richa, you people have been amazing roommates. A special thanks to Shagun for being my major support system at VT. And Bharat, thank you for being an integral part of my journey for the last four years.

I am lucky to have some of the smartest minds as colleagues: Dr. Chiranjib Saha, Anish Pradhan, Morteza Banagar, Dr. J. Kartheek Devineni, Dr. Vishnu Vardhan Chetlur, Dr. Priyabrata Parida. I wish a very good luck to Anish and Morteza for the rest of their PhD. Anish, a special thanks to you for being there for all my late night video calls.

I sincerely acknowledge Laura and Hilda for making lives of grad students so easy with your perfect paperwork and management. Roddy and John, thank you for helping me out in working with GPU and being there whenever I had the smallest query.

Last but not least, I would like to thank my parents for always believing in me, supporting every decision I have made till now, right from joining a boarding school in grade 6 to being the first in the family to go to a foreign country for MS. I wish a very good luck to my sweet brother for his future.

We would like to acknowledge financial support from the US National Science Foundation (NSF Grant ECCS-1731711).

Contents

List of Figures

List of Tables

1

Introduction

With the massive amounts of multi-dimensional data that is being generated by a massive number of distributed devices in various real-world applications, such as computer vision, data-mining, wireless communications, bioinformatics, and neuroscience, there is an increasing need to maintain the efficiency and scalability of the data analysis, signal processing, and ML algorithms. Tensors are multi-dimensional arrays that provide a natural and concise representation of high-dimensional multi-modal data occurring in such applications. As an example, matrices are two-dimensional arrays or second-order tensors. However, while processing the data, the multi-dimensional data is converted to a matrix or a vector form for various learning and inference tasks like feature extraction, classification, or clustering. When the multi-dimensional input data is rearranged into a vector or matrix, the inherent correlation of the data among different dimensions is discarded, making it harder for the learning or inference techniques to exploit the neighborhood relationship between entries of the data. Tensor representation of data respects the multi-dimensional structure and hence retains the structural spatial information which can offer efficient feature extraction from the data. Another issue is that the size of the vectorized data could easily become prohibitively large. The large size of the vector may lead to high computational complexity, huge memory requirements, and the *curse of dimensionality*. Hence, the algorithms that rely on the tensor representation of data are of significant importance. We can leverage the hidden structure within tensor data if the analysis tools account for the multi-dimensional patterns.

Tensor algebra and tensor decompositions [3–6] provide useful tools for representing and analyzing multi-dimensional data in a compact way. A tensor decomposition decomposes a given tensor into a sequence of simpler elementary tensors. Some of the well exploited tensor decomposition techniques in various research communities are Tucker [7], CP [8], and Tensor Train (TT) [9] decompositions. Historically, tensor decompositions have been introduced

for applications in Psychometrics [10] and Chemometrics [11], which have been followed by diverse applications in engineering, science, mathematics, and many more. Tensor algebra enabled natural generalizations of some commonly used signal processing algorithms such as linear regression [12], logistic regression [13], classification [14], CS [15] to tensors. Avoiding multi-dimensional data structure loss, tensor computation has been shown to bring enhancement in a number of classical signal processing techniques. The signal processing paradigms designed for tensors are versatile, which have been known to form the basis for many applications (see [16–19] and references therein) in domains like image, audio and video processing, wireless communications, ML, data mining, and computer vision.

The observed signals in the above examples are often multi-dimensional in nature. In particular, tensor modeling appears in several existing wireless communication studies where the received signal or the channel is multi-dimensional in nature. For signals that occur in wireless communications, the dimensions of the signal usually correspond to *frequency*, *space*, *time* or *users*. For instance, the channel in a multi-carrier MIMO systems such as Orthogonal Frequency Division Multiplexing (OFDM) systems can be modeled as a 3D tensor with transmit antenna, receive antenna, and delay (time) as the dimensions. This tensor channel model along with its sparse nature has been exploited to develop a tensor-CS-based channel estimation technique in [20]. Space-time coding is a common diversity scheme in MIMO systems which allows to build a third-order tensor model for the received signal, where the third dimension is *spreading*, which appears due to the use of a direct sequence spreading at the transmitter along with the common space and time dimensions. For other applications of tensors in wireless communications, we advise readers to refer [21–23] and references therein.

Tensors have garnered attention in data-intensive fields which often require an efficient representation of data. Data structures for conventional learning methods are usually restricted to vectors. Recently, various supervised (regression [13, 24, 25], classification [26]) and unsupervised learning (subspace learning [27], clustering [28, 29], dimensionality reduction [30]) methods that are usually designed for vectors or matrices are extended to tensors using tensor algebra and tensor decompositions. Tensor representation of data not only helps in designing the learning algorithms, but it may also motivate tensor-based data preprocessing or feature extraction [31] that aid the existing learning algorithms.

The above works have been presented as examples to demonstrate few ways to exploit the tensor structure of the data to develop tensor-based techniques but are not meant to be exhaustive. A reason for the success of tensor-based techniques is their very appealing property to efficiently leverage the underlying structural information in the data. From the various examples provided, it is clear that there is no established set of rules on how to identify the structure in the data. This thesis lies at the confluence of tensor representation of data and ML for wireless communications. We demonstrate two applications of ML for wireless communications where the inclusion of tensors gives a new perspective to the corresponding existing problems. Concisely, the first application deals with the channel state information (CSI) quantization of FD-MIMO systems for precoder codebook design, and the second application deals with bandwidth-efficient communication (compression-reconstruction) of gradients in a distributed ML system. The motivation behind both the applications is that the data of interest in each case naturally admits a tensor representation whereas the existing works vectorize or matricize the data and ignore the tensor structure which damages its inherent structure. In the rest of the chapter, we discuss the two applications in detail.

1.1 Background and contributions

In this section we describe the motivation behind the applications and present the respective contributions.

1.1.1 Precoder codebooks for FD-MIMO systems

Background

In frequency division duplexing (FDD) MIMO systems equipped with a limited feedback channel for CSI feedback to the transmitter (Tx), quantization of CSI is necessary. The most well-known approach for CSI quantization in FDD systems is to construct a dictionary of candidate precoders, called *precoder codebook*, that is known to both the Tx and receiver (Rx). The design of precoder codebooks for MIMO systems has been studied extensively under various design criteria, such as rate maximization or bit-error minimization [32], and statistical channel model assumptions, such as independently and identically distributed (i.i.d.) [33] or correlated Rayleigh channel [34] (see [35] for a comprehensive review). Although there exist

well-engineered codebook design methods, the efficiency of these codebooks is partially due to the simplifying assumptions on the channel model and its statistics, which offer analytical tractability in the codebook design. It may not always be possible to model a real propagation environment or expect the analytical tractability of the model. Therefore, there is a need to *learn* the underlying channel distribution from the observations of the propagation environment. With the advent of ML, there is a paradigm shift from model-driven to data-driven MIMO techniques. The precoder codebook design problem has also gained attention from this perspective. A foundational data-driven precoder codebook design method was proposed in [1] which involves vector quantization (VQ) of the space of optimal precoders i.e. right singular matrices of the channel instantiations in the available training dataset. As the number of antennas increase (i.e., for massive MIMO), the dimensionality of the channel increases and introduces quantization or clustering in large dimensions which is not very efficient due to the curse of dimensionality [36]. Due to the feature extraction ability of convolutional layers, CNNs have been used to construct autoencoders [37,38], which demonstrate the ability to learn an unknown low-dimensional feature space of the CSI using an encoder at the Rx and the corresponding decoder at the Tx. However, for MIMO precoding, the feature space is shown to be a GM in some cases [33]. Given that the underlying feature space of the precoders is known, NNs could be an overkill to this CSI quantization problem because of the complexity of their implementation. Therefore, we resort to a "shallow" learning technique that exploits the algebraic properties of optimal precoders, and FD-MIMO channel for efficient CSI quantization even for a large number of Tx antennas.

Contributions

In Chapter 2, we develop an efficient learning procedure for designing low-complexity precoder codebooks for FD-MIMO systems with a UPA antenna at Tx using tensor learning. In particular, instead of using statistical channel models, we utilize a model-free data-driven approach with foundations in ML to generate codebooks that adapt to the surrounding propagation conditions. We identify a natural tensor representation for the FD-MIMO channel and exploit its properties to design a quantized version of the channel precoders. We find the best representation of the optimal precoder as a function of KP of two low-dimensional

precoders, respectively, corresponding to the horizontal and vertical dimensions of the UPA obtained from the tensor decomposition of the channel. We then quantize this precoder to design product codebooks such that an average loss in mutual information due to the quantization of CSI is minimized. The key technical contribution lies in exploiting the constraints on the precoders to reduce the product codebook design problem to an unsupervised clustering problem on a CPM, where the cluster centroids form a finite-sized precoder codebook. This codebook can be found efficiently by running a K-means clustering on the CPM. With a suitable induced distance metric on the CPM, we show that the construction of product codebooks is equivalent to finding the optimal set of centroids on the factor manifolds corresponding to the horizontal and vertical dimensions. We also present the simulation results that demonstrate the capability of the proposed design criterion in learning the codebooks and their attractive performance that stems from the tensor representation of the FD-MIMO channel.

1.1.2 Communication-efficient FL

Background

While most of the conventional ML approaches rely on the assumption of having the data and processing heads in a central entity, this is not always feasible in applications with distributed users because of the privacy issues in sharing user-generated data and large communication overhead required to transmit raw data to central ML processors. As a result, decentralized ML approaches such as FL [39], that allows the server to collectively reap the benefits of the distributed data without the need to centrally store the data are much more appealing. However, exchanging the information necessary for FL training, such as model gradients consumes significant bandwidth, and thus, there is a need for communication-efficient techniques to perform distributed training.

Numerous studies have focused on communication-efficient FL schemes, where there are two main approaches: *gradient quantization* and *gradient sparsification*. To reduce the communication cost, the most intuitive idea is to transmit gradients with reduced precision by using a limited number of bits to represent the gradients efficiently by exploiting their stochastic properties. For instance, TernGrad [40] quantizes each element of a gradient vector

to either of the three values $\{-1, 0, 1\}$, signSGD [41] quantizes each element of gradient to $\{-1, 1\}$ according to its sign, Quantized SGD (QSGD) [42] scales the gradient vector by its norm and quantizes each entry of the scaled gradient. Recently, an adaptive gradient quantization technique [43], which automatically adapts the number of quantization levels used to represent a model update has been proposed. Gradient sparsification is another lossy compression technique that reduces the dimension of the gradient by retaining only a few important entries and sets all the remaining entries of the gradient to zero. Despite being lossy, gradient sparsification has demonstrated significant compression in gradients (and hence reduction in communication bandwidth) without drastic loss in performance of the models. Gradient sparsification has been widely combined with error accumulation [44] to accumulate the residual during sparsification and add them to the next gradient to reduce the loss in gradient information.

The gradient communication to the server takes place over the wireless channel between the devices and the server. However, most of the quantization and sparsification approaches assume reliable links and ignore the wireless nature of the communication medium. Therefore, one of the key challenges in designing communication-efficient gradient compression techniques is to incorporate the physical properties of the wireless medium such as fading, noise impairments, interference from multiple users. Motivated by the multi-user scenario in FL, [45] proposed a noisy Gaussian MAC to model the shared wireless channel between the users and the server and incorporate the interference from the users, and channel noise. The authors exploited the MAC property of the channel and proposed an analog-distributed stochastic gradient descent (A-DSGD) algorithm in which the server is not interested in the individual local gradients, but only in their average. In Chapter 3, we assume an FL system where the distributed users with local datasets implement DSGD with the help of a central server to collaboratively train a CNN model for a learning task. We adopt the Gaussian MAC model from [45] for modeling the wireless channel between the users and the server for communicating the local gradient estimates to the server. Most of the quantization techniques exploit the statistical properties of the gradients and sparsification exploits the sparsity of the gradients. However, both techniques vectorize the gradients of a model irrespective of its original structure. CNNs are popularly applied to images that are represented as 2D or

3D tensors. Therefore, the convolutional gradients offer a natural 4D tensor representation that stems from the 4D tensor structure of the convolutional kernel that transforms a 3D input tensor into a 3D output tensor. We demonstrate a gradient communication technique that leverages the structural properties of a convolutional gradient to further reduce the communication overhead.

Contributions

We propose a novel gradient compression-reconstruction scheme for convolutional gradient communication for FL. The convolutional gradients are also known to be sparse (approximately) in nature. We apply gradient sparsification on the gradients and use a linear transformation with a random projection matrix to compress the gradients as in CS to low-dimensional gradient estimates. We exploit the 4D tensor structure of the convolutional gradients to identify and demonstrate a strong spatial domain consistency between a gradient element and its neighboring elements corresponding to different dimensions of the tensor. We propose a novel prior formulation, a *spike and slab* prior for modeling the convolutional gradients that allow incorporating this prior knowledge about the spatial correlation along with the sparsity. We use the well-known GAMP algorithm for designing the Bayesian gradient reconstruction algorithm that integrates the proposed prior on the convolutional gradients. The numerical results show that the considered FL system using the proposed gradient reconstruction algorithm for convolutional gradient recovery at the server achieves faster convergence than the existing technique that is oblivious of the aforementioned spatial consistency.

1.2 Organization

The technical contributions of the thesis are provided in Chapters 2 and 3. Specifically, Chapter 2 develops a procedure for data-driven low-complexity product precoder codebooks for an FD-MIMO system. The design criterion and the corresponding algorithm for obtaining the optimal product precoder codebooks that exploit the underlying tensor representation of the FD-MIMO channel are provided. Chapter 3 provides a framework for FL to collaboratively train a CNN which employs gradient compression for efficient gradient communication

to the server. A Bayesian reconstruction algorithm for convolutional gradient recovery at the server is developed by imposing a novel prior that captures the properties pertinent to the convolutional gradients. Finally, Chapter 4 summarizes the key contributions of this thesis and provides some potential directions for the extension of this research.

2

Tensor Learning-based Precoder Codebooks for FD-MIMO Systems

With the availability of an unprecendented amount of data, there is a significant interest in applying ML to a variety of problems in communications and signal processing [46, 47]. Many of these problems also have a rich history of research that has led to key insights about their general structures and properties, which are collectively referred to *domain knowledge*. It is well-acknowledged in the ML community that incorporating this domain knowledge in learning algorithms results in efficient solutions, which has generated significant interest around the general idea of *theory-guided ML* [48]. The use of domain knowledge, such as the topological manifold on which the data is lying, often reduces the complexity of the ML models.

In this chapter, we explore the merger of domain knowledge and learning algorithm for the codebook design problem for limited feedback FDD MIMO systems. It is a classical problem in MIMO systems, where the CSI at the Rx needs to be quantized before sending over the limited capacity feedback channel to the Tx for precoding [49]. This codebook design problem has been studied extensively under several statistical channel models (see [35] for a comprehensive survey on *model-based codebooks*) but recently gained attention from the perspective of ML. The reason is that this problem can be viewed as a clustering problem where the set of optimal cluster centers represent the CSI whose distribution is available as a *training set*. Since the fundamental difficulty in this problem is the dimensionality of the channel, the natural tendency is to think in terms of obtaining a low dimensional representation of the channel using deep learning (DL) techniques, such as autoencoders, and use it for codebook construction [37, 38]. An autoencoder operates on the hypothesis

that the data possesses a representation on a lower-dimensional manifold (referred to as feature space), *albeit* unknown, and tries to learn the embedded manifold by training over the dataset [50, Chapter 14]. In contrast, for MIMO beamforming and precoding, the underlying manifold is known to be a GM in some cases [33]. This removes the requirement of "learning" the manifold from the dataset which oftentimes can be extremely complicated. Once the manifold is known, we can leverage the shallow learning techniques like the clustering algorithms on the manifold to find the *precoder codebook*.

2.1 Related work

In a limited feedback FDD-MIMO system, the assumption is that the Tx and Rx agree upon a common precoder codebook. The Rx, after the channel estimation, finds a precoder from this codebook and transmits the corresponding index over the feedback channel to Tx. There are various kinds of codebook design methods based on the above described two philosophies.

Model-based Approach. For i.i.d Rayleigh fading channels, the codebook design problem for precoding is equivalent to packing the subspaces in a GM of appropriate dimensions [33, 51]. For correlated channels, the Grassmann codebook can be modified by applying a channel correlation matrix [34, 52]. The basis of this modification is the assumption that the channel matrix is assumed to be factored into the square-root channel correlation matrix (or the long-term statistics of the channel) and the i.i.d. Rayleigh fading channel (or the instantaneous CSI) [53]. Apart from the Rayleigh fading assumption, another widely used channel model is the spatial channel model (SCM) [54], which has led to the design of discrete Fourier transform (DFT) structured codebooks. The principle of DFT codebooks is to quantize the direction of arrival of the dominant radio path of the channel. Based on the same principle, more advanced hierarchical DFT codebooks were developed. One prominent example of hierarchical codebooks is the so-called double DFT codebooks, where the two codebooks are designed for quantizing the long-term and instantaneous components of the precoder [55]. While the codebooks were primarily developed for linear antenna arrays at Tx and Rx, for FD-MIMO systems these codebooks can be extended by the formulation of *product codebooks*. The product codebook is simply a product (such as KP) of two codebooks

corresponding to the antenna arrays across the horizontal and vertical dimensions. The basis of this design is the Kronecker correlation model that approximates the channel correlation matrix with the KP of channel correlation matrices of horizontal and vertical dimensions. The decomposition of the channel correlation matrix of UPA enables the natural extension of the existing codebooks, e.g. Grassmannian codebooks [56] and DFT codebooks [57–60] for FD-MIMO systems.

Data-driven Approach. Unlike the model-based approach, a more direct approach for codebook design is to *learn* the codebooks from the channel datasets available through extensive channel measurements. The first comprehensive work in this direction is [1], where designing precoder codebooks is shown to be equivalent to a problem of VQ on the space of optimal precoders i.e., right singular matrices of the channel matrices in the training dataset. However, this technique is not useful when the number of antennas increases due to the curse of dimensionality. As an alternate approach, the CSI compression has been cast as an autoencoder problem, where the encoder residing at the receiver compresses and quantizes CSI and the decoder at Tx reconstructs the CSI. The extent of CSI compression of MIMO channels of arbitrary channel statistics and correlation properties in this scheme can be significantly enhanced by using NN-based (more precisely, CNN) structures for the encoder and decoder [37, 38, 61, 62]. Although these DL-based approaches have shown promising results compared to the state-of-the-art CSI compression techniques, their practical importance is questionable. The reason is that the performance is achieved only after using significantly complex architectures of NNs which is prone to a complicated hyperparameter tuning for any particular propagation environment. While the CNN-based techniques were designed to operate on datasets that have natural interpretations in the Euclidean domain (such as images), we can extend CNNs to build autoencoders that operate on topological manifolds. However, it can be very challenging to design such models and is still vastly considered as an open problem in ML. Therefore, in this chapter, we propose an alternate formulation for the data-driven precoder design for FD-MIMO channels by building on the ideas of Grassmannian K-means clustering developed in the sequel. However, as we discussed before, extending this method for higher dimensions of channels is not straightforward. Interestingly, the FD-MIMO systems naturally admit a tensor representation of the channel [63–65]. This enables

us to leverage tools from a more classical form of ML, known as *tensor learning* [17, 66, 67], along with ideas from theory-guided ML to constrain the outputs to a topological manifold, to formulate computationally efficient product codebooks for precoding even for a large number of Tx antennas.

2.2 Contributions and novelty

In this chapter, we propose a data-driven precoder codebook design method by exploiting a tensor representation of the FD-MIMO channel. We reduce the dimensionality of the channel tensor by decomposing it into low-dimensional orthonormal factors using the low-rank Tucker decomposition (TD). This operation simplifies the codebook design explained as follows.

First, the Rx computes the unquantized precoder from the channel tensor as a function of KP of the two low-rank TD factors corresponding to the horizontal and vertical dimensions of the UPA at the Tx. We adopt this KP structure of the unquantized precoders to the quantized precoders as well. We show that this KP structure of the precoders admits a representation on a *Tensor Product Grassmann Manifold* (TPM), where each factor is a GM corresponding to horizontal and vertical dimensions of the UPA at the Tx. We define a measure of loss in mutual information associated with an arbitrary precoder and use it to define the average mutual information loss due to the limited feedback, leading to a new codebook design criterion. With the rotational invariance property of the precoders and the induced chordal distance metric on a GM, we show that the obtained codebook design criterion is equivalent to minimizing the average distortion in representing the optimal unquantized precoders with quantized precoders on a TPM.

Second, we exploit the diffeomorphism between a TPM and a CPM to approximate the described quantization loss as the average distortion between the representations of the optimal unquantized and quantized precoder on the CPM. We show that the optimal product precoder codebook minimizing the defined average distortion due to quantization is equivalent to the set of optimal centroids given by the K-means clustering algorithm on the CPM. The induced chordal distance metric is inherited from the factor GMs to define the chordal distance on a CPM. This provides a natural extension of the K-means

clustering algorithm on a GM to a CPM. With this induced chordal distance metric, we show that the K-means clustering problem on a CPM is reduced to separate K-means clustering problems on its factor manifolds. This simplifies the product precoder codebook construction to finding the optimal set of centroids using the K-means clustering on its factor manifolds corresponding to the horizontal and vertical dimensions of the UPA at the Tx. We also formally show that the proposed tensor-based product codebook design is computationally more efficient than its VQ counterpart which does not use the tensor representation of the FD-MIMO channel, proposed in [1], in terms of asymptotic complexity.

Notations. We use $\mathbf{a} \in \mathbb{C}^{M\times 1}$, $\mathbf{A} \in \mathbb{C}^{M\times N}$, to designate complex column vectors, matrices, respectively, $\mathbf{A}(:,i)$ or $\mathbf{a}_i$ to denote the i-th column, $\mathbf{A}(:,i:j)$ to represent an $M \times (j-i+1)$ matrix, formed by i-th to j-th columns of $\mathbf{A}$ for $1 \leq i \leq j \leq N$. If $\mathcal{I} = \{i_1, \cdots, i_n\}$ denotes a set of indices where $1 \leq i_1 < \cdots < i_n \leq N$, then $\mathbf{A}(:,\mathcal{I})$ or $\mathbf{A}_{\mathcal{I}}$ represents an $M \times |\mathcal{I}|$ matrix formed by the columns of $\mathbf{A}$ whose indices are given by $\mathcal{I}$. We use $\mathcal{U}(M,N)$, $\mathcal{U}_M$ to represent the set of all $M \times N$ complex orthonormal matrices, $M \times M$ unitary matrices, respectively. Further, $a^*(\mathbf{a}^*)$ denotes the complex conjugate of $a \in \mathbb{C}$ ($\mathbf{a} \in \mathbb{C}^{M\times 1}$), $\mathbf{A}^T$, $\mathbf{A}^H$ denote transpose, Hermitian, vec($\mathbf{A}$) denotes the vectorization of $\mathbf{A}$, $\mathbb{E}_A$ denotes expectation over the distribution of A where A is a random matrix or vector. Also, $|\cdot|, \|\cdot\|_F$ denote the absolute value, the Frobenius norm and $j = \sqrt{-1}$.

2.3 System overview

We consider a narrow-band point-to-point MIMO communication system, where the Tx and Rx are equipped with M_t and M_r antennas, respectively. We assume a block fading channel model and represent the channel between Tx and Rx as $\mathbf{H} \in \mathbb{C}^{M_r\times M_t}$. Throughout this chapter, we assume that $M_r \leq M_t$ and let the rank of the channel matrix $\mathbf{H}$ be $\mathtt{r}_o \leq M_r$. The Tx is equipped with a UPA antenna with M_v and M_h antennas in the vertical and horizontal dimensions, respectively with $M_t = M_v M_h$ and the Rx is equipped with a ULA antenna with M_r antennas. The discrete-time baseband input-output relation for this system can be expressed as $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$, where $\mathbf{x} \in \mathbb{C}^{M_t\times 1}$ is the transmitted signal, $\mathbf{y} \in \mathbb{C}^{M_r\times 1}$ is the received signal and $\mathbf{n} \in \mathbb{C}^{M_r\times 1}$ is the additive white Gaussian noise distributed as $\mathcal{CN}(\mathbf{0}, N_o\mathbf{I}_{M_r})$. The average total transmit power is denoted as $\mathcal{E}_s$ where $\mathcal{E}_s = \mathbb{E}[\mathbf{x}^H\mathbf{x}]$. The

SVD of $\mathbf{H}$ is given by $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$, where $\mathbf{U} \in \mathcal{U}_{M_r}$, $\mathbf{V} \in \mathcal{U}_{M_t}$, and $\mathbf{\Sigma}$ is the $M_r \times M_t$ rectangular diagonal matrix with i-th largest singular value σ_i at the entry (i, i).

2.3.1 Beamforming

For the simplicity of exposition, we first consider a multiple-input single-output (MISO) system, where the Rx is equipped with a single antenna i.e., $M_r = 1$. In order to improve the received SNR, the Tx performs beamforming. For this case, the received signal $\mathbf{y}$ simplifies to $y = \mathbf{H}\mathbf{f}s + n$, where $s \in \mathbb{C}$ is the transmitted symbol with average power $\mathbb{E}_s[s^*s] = \mathcal{E}_s$, $\mathbf{f} \in \mathbb{C}^{M_t \times 1}$ is the beamformer. Assuming that the Rx employs maximal ratio combining (MRC) [51], the Rx uses $z = \frac{\mathbf{H}\mathbf{f}}{\|\mathbf{H}\mathbf{f}\|_2}$ to estimate the transmitted symbol $\hat{s}$ which is simplified as $\hat{s} = z^H y = y$. This gives the receive SNR ρ_r as $\rho_r = \mathcal{E}_s \frac{\|\mathbf{H}\mathbf{f}\|_2^2}{\|\mathbf{n}\|_2^2 \|\mathbf{f}\|_2^2} = \frac{\mathcal{E}_s}{N_o} \frac{\|\mathbf{H}\mathbf{f}\|_2^2}{\|\mathbf{f}\|_2^2} = \rho_t \frac{\|\mathbf{H}\mathbf{f}\|_2^2}{\|\mathbf{f}\|_2^2}$ where $\mathcal{E}_s/N_o$ is the transmit SNR ρ_t. The total transmit power $\mathbb{E}[\mathbf{x}^H\mathbf{x}] = \mathbb{E}[\|\mathbf{f}s\|_2^2] = \mathcal{E}_s$ is assumed to be fixed. Because of this, we have the unit norm constraint on the beamformer, i.e., $\|\mathbf{f}\|_2^2 = 1$ and thus $\mathbf{f} \in \mathcal{U}(M_t, 1)$. Following this constraint, the beamforming gain $\Gamma(\mathbf{H}, \mathbf{f})$ is obtained as $\Gamma(\mathbf{H}, \mathbf{f}) := \rho_r/\rho_t = \|\mathbf{H}\mathbf{f}\|_2^2$. The problem of transmit beamforming is to maximize $\Gamma(\mathbf{H}, \mathbf{f})$ i.e., $\hat{\mathbf{f}} = \underset{\mathbf{f} \in \mathcal{U}(M_t,1)}{\arg\max}\, \Gamma(\mathbf{H}, \mathbf{f}) = \underset{\mathbf{f} \in \mathcal{U}(M_t,1)}{\arg\max}\, \|\mathbf{H}\mathbf{f}\|_2^2$. One possible solution for the optimal beamformer $\hat{\mathbf{f}}$ is the right singular vector that is associated with the maximum singular value of $\mathbf{H}$ i.e., $\hat{\mathbf{f}} = \mathbf{v}_1 = \mathbf{V}(:, 1)$ [68]. The corresponding beamforming gain is $\Gamma_{\max} = \underset{\mathbf{f} \in \mathcal{U}(M_t,1)}{\max}\, \Gamma(\mathbf{H}, \mathbf{f}) = \Gamma(\mathbf{H}, \mathbf{v}_1) = \|\mathbf{H}\mathbf{v}_1\|_2^2 = \sigma_1^2$. For transmit beamforming, it has been shown that the beamformer that maximizes the receive SNR ρ_r also maximizes the mutual information between s and y and minimizes the average probability of symbol error [69, 70].

2.3.2 Precoding

Let us now consider a general MIMO system with $M_r > 1$. Since $M_r > 1$, the system can support upto rank $-\,\mathtt{r}$ $(1 \leq \mathtt{r} \leq M_r)$ transmission or the transmission of $\mathtt{r}$ independent streams. For this scheme, we assume transmit precoding, i.e., the Tx transmits $\mathbf{s} \in \mathbb{C}^{\mathtt{r} \times 1}$, a symbol vector of $\mathtt{r}$ independent data streams, which is precoded with a precoder matrix $\mathbf{F} \in \mathbb{C}^{M_t \times \mathtt{r}}$. The transmitted signal $\mathbf{x}$ is obtained as $\mathbf{x} = \mathbf{F}\mathbf{s}$ resulting in the received signal $\mathbf{y} = \mathbf{H}\mathbf{F}\mathbf{s} + \mathbf{n}$. We assume equal power allocation strategy at the Tx where the total transmit power $\mathcal{E}_s$ is split equally among the $\mathtt{r}$ transmitted symbols i.e., $\mathbb{E}_{s_i}[s_i^* s_i] = \frac{\mathcal{E}_s}{\mathtt{r}}$ and

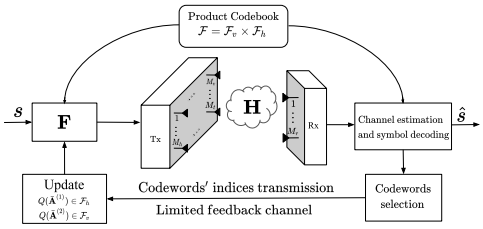


Figure 2.1: Block diagram of an FDD-MIMO system with limited feedback channel of capacity B bits per channel use.

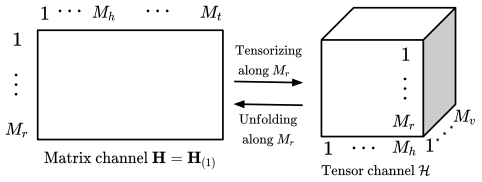


Figure 2.2: Illustration of the tensor representation of FD-MIMO channel and its relation with the matrix channel representation.

also assume that $\mathbf{s}$ is generated by an uncorrelated zero-mean jointly Gaussian symbol source. Thus, $\mathbf{s} \sim \mathcal{N}(\mathbf{0}, \frac{\mathcal{E}_s}{\mathtt{r}}\mathbf{I}_{\mathtt{r}})$. When the Tx precodes $\mathbf{s}$ with $\mathbf{F}$, the equivalent channel is $\mathbf{H}_{eq} = \mathbf{HF}$ and the transmit SNR per spatial stream is $\rho_t = \frac{\mathcal{E}_s}{N_o \mathtt{r}}$. The Rx uses a linear minimum mean square error (MMSE) combiner to estimate the transmitted symbol vector as $\hat{\mathbf{s}} = \mathbf{Z}_{\text{MMSE}}^H \mathbf{y}$ where $\mathbf{Z}_{\text{MMSE}} = \mathbf{H}_{eq}^H \left(\mathbf{H}_{eq}^H \mathbf{H}_{eq} + {\rho_t}^{-1}\mathbf{I}\right)^{-1}$. Under these assumptions, the mutual information $R(\mathbf{H}, \mathbf{F})$ between $\mathbf{s}$ and $\mathbf{y}$ for a given channel $\mathbf{H}$ and a precoder $\mathbf{F}$ is given by

$$R(\mathbf{H}, \mathbf{F}) = \log\det\left(\mathbf{I} + \rho_t \mathbf{H}_{eq}^H \mathbf{H}_{eq}\right) = \log\det\left(\mathbf{I} + \rho_t \mathbf{F}^H \mathbf{H}^H \mathbf{H}\mathbf{F}\right).$$

With full CSI at the Tx (CSIT), the strategy that maximizes the mutual information $R(\mathbf{H}, \mathbf{F})$ is to employ water-filling based optimal power allocation on the $\mathtt{r}$ independent data streams [32, 71]. This necessitates the knowledge of $\bar{\mathbf{V}} = \mathbf{V}(:, 1:\mathtt{r})$ and additionally $\mathbf{\Sigma}$, truncated upto $\mathtt{r}$ dominant singular values, to ensure optimal power splitting across the spatial streams at the Tx for precoding.

For the optimal beamforming (precoding), the Tx needs to know $\mathbf{v}_1$ $(\bar{\mathbf{V}}, \bar{\mathbf{\Sigma}})$. In an FDD system, the Rx estimates the channel $\mathbf{H}$ and sends $\mathbf{v}_1$ $(\bar{\mathbf{V}}, \bar{\mathbf{\Sigma}})$ back to the Tx over a feedback channel. Thus the feedback overhead increases as M_t increases. Since the feedback channel is typically assumed to be a low-rate, zero-delay, and error-free, with a limited capacity of B bits per channel use, it is not always possible to transmit $\mathbf{v}_1$ $(\bar{\mathbf{V}}, \bar{\mathbf{\Sigma}})$ over this channel without any data compression, especially when the number of antennas is large [49]. Thus, it is necessary to introduce some method to quantize $\mathbf{v}_1$ $(\bar{\mathbf{V}}, \bar{\mathbf{\Sigma}})$. The available B feedback bits per each channel use have to be utilized to convey the channel information to the Tx and maximize

the performance of the MIMO system. The most well-known approach for the quantization is to construct a finite-sized dictionary of beamformers (precoders) [49], also known as the *codebook*. In particular, for beamforming, the Tx and Rx agree upon a beamformer codebook, say $\mathcal{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_{2^B}\}, \mathbf{f}_i \in \mathcal{U}(M_t, 1)$. While there are multiple ways to define a precoder codebook for quantizing $\bar{\mathbf{V}}$, we focus on the most common approach of orthonormal precoder codebook where the precoders are always constrained to be orthonormal matrices[1] [33, 72]. The orthonormality constraint follows from the form of the optimal precoders derived with the maximum eigenvalue constraint on $\mathbf{F}$ under the presence of full CSIT [32]. Under the equal power allocation strategy and the orthonormality contraints on $\mathbf{F}$, an optimal rank$-\mathtt{r}$ precoder over $\mathcal{U}(M_t, \mathtt{r})$ that maximizes the mutual information $R(\mathbf{H}, \mathbf{F})$ is $\mathbf{F}_{\text{opt}} = \bar{\mathbf{V}}$ which is formed by the $\mathtt{r}$ dominant columns of $\mathbf{V}$ [32]. Thus a codebook $\mathcal{F}$ of cardinality 2^B with candidate precoder matrices is given as $\mathcal{F} = \{\mathbf{F}_1, \ldots, \mathbf{F}_{2^B}\}$, where $\mathbf{F}_i \in \mathcal{U}(M_t, \mathtt{r})$ and is assumed to be known to the Tx and Rx. The Rx chooses the appropriate beamformer $\mathbf{f} \in \mathcal{F}$ $\big($precoder $\mathbf{F} \in \mathcal{F}\big)$ that maximizes $\Gamma(\mathbf{H}, \mathbf{f})$ $(R(\mathbf{H}, \mathbf{F}))$ and feeds the index of the codeword back to the Tx. For a given beamformer codebook $\mathcal{F}$, the criterion for choosing the optimal beamformer can be stated as $\mathbf{f} = \arg\max_{\mathbf{f}_i \in \mathcal{F}} \Gamma(\mathbf{H}, \mathbf{f}_i) = \arg\max_{\mathbf{f}_i \in \mathcal{F}} \|\mathbf{H}\mathbf{f}_i\|_2^2$. Similarly, for a given precoder codebook $\mathcal{F}$, the criterion for choosing the optimal precoder is $\mathbf{F} = \arg\max_{\mathbf{F}_i \in \mathcal{F}} R(\mathbf{H}, \mathbf{F}_i)$. The system-level diagram of a limited feedback FDD-MIMO system is provided in Fig. 2.1.

2.4 Preliminaries

In this section, we briefly review the background of the topics including a few useful results that are used in developing the codebook design scheme proposed in the sequel.

2.4.1 Tensors

A tensor is a multi-dimensional array and the number of dimensions of the array is defined as the order of the tensor. A matrix, for instance is a two-dimensional array or second-order tensor. We denote an N-th order tensor complex tensor as $\mathcal{X} \in \mathbb{C}^{I_1 \times \cdots \times I_n \times \cdots I_N}$ whose

[1] With limited feedback bits available, we focus first on representing $\bar{\mathbf{V}}$ and do not allocate any bits for power allocation information i.e., $\bar{\mathbf{\Sigma}}$, thus assuming equal power allocation strategy.

$(i_1, \cdots, i_n, \cdots, i_N)$-th element is represented as $x_{i_1 i_2 \cdots i_N}$ or $[\mathcal{X}]_{i_1 i_2 \cdots i_N}$, where $1 \leq i_n \leq I_n$ for $n = (1, \cdots, N)$. The Frobenius norm of a tensor $\mathcal{X}$ is denoted as $\|\mathcal{X}\|_F$ and defined as the square root of the sum of the squares of absolute values of its elements i.e., $\|\mathcal{X}\|_F := \sqrt{\sum_{i_1=1}^{I_1} \cdots \sum_{i_N=N}^{I_N} |x_{i_1 \cdots i_N}|^2}$.

A tensor can be represented by a set of matrices which is possible through unfolding the tensor. The rows and columns of a matrix are generalized as mode-n fibers of a tensor. A mode-n fiber is formed by the set of elements of the tensor where $i_n = (1, \cdots, I_n)$ for a chosen $i_1, \cdots, i_{n-1}, i_{n+1}, \cdots, i_N$. The unfolding of a tensor $\mathcal{X}$ along its n-th dimension is called mode-n unfolding and the resultant matrix is denoted as $\mathbf{X}_{(n)} \in \mathbb{C}^{I_n \times J_n}$ where $J_n = \prod_{k=1, k \neq n}^{N} I_k$. The matrix $\mathbf{X}_{(n)}$ is formed by arranging the mode-n fibers of $\mathcal{X}$ as its columns. An element $x_{i_1 i_2 \cdots i_N}$ of $\mathcal{X}$ is mapped to (i_n, j)-th element of $\mathbf{X}_{(n)}$ where $j = 1 + \sum_{k=1, k \neq n}^{N} (i_k - 1) J_k$, $J_k = \prod_{m=1, m \neq n}^{k-1} I_m$. The product of a tensor and a matrix along the n-th dimension is represented as $\times_n$ and known as n-mode product. The n-mode product of a tensor $\mathcal{X}$ and a matrix $\mathbf{U} \in \mathbb{C}^{J \times I_n}$ is represented as $\mathcal{Y} = \mathcal{X} \times_n \mathbf{U}$ where $\mathcal{Y} \in \mathbb{C}^{I_1 \times \cdots I_{n-1} \times J \times I_{n+1} \times \cdots I_N}$ whose mode-n unfolding is given by $\mathbf{Y}_{(n)} = \mathbf{U}\mathbf{X}_{(n)}$.

Tucker decomposition of a tensor. TD decomposes a tensor into a core tensor and a set of orthonormal matrices corresponding to each mode of the tensor. It is also a form of higher-order principal component analysis [73] and TD of a tensor $\mathcal{X}$ is expressed as $\mathcal{X} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)}$, for $i_n = (1, \cdots, I_n)$, $n = (1, \cdots, N)$. The tensor $\mathcal{G} \in \mathbb{C}^{I_1 \times \cdots I_n \times \cdots I_N}$ is called the *core tensor* and the factor matrices $\mathbf{A}^{(n)} \in \mathcal{U}_{I_n}$. Let $\mathbf{G}_{(n)}$ be the mode-n unfolding of $\mathcal{G}$, then, from the TD of $\mathcal{X}$ we have, $\mathbf{X}_{(n)} = \mathbf{A}^{(n)} \mathbf{G}_{(n)} \left(\mathbf{A}^{(N)} \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)} \right)^T$. The matrices $\mathbf{A}^{(n)}$ can be thought of as the *principal components* in each mode and are analogous to principal components of a matrix. The core tensor $\mathcal{G}$ represents the interaction between different principal components of $\mathcal{X}$ and generally not a diagonal matrix as it is in the SVD of matrices.

Low-rank representation. A tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ can be approximated with a rank $-(r_1, \cdots, r_N)$ tensor $\bar{\mathcal{X}}$ as $\mathcal{X} \approx \bar{\mathcal{X}} = \bar{\mathcal{G}} \times_1 \mathbf{A}^{(1)}_{r_1} \times_2 \mathbf{A}^{(2)}_{r_2} \cdots \times_N \mathbf{A}^{(N)}_{r_N}$ where $\bar{\mathcal{G}} \in \mathbb{C}^{r_1 \times \cdots \times r_n \times \cdots \times r_N}$, $r_n \leq I_n$ for $n = (1, \cdots, N)$ and $\mathbf{A}^{(n)}_{r_n} \in \mathcal{U}(I_n, r_n)$ is a rank $- r_n$ orthonormal matrix. The

best rank $-(r_1, \cdots, r_N)$ approximation $\bar{\mathcal{X}}$ of $\mathcal{X}$ is obtained as

$$(\bar{\mathcal{G}}, \mathbf{A}_{r_1}^{(1)}, \cdots, \mathbf{A}_{r_N}^{(N)}) = \underset{\bar{\mathcal{G}}, \mathbf{A}_{r_i}^{(i)} \in \mathcal{U}(I_i, r_i)}{\arg\min} \left\| \mathcal{X} - \bar{\mathcal{G}} \times_1 \mathbf{A}_{r_1}^{(1)} \times_2 \mathbf{A}_{r_2}^{(2)} \cdots \times_N \mathbf{A}_{r_N}^{(N)} \right\|_F . \tag{2.1}$$

In the case of matrices, the principal components of the best low-rank approximation are obtained directly from its SVD [74], whereas for tensors, the above minimization problem has to be solved for obtaining the principal components of the tensor. One of the algorithms utilized for solving (2.1) is the *Higher-Order Orthogonal Iteration* (HOOI), which will be used in the sequel [3].

2.4.2 Overview of GMs

The *complex* GM $\mathcal{G}(n,k)$ [33] is defined as the set of all k dimensional linear subspaces spanned by orthonormal matrices $\mathcal{U}(n,k)$ i.e., $\mathcal{G}(n,k) := \{\text{span}(\mathbf{F}) : \mathbf{F} \in \mathcal{U}(n,k)\}$, where $\text{span}(\mathbf{F})$ is the k dimensional subspace in $\mathbb{C}^n$ spanned by the columns of the orthonormal basis $\mathbf{F}$. For any $\mathbf{Q} \in \mathcal{U}_k$, $\text{span}(\mathbf{FQ}) = \text{span}(\mathbf{F})$, i.e., the subspaces spanned by the columns of $\mathbf{F}$ and $\mathbf{FQ}$ are the same and are represented by an equivalence relation $\mathbf{F} \sim \mathbf{FQ}$. Therefore the matrix representation of a point in $\mathcal{G}(n,k)$ is not unique. We use the notation $\mathbf{F} \in \mathcal{G}(n,k)$ to represent the subspace $\text{span}(\mathbf{F})$. Let $\mathbf{F}_1, \mathbf{F}_2 \in \mathcal{G}(n,k)$, then the distance between the subspaces spanned by them is characterized by the principal angles between $\text{span}(\mathbf{F}_1), \text{span}(\mathbf{F}_2)$. A number of different geodesic distances between the subspaces can be defined. In this chapter, we will be using the *chordal distance*. The chordal distance (d_c) between two subspaces which are spanned by $\mathbf{F}_1, \mathbf{F}_2 \in \mathcal{U}(n,k)$ is defined as $d_c^2(\mathbf{F}_1, \mathbf{F}_2) := \frac{1}{2} \left\| \mathbf{F}_1 \mathbf{F}_1^H - \mathbf{F}_2 \mathbf{F}_2^H \right\|_F^2 = \left(k - \left\| \mathbf{F}_1^H \mathbf{F}_2 \right\|_F^2 \right) = \|\sin \mathbf{\Theta}\|_2^2$, where $\mathbf{\Theta} = [\theta_1, \cdots, \theta_k]$ and θ_i is the i-th principal angle between $\text{span}(\mathbf{F}_1)$ and $\text{span}(\mathbf{F}_2)$. Any element on a GM is invariant to rotations i.e., $\mathbf{F} \equiv \mathbf{FQ}$ for $\mathbf{Q} \in \mathcal{U}_k$. Therefore the chordal distance $d_c(\mathbf{F}_1, \mathbf{F}_2)$ is invariant under various representations of the subspaces, i.e., $d_c(\mathbf{F}_1, \mathbf{F}_2) = d_c(\mathbf{F}_1 \mathbf{Q}_1, \mathbf{F}_2 \mathbf{Q}_2)$ $\forall\ \mathbf{Q}_1, \mathbf{Q}_2 \in \mathcal{U}_k$.

Product GM

The m-fold CPM $\mathcal{G}^{\times}(\mathbf{n},\mathbf{k})$ is defined as the space $\mathcal{G}(n_1,k_1)\times\cdots\times\mathcal{G}(n_m,k_m)$. A point in $\mathcal{G}^{\times}(\mathbf{n},\mathbf{k})$ is represented as the collection of the points $\mathbf{F}_i\in\mathcal{G}(n_i,k_i)\ \forall i=1,\cdots,m$. Thus,

$$\mathcal{G}^{\times}(\mathbf{n},\mathbf{k}):=\{[\mathbf{F}]=(\mathbf{F}_1,\cdots,\mathbf{F}_m)|\mathbf{F}_i\in\mathcal{G}(n_i,k_i),i=1,\cdots,m\}, \tag{2.2}$$

where $(\mathbf{n},\mathbf{k}):=((n_1,k_1),(n_2,k_2),\cdots,(n_m,k_m))$. Just as different notions of distances on a GM [75], a distance metric on a CPM can be defined in different ways. We extend the chordal distance metric d_c on a GM to define the following distance metric to measure the distance between two points $[\mathbf{F}]$, $[\mathbf{F}']\in\mathcal{G}^{\times}(\mathbf{n},\mathbf{k})$: $d_c([\mathbf{F}],[\mathbf{F}']):=\|\sin\boldsymbol{\Theta}\|_2$, where $\boldsymbol{\Theta}=(\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_m)$, $\boldsymbol{\theta}_i$ is the set of principal angles between the i-th factor GM of $[\mathbf{F}]$ and $[\mathbf{F}']$, i.e., $\mathbf{F}_i$ and $\mathbf{F}'_i$ respectively. Using this expression, the chordal distance on a CPM can also be written as

$$d_c^2([\mathbf{F}],[\mathbf{F}'])=d_c^2\left((\mathbf{F}_1,\cdots,\mathbf{F}_m),(\mathbf{F}'_1,\cdots,\mathbf{F}'_m)\right)=\sum_{i=1}^{m}d_c^2(\mathbf{F}_i,\mathbf{F}'_i). \tag{2.3}$$

It implies that the squared chordal distance between two points on a CPM is equivalent to the sum of squares of distance between the points on the factor GMs that form the product space. This property will be particularly useful in the proposed product codebook construction. In the sequel, we will introduce another type of product GM, termed TPM, while designing the product codebook.

K-means clustering on a GM

The K-means clustering on a given metric space is a method of VQ to partition a set of N data points into K non-overlapping clusters, in which each data point belongs to the cluster with the nearest cluster centroid. The centroids are the quantized representations of the data points that belong to the respective clusters. A quantizer on the given metric space maps the data points to one of the K centroids. The K centroids are chosen such that the average distortion due to quantization is minimized. Before we formally introduce the main steps of the clustering algorithm on $\mathcal{G}(n,k)$, we first define the notion of a distortion measure and a quantizer as follows.

Definition 2.1 (Distortion measure)**.** The distortion caused by representing $\mathbf{F} \in \mathcal{G}(n,k)$ with $\mathbf{F}' \in \mathcal{G}(n,k)$ is defined as the distortion measure d_o which is given by $d_o(\mathbf{F},\mathbf{F}') = d_c^2(\mathbf{F},\mathbf{F}')$.

Definition 2.2 (Grassmann quantizer)**.** Let $\mathcal{F} \subseteq \mathcal{G}(n,k)$ be a B-bit codebook such that $\mathcal{F} = \{\mathbf{F}_1, ..., \mathbf{F}_{2^B}\}$, then a Grassmann quantizer $Q_{\mathcal{F}}$ is defined as a function mapping elements of $\mathcal{G}(n,k)$ to elements of $\mathcal{F}$ i.e., $Q_{\mathcal{F}} : \mathcal{G}(n,k) \mapsto \mathcal{F}$.

A performance measure of a Grassmann quantizer is the average distortion $D(Q_{\mathcal{F}})$, where $D(Q_{\mathcal{F}}) := \mathbb{E}_{\mathbf{X}}[d_o(\mathbf{X}, Q_{\mathcal{F}}(\mathbf{X})] = \mathbb{E}_{\mathbf{X}}[d_c^2(\mathbf{X}, Q_{\mathcal{F}}(\mathbf{X})]$. In most practical settings, we may have access to a set of N data points $\mathcal{X} = \{\mathbf{X}\} \subseteq \mathcal{G}(n,k)$ in lieu of the probability distribution $p(\mathbf{X})$. Then the expectation w.r.t $\mathbf{X}$ in $D(Q_{\mathcal{F}})$ means averaging over the set $\mathcal{X}$. Therefore the objective of K-means clustering with $K = 2^B$ is to find the set of K centroids, i.e., $\mathcal{F}^K$, that minimizes $D(Q_{\mathcal{F}})$ and can be expressed as

$$\mathcal{F}^K = \underset{\mathcal{F}\subseteq\mathcal{G}(n,k)|\mathcal{F}|=2^B}{\arg\min} D(Q_{\mathcal{F}}) = \underset{\mathcal{F}\subseteq\mathcal{G}(n,k)|\mathcal{F}|=2^B}{\arg\min} \mathbb{E}_{\mathbf{X}}\left[d^2(\mathbf{X}, Q_{\mathcal{F}}(\mathbf{X}))\right], \tag{2.4}$$

and the associated quantizer is $Q_{\mathcal{F}^K}(\mathbf{X}) = \underset{\mathbf{F}_i\in\mathcal{F}}{\arg\min}\, d_o(\mathbf{X},\mathbf{F}_i) = \underset{\mathbf{F}\in\mathcal{F}}{\arg\min}\, d_c^2(\mathbf{X},\mathbf{F}_i)$. However, finding the optimal solution for K-means clustering is an NP-hard problem. Therefore, we use the Linde-Buzo-Gray algorithm [76] (outlined in Alg. 1) which is a heuristic algorithm that iterates between updating the cluster centroids and mapping a data point to the corresponding centroid that guarantees convergence to a local optimum. In Alg. 1, the only non-trivial step is the centroid calculation for a set of points. In contrast to the squared distortion measure in the Euclidean domain, the centroid of a set of elements in a general manifold with respect to an arbitrary distortion measure does not necessarily exist in a closed form. However, the centroid computation on $\mathcal{G}(n,k)$ is feasible because of the following lemma [77].

Lemma 2.3 (Centroid computation)**.** *For a set of points* $\mathcal{S}_i = \{\mathbf{X}_j\}_{j=1}^{N_k}$, $\mathbf{X}_j \in \mathcal{G}(n,k)$, *that form the i-th Voronoi partition, the centroid* $\mathbf{F}_i$ *is* $\mathbf{F}_i = \underset{\mathbf{F}\in\mathcal{G}(n,k)}{\arg\min} \sum_{j=1}^{N_k} d_c^2(\mathbf{X}_j,\mathbf{F}) = \mathrm{eig}_r\left(\sum_{j=1}^{N_k} \mathbf{X}_j\mathbf{X}_j^H\right)$, *where the columns of* $\mathrm{eig}_r(\mathbf{Y})$ *are chosen to be the* r *dominant eigenvectors of the* $\mathbf{Y}$.

2.4.3 Submodular optimization

We now introduce a special form of optimization of *set functions* which will be a necessary building block of our proposed codebook design scheme. Consider a set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ which assign a real value to any subset $\mathcal{P}$ of a finite *ground set* $\mathcal{V} \neq \varnothing$. Then a function f is called *monotone* if $f(\mathcal{P} \cup \{a\}) - f(\mathcal{P}) \geq 0$ for all $\mathcal{P} \subseteq \mathcal{U}$, $a \notin \mathcal{P}$ and $a \in \mathcal{V}$. Further, a set function f is *submodular* if $f(\mathcal{P} \cup \{a\}) - f(\mathcal{P}) \geq f(\mathcal{T} \cup \{a\}) - f(\mathcal{T})$ for all possible pairs of subsets $\mathcal{P} \subseteq \mathcal{T} \subseteq \mathcal{V}$ and all elements $a \in \mathcal{V}$, $a \notin \mathcal{T}$. Intuitively, submodularity refers to the law of diminishing return: the marginal gain of $f(\mathcal{P})$ by adding an element a to $\mathcal{P}$ diminishes as the size of $\mathcal{P}$ increases for all P. The submodular maximization problem subjected to the cardinality constraint can be formulated as follows: $\mathcal{P}^* = \underset{\mathcal{P} \subseteq \mathcal{U}, |\mathcal{P}| = n}{\arg\max}\ f(\mathcal{P})$. Submodular optimization problems are known to be NP-hard [78]. However, there exist greedy algorithms with a linear complexity $\mathcal{O}\left(|\mathcal{U}||\mathcal{P}|\right)$ [79], which achieve atleast a $(1-1/e)$-factor approximation of the optimal solution.

2.5 Product codebook design for beamforming

To enable the CSIT for beamforming (precoding) through codebooks, a quantization scheme for quantizing the optimal beamformer (precoder) and a design criterion for constructing the respective codebooks are necessary. An efficient iterative beamformer (precoder) codebook design method based on vector quantization of the space $\mathbb{C}^{M_t \times 1}$ ($\mathbb{C}^{M_t \times \mathtt{r}}$) is proposed in [80], [1]. The complexity of the VQ algorithm increases (exact complexity analysis is shown in Sec. 2.7) with increasing Tx antennas that makes the design algorithm impractical in massive MIMO regime.

In this section, we focus on designing beamformer codebooks for the system model described in 2.3.1 i.e., $M_r = 1$ and rank $- 1$ transmission. The UPA structure of the Tx antenna naturally allows us to represent the channel $\mathbf{H} \in \mathbb{C}^{1 \times M_t}$ as a matrix channel $\tilde{\mathbf{H}} \in \mathbb{C}^{M_v \times M_h}$ whose (i, j)-th element corresponds to the channel between the antenna element at the i-th row and j-th column of the UPA and the receive antenna. We first describe the design of unquantized beamformer for a given $\mathbf{H}$ and then provide a design method to construct the product codebooks for beamformer.

2.5.1 Unquantized beamformer design

The relation between the UPA matrix channel $\tilde{\mathbf{H}} \in \mathbb{C}^{M_v \times M_h}$ and $\mathbf{H} \in \mathbb{C}^{1 \times M_t}$ is $\mathbf{H}^T = \mathrm{vec}(\tilde{\mathbf{H}}^T)$. The SVD of $\tilde{\mathbf{H}}$ is $\tilde{\mathbf{H}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^H$, where $\tilde{\mathbf{U}} \in \mathcal{U}_{M_v}$, $\tilde{\mathbf{V}} \in \mathcal{U}_{M_h}$, $\tilde{\mathbf{\Sigma}}$ is the $M_v \times M_h$ rectangular diagonal matrix with i-th largest singular value $\tilde{\sigma}_i$ at the entry (i,i). Then we have

$$\mathbf{H}^T = \mathrm{vec}(\tilde{\mathbf{H}}^T) = \mathrm{vec}(\tilde{\mathbf{V}}^*\tilde{\mathbf{\Sigma}}\tilde{\mathbf{U}}^T) = \mathrm{vec}\left(\sum_{i=1}^{\mathrm{rank}(\tilde{\mathbf{H}})} \tilde{\sigma}_i \tilde{\mathbf{v}}_i^* \tilde{\mathbf{u}}_i^T\right) = \sum_{i=1}^{\mathrm{rank}(\tilde{\mathbf{H}})} \tilde{\sigma}_i \tilde{\mathbf{u}}_i \otimes \tilde{\mathbf{v}}_i^*. \tag{2.5}$$

Thus, we can represent $\mathbf{H}$ as the linear combination of $\tilde{\mathbf{u}}_i^T \otimes \tilde{\mathbf{v}}_i^H$ scaled with $\tilde{\sigma}_i$ as $\mathbf{H} = \sum_{i=1}^{\mathrm{rank}(\tilde{\mathbf{H}})} \tilde{\sigma}_i \tilde{\mathbf{u}}_i^T \otimes \tilde{\mathbf{v}}_i^H$. In order to facilitate product beamformer codebook construction, we approximate the channel $\mathbf{H}$ with its dominant direction, i.e., $\tilde{\mathbf{u}}_1^T \otimes \tilde{\mathbf{v}}_1^H$, which is called the rank -1 approximation. The approximated channel $\bar{\mathbf{H}}$ is given as $\mathbf{H} \approx \bar{\mathbf{H}} = \tilde{\sigma}_1 \tilde{\mathbf{u}}_1^T \otimes \tilde{\mathbf{v}}_1^H$. Let $\mathbf{f} \in \mathcal{U}(M_t, 1)$ be a beamformer for $\bar{\mathbf{H}}$, then the KP form of $\bar{\mathbf{H}}$ naturally leads us to the idea of using $\mathbf{f}$ of the form $\mathbf{f} = \mathbf{f}_v \otimes \mathbf{f}_h$ where $\mathbf{f}_v \in \mathcal{U}(M_v, 1)$, $\mathbf{f}_h \in \mathcal{U}(M_h, 1)$. The beamforming gain $\Gamma(\bar{\mathbf{H}}, \mathbf{f})$ can now be simplified as $\Gamma(\bar{\mathbf{H}}, \mathbf{f}) = \left\|\bar{\mathbf{H}}\mathbf{f}\right\|_2^2 = \left\|\tilde{\sigma}_1(\tilde{\mathbf{u}}_1^T \otimes \tilde{\mathbf{v}}_1^H)(\mathbf{f}_v \otimes \mathbf{f}_h)\right\|_2^2 = \tilde{\sigma}_1^2 \left\|\tilde{\mathbf{u}}_1^T \mathbf{f}_v\right\|_2^2 \left\|\tilde{\mathbf{v}}_1^H \mathbf{f}_h\right\|_2^2 = \tilde{\sigma}_1^2 \left|\tilde{\mathbf{u}}_1^T \mathbf{f}_v\right|^2 \left|\tilde{\mathbf{v}}_1^H \mathbf{f}_h\right|^2$. The optimal beamformer $\hat{\mathbf{f}}$ for $\bar{\mathbf{H}}$ that maximizes $\Gamma(\bar{\mathbf{H}}, \mathbf{f})$ can be simplified as

$$\begin{aligned}
\hat{\mathbf{f}} &= \underset{\mathbf{f} \in \mathcal{U}(M_t,1)}{\arg\max}\, \Gamma(\bar{\mathbf{H}}, \mathbf{f}) \\
&= \underset{\substack{\mathbf{f}_v \in \mathcal{U}(M_v,1) \\ \mathbf{f}_h \in \mathcal{U}(M_h,1)}}{\arg\max}\, \left|\tilde{\mathbf{u}}_1^T \mathbf{f}_v\right|^2 \left|\tilde{\mathbf{v}}_1^H \mathbf{f}_h\right|^2 \\
&= \underset{\mathbf{f}_v \in \mathcal{U}(M_v,1)}{\arg\max}\, \left|\tilde{\mathbf{u}}_1^T \mathbf{f}_v\right|^2 \otimes \underset{\mathbf{f}_h \in \mathcal{U}(M_h,1)}{\arg\max}\, \left|\tilde{\mathbf{v}}_1^H \mathbf{f}_h\right|^2 \\
&= \hat{\mathbf{f}}_v \otimes \hat{\mathbf{f}}_h,
\end{aligned} \tag{2.6}$$

where $\hat{\mathbf{f}}_v = \underset{\mathbf{f}_v \in \mathcal{U}(M_v,1)}{\arg\max}\, \left|\tilde{\mathbf{u}}_1^T \mathbf{f}_v\right|^2$, $\hat{\mathbf{f}}_h = \underset{\mathbf{f}_h \in \mathcal{U}(M_h,1)}{\arg\max}\, \left|\tilde{\mathbf{v}}_1^H \mathbf{f}_h\right|^2$ and the maximum beamforming gain is $\Gamma(\bar{\mathbf{H}}, \hat{\mathbf{f}}) = \tilde{\sigma}_1^2$. Clearly, a solution for the optimal beamformer $\hat{\mathbf{f}} = \hat{\mathbf{f}}_v \otimes \hat{\mathbf{f}}_h$ in (2.6) is given by the dominant singular vectors of the approximated channel $\tilde{\mathbf{H}}$, i.e., $\hat{\mathbf{f}}_v = \tilde{\mathbf{u}}_1^*$, $\hat{\mathbf{f}}_h = \tilde{\mathbf{v}}_1$ and thus $\hat{\mathbf{f}} = \tilde{\mathbf{u}}_1^* \otimes \tilde{\mathbf{v}}_1$.

2.5.2 Quantized beamformer design

We define the normalized beamforming gain $\Gamma_n(\bar{\mathbf{H}}, \mathbf{f})$ and the loss in $\Gamma_n(\bar{\mathbf{H}}, \mathbf{f})$, i.e., $L(\bar{\mathbf{H}}, \mathbf{f})$ obtained with an arbitrary KP beamformer $\mathbf{f} = \mathbf{f}_v \otimes \mathbf{f}_h$ as

$$\Gamma_n(\bar{\mathbf{H}}, \mathbf{f}) := \frac{\Gamma(\bar{\mathbf{H}}, \mathbf{f})}{\Gamma(\bar{\mathbf{H}}, \hat{\mathbf{f}})} = \frac{\Gamma(\bar{\mathbf{H}}, \mathbf{f})}{\tilde{\sigma}_1^2} \overset{(a)}{=} |\tilde{\mathbf{u}}_1^T \mathbf{f}_v|^2 \ |\tilde{\mathbf{v}}_1^H \mathbf{f}_h|^2, L(\bar{\mathbf{H}}, \mathbf{f}) := 1 - \Gamma_n(\bar{\mathbf{H}}, \mathbf{f}),$$

where $\hat{\mathbf{f}}$ is the optimal unquantized KP beamformer for a given $\bar{\mathbf{H}}$, (a) comes from (2.6). The KP structure of the beamformer $\mathbf{f}$ motivates to employ separate codebooks $\mathcal{F}_v \subseteq \mathcal{U}(M_v, 1)$, $\mathcal{F}_h \subseteq \mathcal{U}(M_h, 1)$ for horizontal and vertical dimensions which enables to design product codebooks by clustering in lower dimensional spaces. The product codebook for the KP beamformer $\mathbf{f} = \mathbf{f}_v \otimes \mathbf{f}_h$ formed by the codebooks $\mathcal{F}_v$, $\mathcal{F}_h$ is represented as $\mathcal{F} = \mathcal{F}_v \times \mathcal{F}_h$. The loss in normalized beamforming gain with $\mathbf{f}$ can be bounded as $L(\bar{\mathbf{H}}, \mathbf{f}) = 1 - \Gamma_n(\bar{\mathbf{H}}, \mathbf{f}) = 1 - |(\tilde{\mathbf{u}}_1^T \otimes \tilde{\mathbf{v}}_1^H)(\mathbf{f}_v \otimes \mathbf{f}_h)|^2 \leq$

$$\begin{aligned}
&2 \ \left(1 - |(\tilde{\mathbf{u}}_1^T \otimes \tilde{\mathbf{v}}_1^H)(\mathbf{f}_v \otimes \mathbf{f}_h)|\right) \leq 2 \min_{\theta,\phi} \left(\left\|(e^{j\theta}\tilde{\mathbf{u}}_1^* \otimes e^{j\phi}\tilde{\mathbf{v}}_1) - (\mathbf{f}_v \otimes \mathbf{f}_h)\right\|\right) \\
&\leq 2 \min_{\theta,\phi} \left(\left\|e^{j\theta}\tilde{\mathbf{u}}_1^*\right\|_2 \left\|e^{j\phi}\tilde{\mathbf{v}}_1 - \mathbf{f}_h\right\|_2 + \left\|e^{j\theta}\tilde{\mathbf{u}}_1^* - \mathbf{f}_v\right\|_2 \left\|e^{j\phi}\mathbf{f}_h\right\|_2\right) \\
&= 2 \min_{\theta,\phi} \left(\left\|e^{j\phi}\tilde{\mathbf{v}}_1 - \mathbf{f}_h\right\|_2 + \left\|e^{j\theta}\tilde{\mathbf{u}}_1^* - \mathbf{f}_v\right\|_2\right) = 2 \ \left[(1 - |\tilde{\mathbf{v}}_1^H \mathbf{f}_h|)^{1/2} + (1 - |\tilde{\mathbf{u}}_1^T \mathbf{f}_v|)^{1/2}\right] \\
&\leq 2 \ \left[(1 - |\tilde{\mathbf{v}}_1^H \mathbf{f}_h|^2) + (1 - |\tilde{\mathbf{u}}_1^T \mathbf{f}_v|^2)\right] := L_{\text{ub}}(\bar{\mathbf{H}}, \mathbf{f}).
\end{aligned}$$

In $L_{\text{ub}}(\bar{\mathbf{H}}, \mathbf{f})$ defined above, for any angles $\alpha, \beta \in [0, 2\pi)$, we have $(1 - |\tilde{\mathbf{v}}_1^H \mathbf{f}_h|^2) + (1 - |\tilde{\mathbf{u}}_1^T \mathbf{f}_v|^2) = (1 - |\tilde{\mathbf{v}}_1^H \mathbf{f}_h e^{j\alpha}|^2) + (1 - |\tilde{\mathbf{u}}_1^T \mathbf{f}_v e^{j\beta}|^2)$. The rotational invariance of $L_{\text{ub}}(\bar{\mathbf{H}}, \mathbf{f})$ from the above equation implies that $\mathbf{f}_v$, $\mathbf{f}_h$ are points on a GM i.e., $\mathbf{f}_v \in \mathcal{G}(M_v, 1)$, $\mathbf{f}_h \in \mathcal{G}(M_h, 1)$ and thus the respective codebooks $\mathcal{F}_v \subseteq \mathcal{G}(M_v, 1)$, $\mathcal{F}_h \subseteq \mathcal{G}(M_h, 1)$. From the definition of chordal distance $d_c(\cdot)$, the upper bound of $L(\bar{\mathbf{H}}, \mathbf{f})$ can also be written as

$$L_{\text{ub}}(\bar{\mathbf{H}}, \mathbf{f}) = (1 - |\tilde{\mathbf{v}}_1^H \mathbf{f}_h|^2) + (1 - |\tilde{\mathbf{u}}_1^T \mathbf{f}_v|^2) = d_c^2(\tilde{\mathbf{u}}_1^*, \mathbf{f}_v) + d_c^2(\tilde{\mathbf{v}}_1, \mathbf{f}_h).$$

Remark 2.4. The upper bound of the loss in normalized beamforming gain i.e., $L_{\text{ub}}(\bar{\mathbf{H}}, \mathbf{f})$ obtained by beamforming with $\mathbf{f} = \mathbf{f}_v \otimes \mathbf{f}_h$ instead of the optimal unquantized beamformer

$\hat{\mathbf{f}} = \tilde{\mathbf{u}}_1^* \otimes \tilde{\mathbf{v}}_1$ for a given $\mathbf{H}$ is equivalent to the squared distance between the points $(\tilde{\mathbf{u}}_1^*, \tilde{\mathbf{v}}_1)$ and $(\mathbf{f}_v, \mathbf{f}_h)$ on the CPM $\mathcal{G}^{\times}((M_v, M_h), (1,1))$ i.e., $L_{\text{ub}}(\bar{\mathbf{H}}, \mathbf{f}) = d_c^2(\tilde{\mathbf{u}}_1^*, \mathbf{f}_v) + d_c^2(\tilde{\mathbf{v}}_1, \mathbf{f}_h) = d_c^2((\tilde{\mathbf{u}}_1^*, \tilde{\mathbf{v}}_1), (\mathbf{f}_v, \mathbf{f}_h))$.

2.5.3 Product codebook design criterion

To measure the average distortion introduced by the quantization with the codebook $\mathcal{F} = \mathcal{F}_v \times \mathcal{F}_h$, we use the upper bound of the average loss in normalized beamforming gain $L_{\text{ub}}(\bar{\mathbf{H}}, \mathbf{f})$ and define $L_{\text{ub}}(\mathcal{F})$ as $L_{\text{ub}}(\mathcal{F}) := \mathbb{E}_{\bar{\mathbf{H}}}\left[L_{\text{ub}}(\bar{\mathbf{H}}, \mathbf{f})\right] = \mathbb{E}_{\tilde{\mathbf{u}}_1, \tilde{\mathbf{v}}_1}\left[L_{\text{ub}}(\bar{\mathbf{H}}, \mathbf{f})\right]$.

Definition 2.5 (Grassmann product codebook for beamforming)**.** Under rank -1 approximation of the channel, $\mathbf{H} \approx \bar{\mathbf{H}} = \tilde{\sigma}_1 \tilde{\mathbf{u}}_1^T \otimes \tilde{\mathbf{v}}_1^H$, the Grassmann product codebook $\hat{\mathcal{F}} = \hat{\mathcal{F}}_v \times \hat{\mathcal{F}}_h$ for beamforming is the one that minimizes $L_{\text{ub}}(\mathcal{F})$ for a given feedback bit allocation $[B_v, B_h]$ where $|\hat{\mathcal{F}}_v| = 2^{B_v}$, $|\hat{\mathcal{F}}_h| = 2^{B_h}$.

We will now state the method to construct the Grassmann product codebook $\hat{\mathcal{F}}$ as follows.

Lemma 2.6. *The Grassmann product codebook $\hat{\mathcal{F}} = \hat{\mathcal{F}}_v \times \hat{\mathcal{F}}_h$ as defined in Def. 2.5 can be constructed using the set of centroids $\mathcal{F}_v^K, \mathcal{F}_h^K$ obtained from the independent K-means clustering of the optimal KP beamformers $\tilde{\mathbf{u}}_1^*$, $\tilde{\mathbf{v}}_1$ on $\mathcal{G}(M_v, 1)$, $\mathcal{G}(M_h, 1)$ with $K = 2^{B_v}, 2^{B_h}$, respectively.*

Proof. From Def. 2.5,

$$\begin{aligned}
\hat{\mathcal{F}} = \hat{\mathcal{F}}_v \times \hat{\mathcal{F}}_h &= \arg\min_{\mathcal{F}_v, \mathcal{F}_h} L_{\text{ub}}(\mathcal{F}) \\
&= \arg\min_{\mathcal{F}_v, \mathcal{F}_h} \mathbb{E}_{\tilde{\mathbf{u}}_1, \tilde{\mathbf{v}}_1}\left[\min_{\substack{\mathbf{f}_v \in \mathcal{F}_v \\ \mathbf{f}_v \in \mathcal{F}_v}} \left(d_c^2(\tilde{\mathbf{u}}_1^*, \mathbf{f}_v) + d_c^2(\tilde{\mathbf{v}}_1, \mathbf{f}_h)\right)\right] \\
&= \arg\min_{\mathcal{F}_v, \mathcal{F}_h} \mathbb{E}_{\tilde{\mathbf{u}}_1}\left[\min_{\mathbf{f}_v \in \mathcal{F}_v} d_c^2(\tilde{\mathbf{u}}_1^*, \mathbf{f}_v)\right] + \mathbb{E}_{\tilde{\mathbf{v}}_1}\left[\min_{\mathbf{f}_h \in \mathcal{F}_h} d_c^2(\tilde{\mathbf{v}}_1, \mathbf{f}_h)\right].
\end{aligned}$$

This objective can be minimized if both the terms in the summation are independently minimized. Therefore the codebooks $\hat{\mathcal{F}}_v, \hat{\mathcal{F}}_h$ that form the Grassmann product codebook $\hat{\mathcal{F}}$

are given as

$$\hat{\mathcal{F}}_v = \underset{\substack{\mathcal{F}_v \subseteq \mathcal{G}(M_v,1) \\ |\mathcal{F}_v|=2^{B_v}}}{\arg\min}\, \mathbb{E}_{\tilde{\mathbf{u}}_1}\left[\min_{\mathbf{f}_v \in \mathcal{F}_v} d_c^2(\tilde{\mathbf{u}}_1^*, \mathbf{f}_v)\right], \quad \hat{\mathcal{F}}_h = \underset{\substack{\mathcal{F}_h \subseteq \mathcal{G}(M_h,1) \\ |\mathcal{F}_h|=2^{B_h}}}{\arg\min}\, \mathbb{E}_{\tilde{\mathbf{v}}_1}\left[\min_{\mathbf{f}_h \in \mathcal{F}_h} d_c^2(\tilde{\mathbf{v}}_1, \mathbf{f}_h)\right]. \tag{2.7}$$

Comparing the general Grassmannian K-means objective in (2.4) in Sec. 2.4.2 and the above codebook design criteria, $\hat{\mathcal{F}}_h, \hat{\mathcal{F}}_v$ can be found by the K-means clustering algorithm for $[K, n, k] = [2^{B_h}, M_h, 1], [2^{B_v}, M_v, 1]$ respectively, in Alg. 1. Therefore we have $\hat{\mathcal{F}}_h = \mathcal{F}_h^K, \hat{\mathcal{F}}_v = \mathcal{F}_v^K$, and $\hat{\mathcal{F}} = \hat{\mathcal{F}}_v \times \hat{\mathcal{F}}_h = \mathcal{F}_v^K \times \mathcal{F}_h^K$ and the criteria for the choosing the optimal beamformer $\hat{\mathbf{f}}$ from $\hat{\mathcal{F}}_v$, $\hat{\mathcal{F}}_h$ for a given $\mathbf{H}$ as $\hat{\mathbf{f}}_v = \underset{\mathbf{f} \in \hat{\mathcal{F}}_v}{\arg\min}\, d_c^2(\tilde{\mathbf{u}}^*, \mathbf{f}), \hat{\mathbf{f}}_h = \underset{\mathbf{f} \in \hat{\mathcal{F}}_h}{\arg\min}\, d_c^2(\tilde{\mathbf{v}}, \mathbf{f}), \hat{\mathbf{f}} = \hat{\mathbf{f}}_v \otimes \hat{\mathbf{f}}_h$. □

2.5.4 Codebook construction

From Lem. 2.6, it is possible to perform K-means clustering independently on $\mathcal{G}(M_v, 1)$, $\mathcal{G}(M_h, 1)$ and construct the product codebook with reduced complexity. We assume a stationary distribution of the channel for a given coverage area of a Tx. In order to construct the Grassmann product codebook for beamforming as defined in Def. 2.5, we construct $\mathcal{H} = \{\mathbf{H}\}$, a set of channel realizations sampled for different user locations. The available channel dataset $\mathcal{H}$ is split into training and testing datasets, $\mathcal{H}_{\text{train}}$ and $\mathcal{H}_{\text{test}}$ for generating beamformer codebooks and evaluating their performance respectively. We assume that the size of the training set is large enough so that the sampling distribution closely approximates the original distribution. The training procedure yields the optimal product codebook whose performance is evaluated by measuring the average normalized beamforming gain for the channel realizations in the test set $\mathcal{H}_{\text{test}}$. The training and testing procedure of the proposed product codebook design for a given set of channel realizations is summarized in the following remark.

Remark 2.7. For a given $\mathcal{H}_{\text{train}}$ and $\mathcal{H}_{\text{test}}$, the Grassmann product codebook for beamforming $\hat{\mathcal{F}} = \hat{\mathcal{F}}_v \times \hat{\mathcal{F}}_h$ is obtained by the procedure BFTrain($\mathcal{H}_{\text{train}}$,$[B_v, B_h]$) and the performance of the codebook $\hat{\mathcal{F}}$ is evaluated by the procedure BFTest($\mathcal{H}_{\text{test}}$,$[\hat{\mathcal{F}}_v, \hat{\mathcal{F}}_h]$) as outlined in Alg. 2, where B_v, B_h are the number of bits used to encode $\tilde{\mathbf{u}}_1^*$, $\tilde{\mathbf{v}}_1$ respectively.

2.6 Product codebook design for precoding

In this section, we present a product codebook design method for rank $-\,\mathtt{r}$ ($M_h > \mathtt{r}, M_v > \mathtt{r}$) transmission in a MIMO system with $M_r > 1$ as described in Sec. 2.3.2. Similar to the beamformer codebook design, we explore the UPA structure of the Tx antenna and tensor representation of the channel to find reduced complexity precoder codebooks. We introduce this scheme as follows.

2.6.1 HOOI-based unquantized precoder design

Tucker decomposition of the channel

The uniform planar structure of the Tx antenna permits a natural representation of the matrix channel $\mathbf{H}$ as tensor $\mathcal{H}$ where $\mathcal{H} \in \mathbb{C}^{M_r \times M_h \times M_v}$ (as demonstrated in Fig. 2.2) and $\mathcal{H}_{ijk}$ represents the channel between the antenna element at k-th row and j-th column of the UPA at the Tx and the i-th antenna at the Rx. Although one can rearrange $\mathbf{H}$ in tensors of arbitrary dimensions, in the rest of this chapter, we will be focusing on the tensors of dimensions $M_r \times M_h \times M_v$. From the tensor representation of channel $\mathbf{H}$ as $\mathcal{H}$, we have that $\mathbf{H}$ is equivalent to the mode-1 unfolding of $\mathcal{H}$ i.e., $\mathbf{H} = \mathbf{H}_{(1)}$ and TD of $\mathcal{H}$ is expressed as

$$\mathcal{H} = \mathcal{G} \times_1 \mathbf{B} \times_2 \mathbf{A}^{(1)} \times_3 \mathbf{A}^{(2)}, \mathbf{H} = \mathbf{H}_{(1)} = \mathbf{B}\mathbf{G}_{(1)}(\mathbf{A}^{(2)} \otimes \mathbf{A}^{(1)})^T = \mathbf{B}\mathbf{G}_{(1)}\mathbf{A}^H,$$

where $\mathcal{G} \in \mathbb{C}^{M_r \times M_h \times M_v}$ is the core tensor, $\mathbf{B} \in \mathcal{U}_{M_r}$, $\mathbf{A}^{(1)} \in \mathcal{U}_{M_h}$, $\mathbf{A}^{(2)} \in \mathcal{U}_{M_v}$,$\mathbf{A} = \left(\mathbf{A}^{(2)} \otimes \mathbf{A}^{(1)}\right)^*$. The best rank $-\,(M_r, \mathtt{r}, \mathtt{r})$ approximation of $\mathcal{H}$ i.e., $\bar{\mathcal{H}}$ obtained as described in Sec. 2.4.1 is

$$\mathcal{H} \approx \bar{\mathcal{H}} = \bar{\mathcal{G}} \times_1 \bar{\mathbf{B}} \times_2 \bar{\mathbf{A}}^{(1)} \times_3 \bar{\mathbf{A}}^{(2)}, \bar{\mathbf{H}} = \bar{\mathbf{H}}_{(1)} = \bar{\mathbf{B}}\bar{\mathbf{G}}_{(1)}(\bar{\mathbf{A}}^{(2)} \otimes \bar{\mathbf{A}}^{(1)})^T = \bar{\mathbf{B}}\bar{\mathbf{G}}_{(1)}\bar{\mathbf{A}}^H, \quad (2.8)$$

where $\bar{\mathcal{G}} \in \mathbb{C}^{M_r \times \mathtt{r} \times \mathtt{r}}$ is the core tensor, $\bar{\mathbf{B}} \in \mathcal{U}_{M_r}$, $\bar{\mathbf{A}}^{(1)} \in \mathcal{U}(M_h, \mathtt{r})$, $\bar{\mathbf{A}}^{(2)} \in \mathcal{U}(M_v, \mathtt{r})$, $\bar{\mathbf{A}} = \left(\bar{\mathbf{A}}^{(2)} \otimes \bar{\mathbf{A}}^{(1)}\right)^*$. Here, $\bar{\mathbf{H}}$ is the mode-1 unfolding of the $\bar{\mathcal{H}}$ and $\bar{\mathbf{A}}^{(1)}$, $\bar{\mathbf{A}}^{(2)}$ are the principal components of $\bar{\mathcal{H}}$ in the horizontal, vertical dimensions, respectively.

From the SVD of channel $\mathbf{H}$, the eigenvalue σ_i^2 represents the power of the channel along the corresponding eigen-direction $\mathbf{v}_i$. We recall that in SVD-based precoding, an

optimal precoder for rank $-\,\mathtt{r}$ transmission is formed by dominant $\mathtt{r}$ columns of $\mathbf{V}$ i.e., the columns of $\mathbf{V}$ corresponding to the dominant $\mathtt{r}$ singular values. The basic principle of the proposed HOOI-based precoder design technique is also to identify the dominant $\mathtt{r}$ columns of $\bar{\mathbf{A}} = \left(\bar{\mathbf{A}}^{(2)} \otimes \bar{\mathbf{A}}^{(1)}\right)^*$ in (2.8) that maximize the mutual information when the rank $-\,\mathtt{r}$ matrix formed by the $\mathtt{r}$ columns is used as precoder for transmission. However, identifying the dominant $\mathtt{r}$ columns of $\bar{\mathbf{A}}$ out of $\mathtt{r}^2$ columns is not immediately clear, since unlike the singular matrix $\mathbf{\Sigma}$, $\bar{\mathbf{G}}_{(1)}$ is not a diagonal matrix. Let $\mathcal{C} \subset \{1, \cdots, \mathtt{r}^2\}$ with $|\mathcal{C}| = \mathtt{r}$ be a set of column indices and $\mathcal{C}_o$ be the set of column indices of dominant $\mathtt{r}$ columns of $\bar{\mathbf{A}}$ and $\bar{\mathbf{A}}_{\mathcal{C}} = \bar{\mathbf{A}}(:, \mathcal{C})$. The construction of $\mathcal{C}_o$ and the proposed unquantized precoder for a given $\mathbf{H}$ are outlined as follows.

Proposition 2.8. *For a given* $\mathbf{H}$*, the proposed unquantized precoder for* rank$-\mathtt{r}$ *transmission is formed by the dominant* $\mathtt{r}$ *columns of* $\bar{\mathbf{A}}$ *i.e.,* $\bar{\mathbf{A}}_{\mathcal{C}_o}$*, where* $\mathcal{C}_o$ *is the set of column indices of dominant* $\mathtt{r}$ *columns of* $\bar{\mathbf{A}}$ *that maximizes the mutual information* $R(\mathbf{H}, \bar{\mathbf{A}}_{\mathcal{C}_o})$*.*

The mutual information obtained with the precoder $\bar{\mathbf{A}}_{\mathcal{C}}$ for a given $\mathbf{H}$ is $R(\mathbf{H}, \bar{\mathbf{A}}_{\mathcal{C}}) = \log\det\left(\mathbf{I} + \rho_t \bar{\mathbf{A}}_{\mathcal{C}}^H \mathbf{H}^H \mathbf{H} \bar{\mathbf{A}}_{\mathcal{C}}\right)$. Then, $\mathcal{C}_o$ is obtained from the following optimization problem:

$$\begin{aligned} \mathcal{C}_o = \underset{\mathcal{C}\subset\{1,\cdots,\mathtt{r}^2\},|\mathcal{C}|=\mathtt{r}}{\arg\max}\; R(\mathbf{H}, \bar{\mathbf{A}}_{\mathcal{C}}) &= \underset{\mathcal{C}\subset\{1,\cdots,\mathtt{r}^2\},|\mathcal{C}|=\mathtt{r}}{\arg\max}\; \log\det\left(\mathbf{I} + \rho_t \bar{\mathbf{A}}_{\mathcal{C}}^H \mathbf{H}^H \mathbf{H} \bar{\mathbf{A}}_{\mathcal{C}}\right) \\ &= \underset{\mathcal{C}\subset\{1,\cdots,\mathtt{r}^2\},|\mathcal{C}|=\mathtt{r}}{\arg\max}\; \log\det\left(\mathbf{I} + \rho_t (\mathbf{H}\bar{\mathbf{A}}_{\mathcal{C}})^H (\mathbf{H}\bar{\mathbf{A}}_{\mathcal{C}})\right). \end{aligned} \tag{2.9}$$

The above optimization is equivalent to choosing the appropriate $\mathtt{r}$ columns out of $\mathtt{r}^2$ columns of $\mathbf{H}\bar{\mathbf{A}}$ and the exact solution $\mathcal{C}_o$ is obtained by maximizing $R(\mathbf{H}, \bar{\mathbf{A}}_{\mathcal{C}})$ over all the possible $\mathtt{r}$ element sets for $\mathcal{C}$. Interestingly, $R(\mathbf{H}, \bar{\mathbf{A}}_{\mathcal{C}})$ is a monotone submodular function [81] and hence (2.9) is a monotone submodular maximization problem with cardinality constraints (see Sec. 2.4.3). Since this problem is NP hard [81], we provide a greedy algorithm in Alg. 3 for the design of $\mathcal{C}_o$.

Lemma 2.9. *The mutual information obtained with the proposed unquantized precoder* $\bar{\mathbf{A}}_{\mathcal{C}_o}$ *is* $R(\mathbf{H}, \bar{\mathbf{A}}_{\mathcal{C}_o}) = R(\bar{\mathbf{H}}, \bar{\mathbf{A}}_{\mathcal{C}_o}) = \log\det\left(\mathbf{I} + \rho_t \bar{\mathbf{G}}_{(1),\mathcal{C}_o}^H \bar{\mathbf{G}}_{(1),\mathcal{C}_o}\right)$.

Proof: Consider the equivalent channel $\mathbf{H}_{eq}$ associated with the precoder $\bar{\mathbf{A}}_{\mathcal{C}}$ and $\mathbf{H}$. Then, we have $\mathbf{H}_{eq}^H \mathbf{H}_{eq} = \bar{\mathbf{A}}_{\mathcal{C}}^H \mathbf{H}_{(1)}^H \mathbf{H}_{(1)} \bar{\mathbf{A}}_{\mathcal{C}} = \bar{\mathbf{A}}_{\mathcal{C}}^H \bar{\mathbf{H}}_{(1)}^H \bar{\mathbf{H}}_{(1)} \bar{\mathbf{A}}_{\mathcal{C}} = \bar{\mathbf{A}}_{\mathcal{C}}^H \bar{\mathbf{A}} \bar{\mathbf{G}}_{(1)}^H \bar{\mathbf{G}}_{(1)} \bar{\mathbf{A}}^H \bar{\mathbf{A}}_{\mathcal{C}} =$

$\bar{\mathbf{G}}^H_{(1),\mathcal{C}}\bar{\mathbf{G}}_{(1),\mathcal{C}}$. From Alg. 3, the proposed unquantized precoder can be expressed as $\bar{\mathbf{A}}_{\mathcal{C}_o} = \textsc{DomCol}(\bar{\mathbf{A}}, \mathbf{H}, \mathtt{r})$ and thus the mutual information is $R(\mathbf{H}, \bar{\mathbf{A}}_{\mathcal{C}_o}) = \max_{\mathcal{C}\subseteq\{1,\cdots,\mathtt{r}^2\},|\mathcal{C}|=\mathtt{r}} R(\mathbf{H}, \bar{\mathbf{A}}_{\mathcal{C}}) = \log\det\left(\mathbf{I} + \rho_t \bar{\mathbf{A}}^H_{\mathcal{C}_o}\mathbf{H}^H\mathbf{H}\bar{\mathbf{A}}_{\mathcal{C}_o}\right) = \log\det\left(\mathbf{I} + \rho_t \bar{\mathbf{G}}^H_{(1),\mathcal{C}_o}\bar{\mathbf{G}}_{(1),\mathcal{C}_o}\right) = R(\bar{\mathbf{H}}, \bar{\mathbf{A}}_{\mathcal{C}_o})$. ■

In optimal precoding, the Tx requires the knowledge of $\bar{\mathbf{V}}$. Whereas, in HOOI-based precoding, the Tx requires the knowledge of $\bar{\mathbf{A}}_{\mathcal{C}_o}$ which is formed using $\bar{\mathbf{A}}^{(1)}$, $\bar{\mathbf{A}}^{(2)}$ and $\mathcal{C}_o$ as described in Lem. 2.9. As the channel realization $\mathbf{H}$ changes, $\bar{\mathbf{A}}^{(1)}$, $\bar{\mathbf{A}}^{(2)}$ change and $\mathcal{C}_o$ that forms the proposed precoder $\bar{\mathbf{A}}_{\mathcal{C}_o}$ also changes. Hence, for this scheme, $(\bar{\mathbf{A}}^{(1)}, \bar{\mathbf{A}}^{(2)}, \mathcal{C}_o)$ is the CSIT required for the construction of the precoder. However, due to the limited capacity of the feedback channel, this information needs to be quantized.

2.6.2 Quantized precoder design

In this section, we propose the design of quantized precoder and a loss in mutual information due to quantization for a given $\mathbf{H}$ that enable the design of product precoder codebooks, which are cartesian product of two lower dimensional codebooks. The KP structure of $\bar{\mathbf{A}} = (\bar{\mathbf{A}}^{(2)} \otimes \bar{\mathbf{A}}^{(1)})^*$ in the precoder $\bar{\mathbf{A}}_{\mathcal{C}_o}$ motivates to construct a rank $-\ \mathtt{r}$ precoder of the form $\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}$, where $Q(\bar{\mathbf{A}}) = \left(Q(\bar{\mathbf{A}}^{(2)}) \otimes Q(\bar{\mathbf{A}}^{(1)})\right)^*$, and $Q(\bar{\mathbf{A}}^{(1)}) \in \mathcal{U}(M_h, \mathtt{r})$, $Q(\bar{\mathbf{A}}^{(2)}) \in \mathcal{U}(M_v, \mathtt{r})$ are the quantized versions of $\bar{\mathbf{A}}^{(1)}$, $\bar{\mathbf{A}}^{(2)}$, respectively, $\mathcal{C}_Q$ is a set of $\mathtt{r}$ column indices of $Q(\bar{\mathbf{A}})$. On the similar lines of design of unquantized precoder in Prop. 2.8, $\mathcal{C}_Q$ is designed to maximize the mutual information with the precoder $\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}$. We formally describe the construction of the optimal quantized precoder in the following proposition.

Proposition 2.10. *Let $Q(\bar{\mathbf{A}}^{(1)}) \in \mathcal{U}(M_h, \mathtt{r})$ and $Q(\bar{\mathbf{A}}^{(2)}) \in \mathcal{U}(M_v, \mathtt{r})$ be the quantized representations of $\bar{\mathbf{A}}^{(1)}$ and $\bar{\mathbf{A}}^{(2)}$ respectively. Then, for a given $\mathbf{H}$, the proposed quantized precoder for* rank $-\ \mathtt{r}$ *transmission is formed by the dominant $\mathtt{r}$ columns of $Q(\bar{\mathbf{A}}) = \left(Q(\bar{\mathbf{A}}^{(2)}) \otimes Q(\bar{\mathbf{A}}^{(1)})\right)^*$ i.e., $\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}$, where $\mathcal{C}_Q$ is the set of column indices of dominant $\mathtt{r}$ columns of $Q(\bar{\mathbf{A}})$ which maximizes $R\left(\mathbf{H}, \left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}\right)$.*

The mutual information obtained with the precoder $\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}}$ for a given $\mathbf{H}$ is $R\left(\mathbf{H}, \left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}}\right) =$

$\log\det\left(\mathbf{I}+\rho_t\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}}^{H}\mathbf{H}^H\mathbf{H}\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}}\right)$. From Prop. 2.10, $\mathcal{C}_Q$ is obtained as

$$\mathcal{C}_Q=\underset{\substack{\mathcal{C}\subseteq\{1,\cdots,\mathtt{r}^2\}\\|\mathcal{C}|=\mathtt{r}}}{\arg\max}\, R\left(\mathbf{H},\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}}\right)=\underset{\substack{\mathcal{C}\subseteq\{1,\cdots,\mathtt{r}^2\}\\|\mathcal{C}|=\mathtt{r}}}{\arg\max}\log\det\left(\mathbf{I}+\rho_t\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}}^{H}\mathbf{H}^H\mathbf{H}\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}}\right). \tag{2.10}$$

The above optimization corresponds to maximizing a monotone submodular function with cardinality constraints similar to (2.9). The exact solution for $\mathcal{C}_Q$ is obtained by maximizing $R\left(\mathbf{H},\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}}\right)$ over all the possible $\mathtt{r}$ element sets for $\mathcal{C}$ which is NP-hard to determine. Thus, the proposed optimal quantized precoder can be expressed as $\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}=\textsc{DomCol}\left(Q(\bar{\mathbf{A}}),\mathbf{H},\mathtt{r}\right)$ (refer to Alg. 3). With the quantized principal components $Q(\bar{\mathbf{A}}^{(1)})$, $Q(\bar{\mathbf{A}}^{(2)})$ and $\mathcal{C}_Q$, the Tx is able to construct $\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}$ for precoding.

To measure the average loss in mutual information due to the limited capacity of the feedback channel, we first define a loss in mutual information associated with an arbitrary precoder $\mathbf{F}\in\mathcal{U}(M_t,\mathtt{r})$ for a given $\mathbf{H}$ as $L(\bar{\mathbf{H}},\mathbf{F}):=R(\bar{\mathbf{H}},\bar{\mathbf{A}}_{\mathcal{C}_o})-R(\bar{\mathbf{H}},\mathbf{F})$ where

$$\begin{aligned}R(\bar{\mathbf{H}},\mathbf{F})&=\log\det\left(\mathbf{I}+\rho_t\mathbf{F}^H\bar{\mathbf{H}}^H\bar{\mathbf{H}}\mathbf{F}\right)=\log\det\left(\mathbf{I}+\rho_t\mathbf{F}^H\bar{\mathbf{A}}\bar{\mathbf{G}}_{(1)}^H\bar{\mathbf{G}}_{(1)}\bar{\mathbf{A}}^H\mathbf{F}\right)\\&\gtrsim\log\det\left(\mathbf{I}+\rho_t\mathbf{F}^H\bar{\mathbf{A}}_{\mathcal{C}_o}\bar{\mathbf{G}}_{(1),\mathcal{C}_o}^H\bar{\mathbf{G}}_{(1),\mathcal{C}_o}\bar{\mathbf{A}}_{\mathcal{C}_o}^H\mathbf{F}\right):=R_{\text{lb}}(\bar{\mathbf{H}},\mathbf{F}).\end{aligned} \tag{2.11}$$

For concise notation let $\bar{\mathbf{G}}_{(1),\mathcal{C}_o}^H\bar{\mathbf{G}}_{(1),\mathcal{C}_o}=\bar{\mathbf{\Lambda}}_{\mathcal{C}_o}$, then

$$R_{\text{lb}}(\bar{\mathbf{H}},\mathbf{F})=\log\det\left(\mathbf{I}+\rho_t\bar{\mathbf{\Lambda}}_{\mathcal{C}_o}\right)+\log\det\left[\mathbf{I}-(\mathbf{I}+\rho_t\bar{\mathbf{\Lambda}}_{\mathcal{C}_o})^{-1}\rho_t\bar{\mathbf{\Lambda}}_{\mathcal{C}_o}\left(\mathbf{I}-\bar{\mathbf{A}}_{\mathcal{C}_o}^H\mathbf{F}\mathbf{F}^H\bar{\mathbf{A}}_{\mathcal{C}_o}\right)\right], \tag{2.12}$$

since $(\mathbf{I}+\rho_t\bar{\mathbf{\Lambda}}_{\mathcal{C}_o}\bar{\mathbf{A}}_{\mathcal{C}_o}^H\mathbf{F}\mathbf{F}^H\bar{\mathbf{A}}_{\mathcal{C}_o})=\left[(\mathbf{I}+\rho_t\bar{\mathbf{\Lambda}}_{\mathcal{C}_o})-\rho_t\bar{\mathbf{\Lambda}}_{\mathcal{C}_o}(\mathbf{I}-\bar{\mathbf{A}}_{\mathcal{C}_o}^H\mathbf{F}\mathbf{F}^H\bar{\mathbf{A}}_{\mathcal{C}_o})\right]$. $L(\bar{\mathbf{H}},\mathbf{F})$ can be bounded as

$$\begin{aligned}L(\bar{\mathbf{H}},\mathbf{F})&:=R(\bar{\mathbf{H}},\bar{\mathbf{A}}_{\mathcal{C}_o})-R(\bar{\mathbf{H}},\mathbf{F})\overset{(a)}{\leq}R(\bar{\mathbf{H}},\bar{\mathbf{A}}_{\mathcal{C}_o})-R_{\text{lb}}(\bar{\mathbf{H}},\mathbf{F})\\&\leq\log\det\left[\mathbf{I}-(\mathbf{I}+\rho_t\bar{\mathbf{\Lambda}}_{\mathcal{C}_o})^{-1}\rho_t\bar{\mathbf{\Lambda}}_{\mathcal{C}_o}\left(\mathbf{I}-\bar{\mathbf{A}}_{\mathcal{C}_o}^H\mathbf{F}\mathbf{F}^H\bar{\mathbf{A}}_{\mathcal{C}_o}\right)\right]:=L_{\text{ub}}(\bar{\mathbf{H}},\mathbf{F}),\end{aligned} \tag{2.13}$$

where (a) is obtained from (2.12). Because of the difficulty in directly working with the

upper bound of loss, we approximate $L_{\text{ub}}(\bar{\mathbf{H}}, \mathbf{F})$ under high-resolution (number of feedback bits B is reasonably large) and high-SNR ($\rho_t \to \infty$) approximations. When the number of feedback bits B (high-resolution) are large, we have that $\bar{\mathbf{A}}_{\mathcal{C}_o}^H \mathbf{F}\mathbf{F}^H \bar{\mathbf{A}}_{\mathcal{C}_o}$ is close to $\mathbf{I}$ and when ρ_t is large, $(\mathbf{I} + \rho_t \bar{\mathbf{\Lambda}}_{\mathcal{C}_o})^{-1} \rho_t \bar{\mathbf{\Lambda}}_{\mathcal{C}_o} \approx \mathbf{I}$. Therefore $L_{\text{ub}}(\bar{\mathbf{H}}, \mathbf{F})$ can be further approximated as

$$L_{\text{ub}}(\bar{\mathbf{H}}, \mathbf{F}) \overset{\text{large B}}{\approx} \operatorname{tr}\left((\mathbf{I} + \rho_t \bar{\mathbf{\Lambda}}_{\mathcal{C}_o})^{-1} \rho_t \bar{\mathbf{\Lambda}}_{\mathcal{C}_o} \left(\mathbf{I} - \bar{\mathbf{A}}_{\mathcal{C}_o}^H \mathbf{F}\mathbf{F}^H \bar{\mathbf{A}}_{\mathcal{C}_o}\right)\right) \overset{\text{high } \rho_t}{\approx} \operatorname{tr}(\mathbf{I} - \bar{\mathbf{A}}_{\mathcal{C}_o}^H \mathbf{F}\mathbf{F}^H \bar{\mathbf{A}}_{\mathcal{C}_o}). \tag{2.14}$$

In the next section, we use the above defined loss for designing the low-complexity product precoder codebooks.

2.6.3 Product codebook design criterion

Let $\mathcal{F}_h \subseteq \mathcal{U}(M_h, \mathtt{r})$, $\mathcal{F}_v \subseteq \mathcal{U}(M_v, \mathtt{r})$ be the codebooks to quantize $\bar{\mathbf{A}}^{(1)}$, $\bar{\mathbf{A}}^{(2)}$, respectively. Then the codebook $\mathcal{F}$ corresponding to $\bar{\mathbf{A}}$ is constructed using $\mathcal{F}_h$ and $\mathcal{F}_v$ as below.

$$\mathcal{F} = \{(\mathbf{F}_v \otimes \mathbf{F}_h)^*\} \ \forall \ \mathbf{F}_h \in \mathcal{F}_h, \mathbf{F}_v \in \mathcal{F}_v. \tag{2.15}$$

Therefore $\mathcal{F} \subseteq \mathcal{U}(M_t, \mathtt{r}^2)$ and precisely, $\mathcal{F}$ is a finite collection of orthonormal matrices from the tensor product space $\mathcal{U}(M_h, \mathtt{r})$ and $\mathcal{U}(M_v, \mathtt{r})$ i.e., $\mathcal{F} \subseteq \mathcal{U}(M_v, \mathtt{r}) \otimes \mathcal{U}(M_h, \mathtt{r})$. The mapping of $\bar{\mathbf{A}}^{(1)}$, $\bar{\mathbf{A}}^{(2)}$ to the appropriate codewords from $\mathcal{F}_h$, $\mathcal{F}_v$ can be represented as $Q : \mathcal{U}(M, \mathtt{r}) \mapsto \mathcal{F}$, where $(M, \mathcal{F}) = (M_h, \mathcal{F}_h)$, $(M, \mathcal{F}) = (M_v, \mathcal{F}_v)$ for $\bar{\mathbf{A}}^{(1)}$, $\bar{\mathbf{A}}^{(2)}$, respectively and thus the quantized $\bar{\mathbf{A}}$ is obtained as $Q(\bar{\mathbf{A}}) = \left(Q(\bar{\mathbf{A}}^{(2)}) \otimes Q(\bar{\mathbf{A}}^{(1)})\right)^*$. As we proceed, we design the optimal codebooks $\hat{\mathcal{F}}, \hat{\mathcal{F}}_h, \hat{\mathcal{F}}_v$ and the quantizer mapping $Q(\cdot)$ such that average distortion due to quantization is minimized.

From (2.14), the average of the defined loss in mutual information with precoder $\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}$ is

$$\mathbb{E}_{\mathbf{H}}\left[L\left(\bar{\mathbf{H}}, \left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}\right)\right] = \mathbb{E}_{\mathbf{H}}\left[\operatorname{tr}\left(\mathbf{I} - \left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}^H \bar{\mathbf{A}}_{\mathcal{C}_o} \bar{\mathbf{A}}_{\mathcal{C}_o}^H \left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}\right)\right], \tag{2.16}$$

and the optimal codebook $\hat{\mathcal{F}}$ that minimizes the above average loss is

$$\begin{aligned}\hat{\mathcal{F}} &= \underset{\mathcal{F}\subseteq\mathcal{U}(M_v,\mathtt{r})\otimes\mathcal{U}(M_h,\mathtt{r})}{\arg\min} \min_{Q(\cdot)} \mathbb{E}_{\mathbf{H}}\left[L\left(\bar{\mathbf{H}}, \left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}\right)\right] \\ &= \underset{\mathcal{F}\subseteq\mathcal{U}(M_v,\mathtt{r})\otimes\mathcal{U}(M_h,\mathtt{r})}{\arg\min} \max_{Q(\cdot)} \mathbb{E}_{\mathbf{H}}\left[\mathrm{tr}\left(\left(Q(\bar{\mathbf{A}})\right)^H_{\mathcal{C}_Q} \bar{\mathbf{A}}_{\mathcal{C}_o}\bar{\mathbf{A}}^H_{\mathcal{C}_o}\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}\right)\right] \\ &= \underset{\mathcal{F}\subseteq\mathcal{U}(M_v,\mathtt{r})\otimes\mathcal{U}(M_h,\mathtt{r})}{\arg\max} \max_{Q(\cdot)} \mathbb{E}_{\mathbf{H}}\left[\left\|\bar{\mathbf{A}}^H_{\mathcal{C}_o}\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}\right\|^2_F\right].\end{aligned}$$

For every $\mathbf{H}$, the set of indices of $\mathtt{r}$ dominant columns of the unquantized and quantized precoder i.e., $\mathcal{C}_o$ and $\mathcal{C}_Q$ change. To enable the product codebook structure and de-tangle the maximization objective, instead of maximizing $\mathbb{E}_{\mathbf{H}}\left[\left\|\bar{\mathbf{A}}^H_{\mathcal{C}_o}\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}\right\|^2_F\right]$ for designing the codebooks, $\mathbb{E}_{\mathbf{H}}\left[\left\|\bar{\mathbf{A}}^H Q(\bar{\mathbf{A}})\right\|^2_F\right]$ is maximized. Thus the codebook design criterion is modified as

$$\hat{\mathcal{F}} = \underset{\mathcal{F}\subseteq\mathcal{U}(M_v,\mathtt{r})\otimes\mathcal{U}(M_h,\mathtt{r})}{\arg\max} \max_{Q(\cdot)} \mathbb{E}_{\mathbf{H}}\left[\left\|\left(\bar{\mathbf{A}}^{(2)}\otimes\bar{\mathbf{A}}^{(1)}\right)^H\left(Q(\bar{\mathbf{A}}^{(2)})\otimes Q(\bar{\mathbf{A}}^{(1)})\right)\right\|^2_F\right]. \tag{2.17}$$

2.6.4 Connection with product GM

In the above objective, for any rank $-\,\mathtt{r}$ unitary matrices $\mathbf{Q}_1, \mathbf{Q}_2 \in \mathcal{U}_{\mathtt{r}}$ we have

$$\left\|\left(\bar{\mathbf{A}}^{(2)}\otimes\bar{\mathbf{A}}^{(1)}\right)^H\left(Q(\bar{\mathbf{A}}^{(2)})\otimes Q(\bar{\mathbf{A}}^{(1)})\right)\right\|^2_F = \left\|\left(\bar{\mathbf{A}}^{(2)}\otimes\bar{\mathbf{A}}^{(1)}\right)^H\left(Q(\bar{\mathbf{A}}^{(2)})\mathbf{Q}_2\otimes Q(\bar{\mathbf{A}}^{(1)})\mathbf{Q}_1\right)\right\|^2_F.$$

It follows that $\left\|\left(\bar{\mathbf{A}}^{(2)}\otimes\bar{\mathbf{A}}^{(1)}\right)^H\left(Q(\bar{\mathbf{A}}^{(2)})\otimes Q(\bar{\mathbf{A}}^{(1)})\right)\right\|^2_F$ should be maximized not just over orthonormal matrices in $\mathcal{U}(M_v,\mathtt{r})\otimes\mathcal{U}(M_h,\mathtt{r})$ but over equivalence classes of such matrices i.e., over all the matrices such that $Q(\bar{\mathbf{A}}^{(1)})\mathbf{Q}_1 \sim Q(\bar{\mathbf{A}}^{(1)})$ and $Q(\bar{\mathbf{A}}^{(2)})\mathbf{Q}_2 \sim Q(\bar{\mathbf{A}}^{(2)})$. This means that (2.17) should be maximized over GMs. Therefore the codebooks $\mathcal{F}$, $\mathcal{F}_h$ and $\mathcal{F}_v$ can be interpreted as collection of orthonormal basis of subspaces in the GMs i.e., $\mathcal{F}_h \subseteq \mathcal{G}(M_h,\mathtt{r})$, and $\mathcal{F}_v \subseteq \mathcal{G}(M_v,\mathtt{r})$ and thus $\mathcal{F} \subseteq \mathcal{G}(M_v,\mathtt{r})\otimes\mathcal{G}(M_h,\mathtt{r})$. Similar to a CPM $\mathcal{G}^{\times}((M_v,M_h),(\mathtt{r},\mathtt{r}))$, $\mathcal{G}(M_v,\mathtt{r})\otimes\mathcal{G}(M_h,\mathtt{r})$ represents another type of product manifold known as TPM. The *m-fold TPM* is the subset $\mathcal{G}^{\otimes}(\mathbf{n},\mathbf{k}) := \{\mathbf{F}_1\otimes\cdots\otimes\mathbf{F}_m | \mathbf{F}_i \in \mathcal{G}(n_i,k_i), i = 1,\cdots,m\} \subset \mathcal{G}(N,K)$, where $(\mathbf{n},\mathbf{k}) := ((n_1,k_1),(n_2,k_2),\cdots,(n_m,k_m))$, $N = n_1n_2\cdots n_m, K = k_1k_2\cdots k_m$. The

following lemma draws a relation between the two product manifolds, TPM and CPM.

Lemma 2.11. *The m-fold TPM* $\mathcal{G}^{\otimes}(\mathbf{n},\mathbf{k})$ *is diffeomorphic to the m-fold CPM* $\mathcal{G}^{\times}(\mathbf{n},\mathbf{k})$ *i.e., the map* $\varphi : \mathcal{G}^{\times}(\mathbf{n},\mathbf{k}) \mapsto \mathcal{G}^{\otimes}(\mathbf{n},\mathbf{k})$ *is a diffeomorphism*[2].

Hence, there exists a one-to-one mapping from any point $\mathbf{F}_1 \otimes \cdots \otimes \mathbf{F}_m \in \mathcal{G}^{\otimes}(\mathbf{n},\mathbf{k})$ to $(\mathbf{F}_1, \cdots, \mathbf{F}_m) \in \mathcal{G}^{\times}(\mathbf{n},\mathbf{k})$ and vice-versa. Now we provide an approximation for $d_c(\cdot)$ on $\mathcal{G}^{\otimes}(\mathbf{n},\mathbf{k})$ which will be used in constructing the proposed product precoder codebooks.

Assumption 1. If $\mathbf{F}_1 \otimes \cdots \otimes \mathbf{F}_m$, $\mathbf{F}'_1 \otimes \cdots \otimes \mathbf{F}'_m$ are any two points on $\mathcal{G}^{\otimes}(\mathbf{n},\mathbf{k})$, then their preimages on $\mathcal{G}^{\times}(\mathbf{n},\mathbf{k})$ are $[\mathbf{F}] = (\mathbf{F_1}, \cdots, \mathbf{F_m})$, $[\mathbf{F}'] = (\mathbf{F'_1}, \cdots, \mathbf{F'_m})$, respectively. We approximate the distance between the points on the TPM with the distance between their preimages on the CPM as $d_c^2(\mathbf{F}_1 \otimes \cdots \otimes \mathbf{F}_m, \mathbf{F}'_1 \otimes \cdots \otimes \mathbf{F}'_m) \approx d_c^2([\mathbf{F}],[\mathbf{F}']) \approx \sum_{i=1}^{m} d_c^2(\mathbf{F}_i, \mathbf{F}'_i)$.

The codebook design criterion in (2.17) can be interpreted using $d_c(\cdot)$ defined on a GM and can be modified as $\hat{\mathcal{F}} = \underset{\mathcal{F}\subseteq\mathcal{G}^{\otimes}((M_v,M_h),(\mathbf{r},\mathbf{r}))}{\arg\min} \underset{Q(\cdot)}{\min} \mathbb{E}_{\mathbf{H}}\left[d_c^2\left(\bar{\mathbf{A}}, Q(\bar{\mathbf{A}})\right)\right]$. Therefore, the objective for designing the optimal codebook $\hat{\mathcal{F}}$ is equivalent to minimizing the average chordal distance between the two points $\left(\bar{\mathbf{A}}^{(2)} \otimes \bar{\mathbf{A}}^{(1)}\right)$ and $\left(Q(\bar{\mathbf{A}}^{(2)}) \otimes Q(\bar{\mathbf{A}}^{(1)})\right)$ on $\mathcal{G}^{\otimes}\left((M_v, M_h), (\mathbf{r},\mathbf{r})\right)$. From the diffeomorphism between the TPM $\mathcal{G}^{\otimes}\left((M_v, M_h), (\mathbf{r},\mathbf{r})\right)$ and the CPM $\mathcal{G}^{\times}\left((M_v, M_h), (\mathbf{r},\mathbf{r})\right)$, the above optimization objective for $\hat{\mathcal{F}}$ has the following equivalent statement.

$$\hat{\mathcal{F}} = \underset{\mathcal{F}\subseteq\mathcal{G}^{\times}((M_v,M_h),(\mathbf{r},\mathbf{r}))}{\arg\min} \underset{Q(\cdot)}{\min} \mathbb{E}_{\mathbf{H}}\left[d_c^2\left(\left(\bar{\mathbf{A}}^{(2)}, \bar{\mathbf{A}}^{(1)}\right), \left(Q(\bar{\mathbf{A}}^{(2)}), Q(\bar{\mathbf{A}}^{(1)})\right)\right)\right] \tag{2.18}$$

Also, the minimization objective in the above design criterion can be regarded as a measure of average loss in mutual information with a codebook $\mathcal{F}$, where $Q(\bar{\mathbf{A}}^{(1)}) \in \mathcal{F}_h$, $Q(\bar{\mathbf{A}}^{(2)}) \in \mathcal{F}_v$ and thus $L_{\text{ub}}(\mathcal{F}) = \mathbb{E}_{\mathbf{H}}\left[d_c^2\left(\left(\bar{\mathbf{A}}^{(2)}, \bar{\mathbf{A}}^{(1)}\right), \left(Q(\bar{\mathbf{A}}^{(2)}), Q(\bar{\mathbf{A}}^{(1)})\right)\right)\right]$.

Definition 2.12 (Grassmann product codebook for precoding)**.** Under the rank$-(M_r, \mathbf{r}, \mathbf{r})$ approximation of the channel, $\mathbf{H} \approx \bar{\mathbf{H}}_{(1)} = \bar{\mathbf{B}}\bar{\mathbf{G}}_{(1)}(\bar{\mathbf{A}}^{(2)} \otimes \bar{\mathbf{A}}^{(1)})^T$, the Grassmann product codebook $\hat{\mathcal{F}} = \hat{\mathcal{F}}_v \times \hat{\mathcal{F}}_h$ for precoding is the one that minimizes $L_{\text{ub}}(\mathcal{F})$ for a given feedback bit allocation $[B_v, B_h]$ where $|\hat{\mathcal{F}}_h| = 2^{B_h}$, $|\hat{\mathcal{F}}_v| = 2^{B_v}$.

[2] The existence of diffeomorphism between the two manifolds $\mathcal{G}^{\otimes}(\mathbf{n},\mathbf{k})$ and $\mathcal{G}^{\otimes}(\mathbf{n},\mathbf{k})$ implies that the map φ is bijective, φ, φ^{-1} are smooth, continuous, and differentiable as well. See [82] for a more rigorous discussion.

We now state the method to construct $\hat{\mathcal{F}}$ as follows.

Lemma 2.13. *The Grassmann product codebook $\hat{\mathcal{F}} = \hat{\mathcal{F}}_v \times \hat{\mathcal{F}}_h$ as defined in Def. 2.12 can be constructed using the set of centroids $\mathcal{F}_h^K$, $\mathcal{F}_v^K$ obtained from the independent K-means clustering of the principal components $\bar{\mathbf{A}}^{(1)}$, $\bar{\mathbf{A}}^{(2)}$ on $\mathcal{G}(M_h, \boldsymbol{r})$, $\mathcal{G}(M_v, \boldsymbol{r})$ with $K = 2^{B_h}, 2^{B_v}$, respectively.*

Proof. From Def. 2.12 and (2.18), we modify the optimization objective according to the chordal distance approximation in Assum. 1 which gives the following codebook design criterion.

$$\begin{aligned}\hat{\mathcal{F}} &= \hat{\mathcal{F}}_v \times \hat{\mathcal{F}}_h \\ &= \underset{\mathcal{F}\subseteq\mathcal{G}^{\times}((M_v,M_h),(\mathbf{r},\mathbf{r}))}{\arg\min} \min_{Q(\cdot)} \mathbb{E}_{\mathbf{H}}\left[d_c^2\left(\left(\bar{\mathbf{A}}^{(2)},\bar{\mathbf{A}}^{(1)}\right),\left(Q(\bar{\mathbf{A}}^{(2)}),Q(\bar{\mathbf{A}}^{(1)})\right)\right)\right] \\ &= \underset{\substack{\mathcal{F}_h\subseteq\mathcal{G}(M_h,\mathbf{r})\\ \mathcal{F}_v\subseteq\mathcal{G}(M_v,\mathbf{r})}}{\arg\min} \min_{Q(\cdot)} \mathbb{E}_{\bar{\mathbf{A}}^{(2)}}\left[d_c^2\left(\bar{\mathbf{A}}^{(2)},Q(\bar{\mathbf{A}}^{(2)})\right)\right] + \mathbb{E}_{\bar{\mathbf{A}}^{(1)}}\left[d_c^2\left(\bar{\mathbf{A}}^{(1)},Q(\bar{\mathbf{A}}^{(1)})\right)\right].\end{aligned} \tag{2.19}$$

Thus the design criteria for $\hat{\mathcal{F}}_h$, $\hat{\mathcal{F}}_v$ for $\bar{\mathbf{A}}^{(1)}$, $\bar{\mathbf{A}}^{(2)}$ is

$$\hat{\mathcal{F}}_h = \underset{\substack{\mathcal{F}_h\subseteq\mathcal{G}(M_h,\mathbf{r})\\ |\mathcal{F}_h|=2^{B_h}}}{\arg\min} \min_{Q(\cdot)} \mathbb{E}_{\bar{\mathbf{A}}^{(1)}}\left[d_c^2\left(\bar{\mathbf{A}}^{(1)},Q(\bar{\mathbf{A}}^{(1)})\right)\right],\ \hat{\mathcal{F}}_v = \underset{\substack{\mathcal{F}_v\subseteq\mathcal{G}(M_v,\mathbf{r})\\ |\mathcal{F}_v|=2^{B_v}}}{\arg\min} \min_{Q(\cdot)} \mathbb{E}_{\bar{\mathbf{A}}^{(2)}}\left[d_c^2\left(\bar{\mathbf{A}}^{(2)},Q(\bar{\mathbf{A}}^{(2)})\right)\right].$$

Comparing the general Grassmannian K-means clustering objective in (2.4) in Sec. 2.4.2 with the above codebook design criteria for $\hat{\mathcal{F}}_h$, $\hat{\mathcal{F}}_v$, we have $\hat{\mathcal{F}}_h = \mathcal{F}_h^K$ for $[K, n, k] = [2^{B_h}, M_h, \mathtt{r}]$, $\hat{\mathcal{F}}_v = \mathcal{F}_v^K$ for $[K, n, k] = [2^{B_v}, M_v, \mathtt{r}]$, thus $\hat{\mathcal{F}} = \hat{\mathcal{F}}_v \times \hat{\mathcal{F}}_h = \mathcal{F}_v^K \times \mathcal{F}_h^K$ and the corresponding optimal quantizers for $\bar{\mathbf{A}}^{(1)}$, $\bar{\mathbf{A}}^{(2)}$ that minimize the average distortion are $Q(\bar{\mathbf{A}}^{(1)}) = \underset{\mathbf{F}\in\hat{\mathcal{F}}_h}{\arg\min}\, d_c^2(\bar{\mathbf{A}}^{(1)}, \mathbf{F})$, $Q(\bar{\mathbf{A}}^{(2)}) = \underset{\mathbf{F}\in\hat{\mathcal{F}}_v}{\arg\min}\, d_c^2(\bar{\mathbf{A}}^{(2)}, \mathbf{F})$. □

Remark 2.14. The design criterion for optimal product codebook in (2.19) is equivalent to finding the set of optimal K centroids using the K-means clustering algorithm on the CPM $\mathcal{G}^{\times}((M_v, M_h), (\mathtt{r}, \mathtt{r}))$ with the chordal distance metric induced on a CPM. The relation between the chordal distance between two points on a CPM and its factor manifolds as given in

Algorithm 1 Grassmannian K-means Algorithm

procedure Codebook($\mathcal{X}, [K, n, k]$)
  Initialize random $\mathcal{F} = \{\mathbf{F}_1, \cdots, \mathbf{F}_K\}$ on $\mathcal{G}(n, k)$
  Cluster Update: $\mathcal{S}_i \leftarrow \{\mathbf{X} : d_c(\mathbf{X}, \mathbf{F}_i) \leq d_c(\mathbf{X}, \mathbf{F}_j), \forall \mathbf{X} \in \mathcal{X},\ i \neq j\}\ \forall i \in \{1, \cdots, K\}$
  Quantization: $Q_{\mathcal{F}}(\mathbf{X}) \leftarrow \arg\min_{\mathbf{F} \in \mathcal{F}} d_c^2(\mathbf{X}, \mathbf{F})\ \forall \mathbf{X} \in \mathcal{X}$
  while ! stopping criteria **do**
    Centroid Update: $\mathbf{F}_i \leftarrow \arg\min_{\mathbf{F} \in \mathcal{G}(n,k)} \sum d_c^2(\mathbf{X}, \mathbf{F})\ \forall \mathbf{X} \in \mathcal{S}_i, \forall i \in \{1, \cdots, K\}$
    Cluster Update and **Quantization**
  return $\mathcal{F}$

(2.3) simplifies the objective to two separate objectives of finding the optimal centroids using K-means clustering algorithm on the factor manifolds of the CPM $\mathcal{G}^{\times}((M_v, M_h), (\mathtt{r}, \mathtt{r}))$.

The step-wise construction of the proposed unquantized and quantized precoders is summarized in the following remark.

2.6.5 Codebook construction

From Lem. 2.13, it is possible to perform K-means clustering independently on $\mathcal{G}(M_v, \mathtt{r})$, $\mathcal{G}(M_h, \mathtt{r})$ and construct the product precoder codebook with reduced complexity. The construction of the training and testing channel datasets $\mathcal{H}_{\text{train}}$ and $\mathcal{H}_{\text{test}}$ for precoder codebook design is similar to the construction provided for beamforming product codebook design in Sec. 2.5.4. The training procedure yields the optimal precoder codebooks whose performance is evaluated by measuring the average mutual information R_{av} for the channel realizations in the test set $\mathcal{H}_{\text{test}}$ obtained with the proposed quantized precoder construction. The training and testing procedure of the codebook design for a given set of channel realizations is given in the following remark.

Remark 2.15. For a given $\mathcal{H}_{\text{train}}$ and $\mathcal{H}_{\text{test}}$, the Grassmann product codebook for precoding $\hat{\mathcal{F}} = \hat{\mathcal{F}}_v \times \hat{\mathcal{F}}_h$ is obtained by the procedure PCTrain($\mathcal{H}_{\text{train}}$,$[B_v, B_h]$) and the performance of the codebook $\hat{\mathcal{F}}$ is evaluated by the procedure PCTest($\mathcal{H}_{\text{test}}$,$[\hat{\mathcal{F}}_v, \hat{\mathcal{F}}_h]$) as outlined in Alg. 4, where B_h, B_v are the number of bits used to encode $\bar{\mathbf{A}}^{(1)}$, $\bar{\mathbf{A}}^{(2)}$ respectively.

Algorithm 2 Training, testing of the Grassmann product codebook for beamforming

procedure BFTrain($\mathcal{H}_{\text{train}}$, $[B_v, B_h]$)
  Initialize training sets $\mathcal{X}_{\text{train}} = \varnothing$ and $\mathcal{Y}_{\text{train}} = \varnothing$ on $\mathcal{G}(M_h, 1)$ and $\mathcal{G}(M_v, 1)$ respectively
  for $\mathbf{H} \in \mathcal{H}_{\text{train}}$ **do**
    Construct $\tilde{\mathbf{H}}$ from $\tilde{\mathbf{H}}$
    $\tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^H \leftarrow \text{svd}(\tilde{\mathbf{H}})$
    $\mathcal{X}_{\text{train}} \leftarrow \mathcal{X}_{\text{train}} \cup \mathbf{v}_1$, $\mathcal{Y}_{\text{train}} \leftarrow \mathcal{Y}_{\text{train}} \cup \mathbf{u}_1^*$
  $\hat{\mathcal{F}}_h \leftarrow$ CodeBook($\mathcal{X}_{\text{train}}$, $[2^{B_h}, M_h, 1]$, $\hat{\mathcal{F}}_v \leftarrow$ CodeBook($\mathcal{Y}_{\text{train}}$, $[2^{B_v}, M_v, 1]$)
  return $[\hat{\mathcal{F}}_v, \hat{\mathcal{F}}_h]$
procedure BFTest($\mathcal{H}_{\text{test}}$, $[\hat{\mathcal{F}}_v, \hat{\mathcal{F}}_h]$)
  Initialize $\Gamma_{\text{av}} = 0$
  for $\mathbf{H} \in \mathcal{H}_{\text{test}}$ **do**
    Generate $\tilde{\mathbf{H}}$ from $\mathbf{H}$
    $\tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^H \leftarrow \text{svd}(\tilde{\mathbf{H}})$
    $\mathbf{f}_h \leftarrow \arg\min_{\mathbf{f} \in \hat{\mathcal{F}}_h} d_c^2(\mathbf{v}_1, \mathbf{f})$, $\mathbf{f}_v \leftarrow \arg\min_{\mathbf{f} \in \hat{\mathcal{F}}_v} d_c^2(\mathbf{u}_1^*, \mathbf{f})$
    $\Gamma_{\text{av}} \leftarrow \Gamma_{\text{av}} + \frac{1}{\#\mathcal{H}_{\text{test}}} \frac{\Gamma(\mathbf{H}, \mathbf{f}_v \otimes \mathbf{f}_h)}{\Gamma(\mathbf{H}, \mathbf{u}_1^* \otimes \mathbf{v}_1)}$
  return Γ_{av}

2.7 Complexity analysis

In this section, we compute and compare the complexity of the proposed product codebook design technique with the VQ based iterative codebook design method provided in [1, 80] using a detailed complexity analysis. Let the total number of points in the channel training dataset available for the codebook design be N, number of codewords in the codebook be K. Each iteration of the Grassmannian K-means clustering algorithm involves the following steps: the computation of pairwise distances between cluster centroids and data points and the computation of centroid of the data points that belong to each cluster and updating the codebook.

The distance $d_c(\mathbf{X}, \mathbf{Y})$ between any two points $\mathbf{X}, \mathbf{Y} \in \mathcal{G}(M, \mathtt{r})$ requires computation of SVD of $\mathbf{X}^H\mathbf{Y} \in \mathbb{C}^{\mathtt{r}\times\mathtt{r}}$ whose complexity is $\mathcal{O}(\mathtt{r}^3 + M\mathtt{r}^2)$. Therefore the complexity of computing the distance between K centroids and N data points on $\mathcal{G}(M, \mathtt{r})$ is $\mathcal{O}(KN\mathtt{r}^3 + KNM\mathtt{r}^2)$. For the calculation of centroid of a set of p points belonging to a cluster according

Algorithm 3 Greedy algorithm to find the $\mathtt{r}$ dominant columns that forms the precoder in (2.9) and (2.10) for a given $\mathbf{H}$

procedure DomCol($\mathbf{X}, \mathbf{H}, \mathtt{r}$)
  Initialize $\mathcal{C}_o^1 = \varnothing$, $i = 1$
  while $i \leq \mathtt{r}$ **do**
    $c_i = \underset{c_i \notin \mathcal{C}_o^{i-1}}{\arg\max} \log\det\left(\mathbf{I} + \rho_t(\mathbf{H}\mathbf{X}_{\mathcal{C}_o^{i-1}})(\mathbf{H}\mathbf{X}_{\mathcal{C}_o^{i-1}})^H\right)$
    $\mathcal{C}_o^i = \mathcal{C}_o^{i-1} \cup \{c_i\}$
  $\mathcal{C}_o \leftarrow \mathcal{C}_o^{\mathtt{r}}$
  return $\mathbf{X}_{\mathcal{C}_o}$

to Lem. 2.3, it is required to compute SVD of an $M \times M$ matrix obtained by the sum of p $M \times M$ matrices and hence the complexity is $\mathcal{O}(M^2\mathtt{r}p + M^3)$. This gives the computational cost of calculation of K centroids as $\mathcal{O}(M^2N\mathtt{r}+KM^3)$. Thus the total computation cost for a single iteration of the Grassmannian K-means clustering algorithm on $\mathcal{G}(M, \mathtt{r})$ is $\mathcal{O}(M^2N\mathtt{r}+KM^3 + KN\mathtt{r}^3 + KNM\mathtt{r}^2)$.

For the iterative VQ design method in [1], the set of optimal centroids of the rank $-\,\mathtt{r}$ right singular matrices $\bar{\mathbf{V}} \in \mathbb{C}^{M_t \times \mathtt{r}}$ of the channel dataset $\mathcal{H}_{\text{train}}$ forms the precoder codebook. This gives the complexity of single iteration of the VQ design method as $\mathcal{O}(M_t^2N\mathtt{r}+KM_t^3 + KN\mathtt{r}^3 + KNM_t\mathtt{r}^2)$. For the proposed product beamformer and precoder codebook design method, two codebooks with K' codewords each, corresponding to horizontal and vertical dimensions have to constructed using Alg. 2 and 4. The complexity of a single iteration of

Name of scenario	I1_2p5
Active BS	3
Active users	1 to 702
Number of antennas (x, y, z)	(M_v, M_h, M_r)
System bandwidth	0.02 GHz
Antennas spacing	0.5
Number of OFDM sub-carriers	1
OFDM sampling factor	1
OFDM limit	1

Table 2.1: Parameters of the DeepMIMO dataset [2]

Algorithm 4 Training, testing of the Grassmann product codebook for precoding

procedure PCTrain($\mathcal{H}_{\text{train}}$, $[B_v, B_h]$)
  Initialize training sets $\mathcal{A}_{i,\text{train}} = \varnothing$ and $\mathcal{A}_{2,\text{train}} = \varnothing$ on $\mathcal{G}(M_h, \mathtt{r})$ and $\mathcal{G}(M_v, \mathtt{r})$ respectively
  for $\mathbf{H} \in \mathcal{H}_{\text{train}}$ **do**
    Construct $\mathcal{H}$ from $\mathbf{H}$, $\bar{\mathcal{H}}$ from $\mathcal{H}$
    $\bar{\mathbf{B}}\bar{\mathbf{G}}_{(1)}(\bar{\mathbf{A}}^{(2)} \otimes \bar{\mathbf{A}}^{(1)})^T \leftarrow \bar{\mathbf{H}}_{(1)}$
    $\mathcal{A}_{i,\text{train}} \leftarrow \mathcal{A}_{i,\text{train}} \cup \bar{\mathbf{A}}^{(i)}$, $(i = 1, 2)$
  $\hat{\mathcal{F}}_j \leftarrow$ CodeBook($\mathcal{A}_{i,\text{train}}$, $[2^{B_j}, M_j, r]$) $((i, j) = (1, h), (2, v))$
  return $[\hat{\mathcal{F}}_v, \hat{\mathcal{F}}_h]$
procedure PCTest($\mathcal{H}_{\text{test}}$, $[\hat{\mathcal{F}}_v, \hat{\mathcal{F}}_h]$)
  Initialize $R_{\text{av}} = 0$
  for $\mathbf{H} \in \mathcal{H}_{\text{test}}$ **do**
    Construct $\mathcal{H}$ from $\mathbf{H}$ and $\bar{\mathcal{H}}$ from $\mathcal{H}$
    $\bar{\mathbf{B}}\bar{\mathbf{G}}_{(1)}(\bar{\mathbf{A}}^{(2)} \otimes \bar{\mathbf{A}}^{(1)})^T \leftarrow \bar{\mathbf{H}}_{(1)}$
    $Q(\bar{\mathbf{A}}^{(i)}) \leftarrow \arg\min_{\mathbf{F} \in \hat{\mathcal{F}}_j} d_c^2(\bar{\mathbf{A}}^{(i)}, \mathbf{F}), \forall (i, j)$
    $Q(\bar{\mathbf{A}}) \leftarrow \left(Q(\bar{\mathbf{A}}^{(2)}) \otimes Q(\bar{\mathbf{A}}^{(1)})\right)^*$
    $\left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q} \leftarrow$ DomCol $\left(Q(\bar{\mathbf{A}}), \mathbf{H}, \mathtt{r}\right)$
    $R_{\text{av}} \leftarrow R_{\text{av}} + \frac{1}{\#\mathcal{H}_{\text{test}}} R\left(\mathbf{H}, \left(Q(\bar{\mathbf{A}})\right)_{\mathcal{C}_Q}\right)$
  return R_{av}

construction of $\hat{\mathcal{F}}_h$ from $\mathcal{A}_{1,\text{train}}$ is $\mathcal{O}(M_h^2 \mathtt{r} N + K' M_h^3 + K' N M_h^2 \mathtt{r}^2 + K' N \mathtt{r}^3)$ and that of $\hat{\mathcal{F}}_v$ from $\mathcal{A}_{2,\text{train}}$ is $\mathcal{O}(M_v^2 \mathtt{r} N + K' M_v^3 + K' N M_v^2 \mathtt{r}^2 + K' N \mathtt{r}^3)$.

Remark 2.16. Let $M_h = M_v = n$, then $M_t = n^2$ and the computational complexity of the VQ-based codebook design method in [1] is $\mathcal{O}(n^4 N \mathtt{r} + K n^6 + K N \mathtt{r}^3 + K N n^2 \mathtt{r}^2)$ whereas the proposed scheme has significantly lower complexity of $\mathcal{O}(2n^2 \mathtt{r} N + 2K' n^3 + 2K' N \mathtt{r}^3 + 2K' N n \mathtt{r}^2)$ for rank $- \mathtt{r}$ transmission.

After the optimal codebook is designed, Rx needs to choose the optimal quantized precoder from the corresponding precoder codebook, which involves finding the precoder which is closest to the optimal unquantized precoder. For the iterative VQ method in [1], the complexity is $\mathcal{O}(K\mathtt{r}^3 + K M_t \mathtt{r}^2)$ for a codebook with K codewords. Similarly, the complexity of the finding the best product precoder from the proposed product codebook with K' codewords each, corresponding to horizontal and vertical dimensions is $\mathcal{O}(K'\mathtt{r}^3 + K' M_h \mathtt{r}^2) +$

$\mathcal{O}(K'\mathtt{r}^3 + K'M_v\mathtt{r}^2) = \mathcal{O}(2K'\mathtt{r}^3 + K'\mathtt{r}^2(M_v + M_h))$. The computational complexity of determining the dominant $\mathtt{r}$ columns for the quantized precoder using DomCol$(\cdot)$ as shown in Alg 3 involves $\mathcal{O}(\mathtt{r}^3)$ computations of mutual information $R(\mathbf{H}, \mathbf{F})$ [81]. The computation of $R(\mathbf{H}, \mathbf{F})$ involves computing a determinant of a matrix of size $M_r \times M_r$, multiplication of matrices of size $M_r \times M_t$, $M_t \times \mathtt{r}$ and another multiplication of matrices of size $M_r \times \mathtt{r}, \mathtt{r} \times M_r$. Therefore, the computational complexity of DomCol$(\cdot)$ is $\mathcal{O}(\mathtt{r}^3(M_r^3 + \mathtt{r}^2 M_r + M_t M_r \mathtt{r}))$.

Remark 2.17. Let $M_h = M_v = n$, then $M_t = n^2$, then the computational complexity of choosing the optimal precoder from the designed codebooks using the VQ method in [1] is $\mathcal{O}(K\mathtt{r}^3 + Kn^2\mathtt{r}^2)$ whereas the proposed scheme has a complexity of $\mathcal{O}(2K'\mathtt{r}^3 + 2K'n\mathtt{r}^2 + \mathtt{r}^3(M_r^3 + \mathtt{r}^2 M_r + n^2 M_r \mathtt{r}))$ for rank $-\ \mathtt{r}$ transmission.

In the massive MIMO regime, as M_h, M_v increase, construction of codebooks with quartic complexity in [1] can become impractical whereas the proposed method with quadratic complexity is relatively computationally efficient. However, determining the optimal quantized precoder from both the VQ-based codebook [1] and the proposed product codebook has the same asymptotic complexity. We will validate this fact with numerical results presented next.

2.8 Results and discussions

2.8.1 Dataset generation

For the performance evaluation of the Grassmann product codebooks, we consider an indoor communication scenario between the base station and the users operating at 2.5 GHz. The channel realizations are obtained from the DeepMIMO dataset [2], which specifies the ray tracing channel parameters for different locations. The parameters for the generation of channel dataset are provided in Table. 2.1.

2.8.2 Results

We present numerical results to assess the performance of the designed product codebooks for beamforming and precoding in FD-MIMO systems in terms of Γ_{av} and R_{av}, respectively. For a given Tx antenna configuration $M_v \times M_h$ and feedback bits allocation

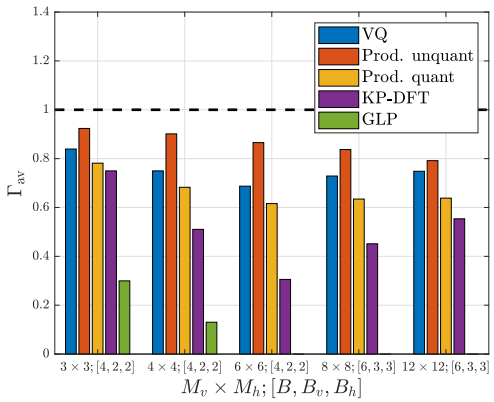


Figure 2.3: Performance comparison (Γ_{av}) of the proposed Grassmann product codebooks with VQ method [1] for various Tx antenna configurations $M_v \times M_h$, feedback bit allocations $[B, B_v, B_h]$, $M_r = 1, \mathtt{r} = 1$.

($[B, B_v, B_h]$), the codebooks are generated using Lem. 2.6 and 2.13, respectively. Here, $[B, B_v, B_h]$ denotes the feedback bit allocation for the limited feedback scheme where B bits are used for the codebooks using the VQ method (referred to as 'VQ') [1,80,83] and $[B_v, B_h]$ is the feedback bit allocation for the Grassmann product codebooks (referred to as 'Prod. quant'). To demonstrate the quantization loss, we also plot Γ_{av} and R_{av} for the unquantized beamformer and precoder (referred to as 'Prod. unquant') as defined in Sec. 2.5.1 and Prop. 2.8 respectively.

In Fig. 2.3, we compare Γ_{av} obtained with the Grassmann product beamformer codebooks with that of the DFT KP codebooks [60] (referred to as 'KP-DFT'), and the codebooks generated based on the Grassmannian line packings (GLP) for correlated channel [34] (referred to as 'Corr-GLP'). For Corr-GLP, the channel correlation matrix $\mathbf{R}$ is calculated from $\mathcal{H}_{\text{train}}$ as $\mathbf{R} = \mathbb{E}_{\mathbf{H}}\left(\mathbf{H}^H\mathbf{H}\right)$. It was not possible to show the performance of the Corr-GLP codebooks for large M_v, M_h because finding the GLP in large dimensions is extremely computation intensive. The KP-DFT codebooks are simple to construct but is outperfromed by our method. This is because the KP-DFT codebooks contain only the beams lying in the direction of the right and left dominant singular vectors of the reshaped FD-MISO channel $\tilde{\mathbf{H}}$ as given in (2.6).

In Fig. 2.4a and 2.4b, we plot the normalized mutual information gain obtained with

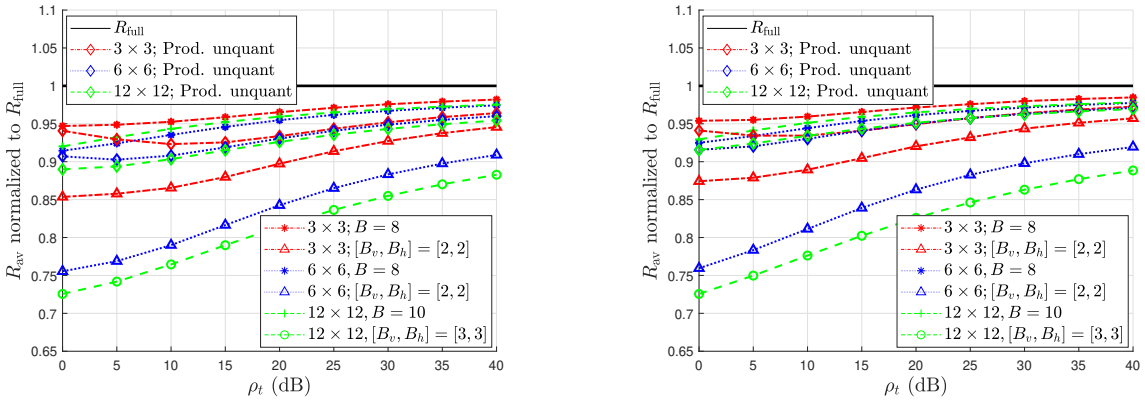
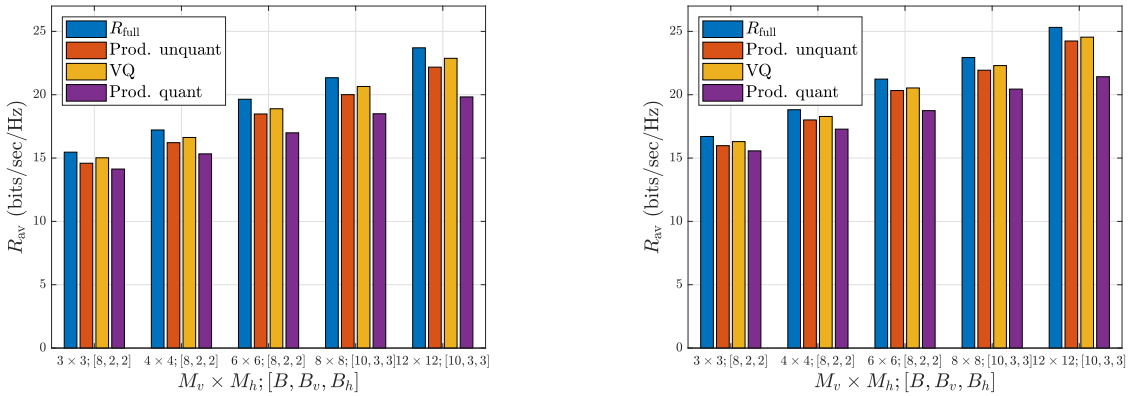


Figure 2.4: Performance comparison of the proposed Grassmann product codebooks with VQ method [1] for various Tx antenna configurations $M_v \times M_h$ and feedback bit allocations $[B, B_v, B_h]$. (a) R_{av} normalized to R_{full} for $M_r = 2, \mathtt{r} = 2$ at varying ρ_t, and (b) R_{av} normalized to R_{full} for $M_r = 3, \mathtt{r} = 2$ at varying ρ_t.

the product precoder codebooks with varying SNR at different feedback bit allocations and Tx antenna configurations. We observe that the performance of the precoder codebooks approach the gain with unquantized product precoders as the number of feedback bits and SNR increase. The sub-optimality of the product codebooks is caused by the loss in beamforming gain and mutual information by the approximation with the unquantized beamformer (Sec. 2.5.1) and precoder (Lem. 2.9). In Fig. 2.5a and 2.5b, we compare the performance of the product codebook and the VQ codebook. As expected, R_{av} for the product precoder codebook is slightly worse than R_{av} of the VQ codebook. This is expected because the VQ works directly on the space of optimal precoders obtained from $\mathcal{H}_{\text{train}}$ while in our method, some accuracy is lost while finding the representation of the product precoder in the TPM. However, as discussed in detail already in Remark 2.16, the VQ codebook construction is significantly more computation intensive than our codebook, as M_v, M_h are large, with diminishing gains in R_{av} as seen in Fig. 2.4, 2.5. To demonstrate the difference in complexity, in Fig. 2.6, we compare the run-times of construction of the codebooks and determing the optimal precoder from the constructed codebooks using the VQ method [1] and the Grassmann product codebooks for different antenna configurations and codebook sizes. The run-times



(a) (b)

Figure 2.5: Performance comparison of the proposed Grassmann product codebooks with VQ method [1] for various Tx antenna configurations $M_v \times M_h$ and feedback bit allocations $[B, B_v, B_h]$. (a) R_{av} for $M_r = 2, \mathtt{r} = 2, \rho_t = 25$ dB, (b) R_{av} for $M_r = 2, \mathtt{r} = 3, \rho_t = 25$ dB, and (c) Normalized run-times for $M_r = 2, \mathtt{r} = 2$.

were obtained by averaging the run-times of the codebook construction algorithms over 500 iterations in the same computation environment. In order to obtain a unit-free measure, we normalized the absolute run-times by dividing them with the average absolute run-time of the Grassmann product codebook for $M_v \times M_h = 3 \times 3$ with $[B_v, B_h] = [4, 4]$. As is evident from this discussion, the VQ method will not scale to large antenna configurations, whereas our method will work well in those cases as well.

2.9 Summary

We explored a classical problem of precoder codebook design in FDD FD-MIMO systems. Given a dataset of channel realizations, this problem has been identified as an application of ML for wireless communications. The novelty lies in identifying a natural tensor representation of the FD-MIMO channel and exploiting it to design low-complexity product precoder codebooks. Using the tensor representation of the channel, we designed a precoder that can be approximated as an element in a TPM which allows us to construct codebooks in its factor manifolds. We also showed that finding the codebooks in the factor manifolds is equivalent to K-means clustering in the factor GMs with chordal distance metric.

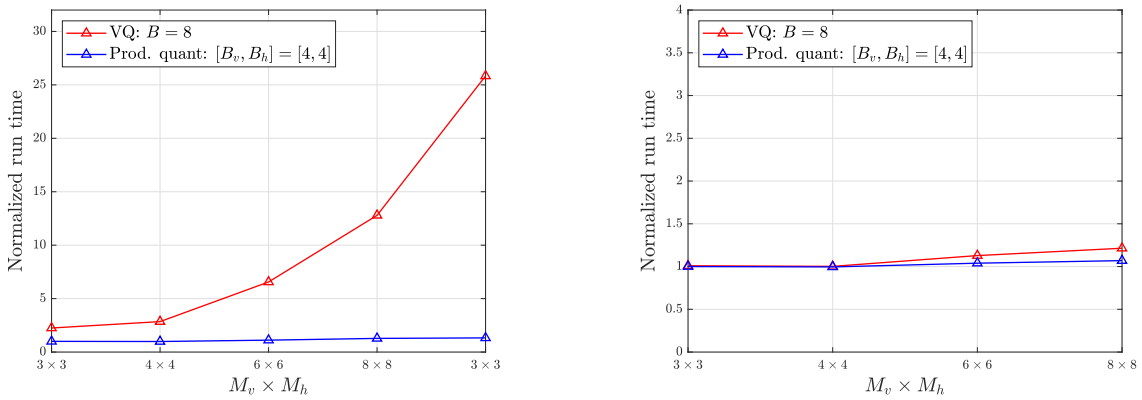


Figure 2.6: Comparison of normalized run-times of the proposed Grassmann product codebooks with VQ method [1] for various Tx antenna configurations $M_v \times M_h$ and feedback bit allocations $[B, B_v, B_h]$. (a) Normalized run-times for construction of codebooks (b) Normalized run-times for choosing the optimal precoder from the designed codebooks.

Through numerical simulations, we demonstrated that this codebook construction method considerably reduces the computational complexity without significantly compromising on the performance of the codebooks when compared to the VQ method, which does not consider the tensor structure of the FD-MIMO channel.

3

Tensor-based Communication-Efficient FL with CNN

In the recent years, we have witnessed the success of ML techniques in tackling many real-world problems. However, the traditional ML algorithms are based on centralized training where the entire training dataset is either present on a single device with high storage and computation power that is required for implementing the ML algorithms or transferred to such a data center or a cloud. With the increasing adoption of technologies like the Internet of Things (IoT), Internet of Everything (IoE), the number of distributed intelligent, powerful, low-energy consumption, smart mobile devices, or users in the world has grown rapidly in the last few years. These devices are usually capable of collecting data, performing local data processing, and implementing local training or inference tasks. The traditional way of implementing an ML algorithm in a distributed user scenario would be to gather the local data from all the users to a data center or a server where the learning takes place and then send the trained model to all the users. This transfer of raw local data to the server introduces many challenges like privacy, network congestion, power consumption, communication overhead, and latency which make the traditional ML algorithms less feasible. This has led to significant research on ideas related to *edge ML* [84, 85], where the edge devices perform the learning tasks, thereby reducing the reliance of the users on the servers and circumventing the need for implementing centralized ML.

An interesting example of edge ML training architecture is FL [39, 86], which facilitates collaborative learning of models among distributed users under the coordination of a central

server. In FL, it is guaranteed that the raw local data remains at the users and is not exchanged or transferred but the distributed users jointly train a model governed by the central server. At each training step, the users share the model updates i.e., model weights or gradients with the server for aggregation to achieve a learning objective. The distributed users are usually mobile and connected to the server through wireless links, thus inducing a distributed wireless computation problem on a high level, taking into account the physical properties of the wireless medium. This intertwining of FL and Wireless Communications for the purpose of collaboratively learning a model or inferring a task leads to applications and challenges which is a two-way synergy, as described below.

FL for Wireless Communications. FL has been used in wireless networks formed by a federation of distributed users connected through wireless links, to improve the network performance like rate, error, latency or infer the characteristics of the network for further improvement in communication(see [87] and references therein). Some of the applications of FL at improving the communication between the distributed users include reducing latency in vehicular communications [88], communication-efficient unmanned aerial vehicle (UAV) online path control [89], localization [90], intrusion detection [91], orientation and mobility prediction in wireless networks [92], power allocation, scheduling, and traffic prediction in connected and automated vehicles (CAV) swarms [93–95], improving the scheduling between users in wireless networks [96], and many more.

Wireless Communications for FL. The existence of the wireless channels between the server and the distributed users brings forth some unique challenges for the practical implementation of FL. The stochastic nature of wireless channels due to interference, fading, and noise introduces new impediments to the learning task. The key challenges in wireless channels, such as transmission outage, latency, or bit errors, lead to poor convergence and affect the accuracy of FL models. Therefore, it is important to facilitate these factors pertinent to the wireless network in the distributed learning problem at hand. Interested readers are advised to refer to the surveys [87, 97–99] and references therein for the impact of wireless channels on FL. Another important aspect of this problem is the scarcity of spectrum. The exchange of model parameters from the users to the server can consume a significant amount of the uplink bandwidth, and thus, there is a need for efficient gradient communication tech-

niques for FL, especially for NN models. Due to the large size and millions or billions of parameters of NN, the model gradients are high-dimensional data structures. This calls for a low-rate efficient gradient communication scheme, where the high dimensional gradients can be encoded, transmitted to the server, and decoded for updating the global model. This chapter focuses on developing such an efficient gradient communication technique, keeping in mind the physical properties of the wireless channel connecting the users and the server.

3.1 Related work

The communication bottleneck has already been acknowledged as a major challenge in the FL literature and several strategies have been proposed by the ML community to reduce the communication overhead (see [86, 100] and references therein). Some of the common approaches in FL for reducing the bandwidth consumption are *gradient quantization* and *gradient sparsification*. The principle of gradient quantization for communication-efficient FL is based on low-precision training by exploiting the stochastic properties of gradients and using a limited number of bits for the quantization of gradients. Some of those works include 1-bit implementation of SGD [44], Quantized SGD (QSGD) [42] where each entry of gradients is quantized, signSGD [41] where only the sign of each mini-batch stochastic gradient is transmitted to the server, and TernGrad [40] where each entry of gradients is mapped to $\{-1, 0, 1\}$. Most of these works quantize each entry of the gradient to bits and transmit the quantized gradients but do not treat the gradient vector as a whole entity, thus, ignoring its original structure. Despite being simple and lossy, these quantization schemes showed significant performance gains in gradient compression while being grounded in theory in terms of their convergence properties. Gradient sparsification reduces the dimension of the gradients by transmitting only a few selected gradient entries and setting the remaining entries to zeros. Various strategies to select the important entries of a gradient such as *top-K* [101, 102], *rand-K* sparsification [103], and using a pre-defined threshold [104] have been been proposed. Although there is a loss in gradient information, gradient sparsification has shown great compression of gradients (and hence reduction in communication bandwidth) without a significant loss in performance of the models. Several approaches, including QSGD [42], ATOMO [105], and TernGrad [40] combine quantization and sparsification to improve per-

formance gains while providing provable guarantees for convergence [106].

The gradient communication approaches in the literature mostly assume that the links between the distributed devices and the server are reliable, interference-and-error free and thus, ignoring the uncertainties introduced by the wireless nature of the communication medium between them. However, there has been a recent shift in focus on designing efficient gradient communication techniques that account for the physical properties of wireless channels [45, 107–109]. Motivated by the multi-user scenario in FL, various multi-access schemes for FL have been proposed. The classic orthogonal-access schemes such as OFDMA [110], and TDMA [99] have been used for creating independent links between the users and the server to support gradient communication for the distributed training. In [45, 107, 111], the wireless channel between the users and the server is modeled as a Gaussian or fading MAC, to incorporate the interference from the users and channel impairments. In [45], where the channel is modeled as a Gaussian MAC, the users transmit their local gradient estimates over the MAC to the server. In the case of fading MAC [107], additionally, each user controls the transmit power to mitigate the fading effect at the server, and the users that experience deep fading do not transmit for that training iteration. The authors in [45, 107, 111] designed a series of analog distributed stochastic gradient descent (A-DSGD) algorithms under different MAC models, power allocation schemes, and scheduling constraints, in which users transmit their compressed local gradient over the MAC which is aggregated over-the-air. The authors combined the physical properties of MAC with gradient quantization, sparsification, and error accumulation to design gradient communication techniques.

Most of the existing gradient communication techniques flatten the gradients to vectors, thereby discarding any of its structural properties. However, multi-dimensional array structures also naturally occur as gradients in NN. For instance, gradients of a fully connected layer can be naturally represented as a matrix, gradients of a convolutional layer can be naturally represented as a 4D-tensor. Flattening the gradients to a vector may destroy the relationship between the elements across different dimensions of the gradient. In this chapter, we study analog gradient communication in a distributed wireless system, where the users are connected through a Gaussian MAC adopted from [45], CNNs are trained at the users and the server and design a novel gradient communication technique that is inspired

by the tensor structure of the convolutional gradients.

3.2 Contributions and novelty

In this chapter, we propose an efficient gradient communication (compression-reconstruction) approach for FL with CNNs by leveraging the tensor structure of the convolutional gradients. We employ gradient sparsification at the users to sparsify the gradients which are then compressed to a low-dimensional estimate using CS techniques before transmitting to the server. The novelty lies in the gradient reconstruction algorithm at the server which exploits the properties of the convolutional gradients that arise from its tensor structure. In particular, our primary contributions are (1) the identification of a natural tensor representation of the convolutional gradients, (2) the empirical demonstration of spatial consistency in the convolutional gradients among neighboring gradient elements in different dimensions, that stems from its tensor structure and the gradient computation using back-propagation, (3) the utilization of the demonstrated properties of the convolutional gradients to propose a novel prior for modeling them appropriately, and (4) the selection of a Bayesian reconstruction framework that can be applied to this setting and modifying it to combine with the imposed prior and the underlying correlation structure to design an efficient algorithm for gradient reconstruction. Numerical results show that the federated training of the global CNN model with the proposed reconstruction algorithm has a faster convergence compared to its existing counterpart A-DSGD [45].

Notations. $\mathbb{R}$ represents the set of all real values. We use boldface capital letters like $\mathbf{A}$ to represent matrices and $\mathbf{A}^T, \|\mathbf{A}\|_2$, and $\mathbf{A}_{ij}$ denote the transpose, l_2 induced norm of $\mathbf{A}$, and the element of $\mathbf{A}$ at the i-th row, j-th column, respectively. We use boldface small letters like $\mathbf{a}$ to represent vectors and $\mathbf{a}^T, \|\mathbf{a}\|_F$, and $\mathbf{a}_i$ denote the transpose, Frobenius norm of $\mathbf{a}$, and the i-th element of $\mathbf{a}$, respectively. For a positive integer i, we define $[i] = \{1, \cdots, i\}, [i, j] = \{i, i+1, \cdots, j\}$ for $j < i$. $\mathcal{N}(a; \mu, \psi)$ denotes the pdf of a Gaussian random variable a with mean μ and variance ψ, $\mathbb{E}_a(\cdot)$, and $\text{var}_a(\cdot)$ represent the expectation and the variance over the distribution of any random variable a. Finally, we use $|\cdot|$, and $\delta(\cdot)$ to represent the absolute value, and Dirac delta, respectively. If $\mathcal{A}$ is a set, then the cardinality of $\mathcal{A}$ is denoted by $|\mathcal{A}|$. Additionally, we adopt the notations and premilinaries

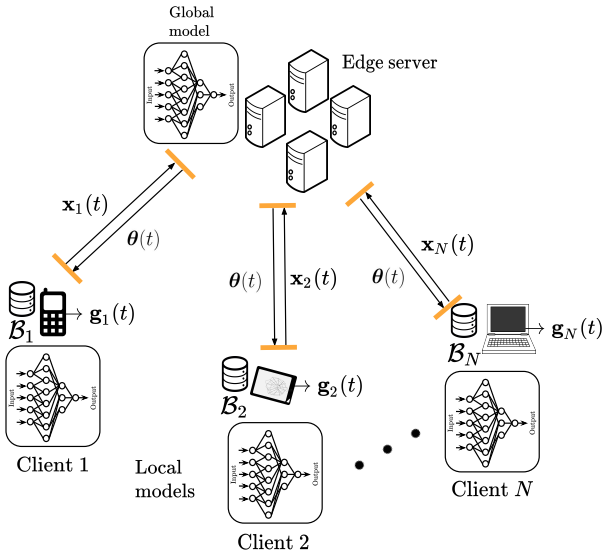


Figure 3.1: Illustration of the FL framework with N distributed wireless users and the server.

on tensors from Chapter 2.

3.3 Problem setup

In this section, we adopt the FL system with N distributed wireless users connected to a central server through a Gaussian MAC and use DSGD to collaboratively train a global model from [45]. We discuss the procedure for FL with DSGD and the implications of the MAC channel in the FL procedure in the sequel.

3.3.1 Preliminaries

FL with DSGD. Fig. 3.1 illustrates the described FL system which solves an unconstrained optimization problem of minimizing an empirical loss function

$$F(\boldsymbol{\theta}) = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{u} \in \mathcal{B}} l(\boldsymbol{\theta}, \mathbf{b}),$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$ represents the model parameters that are to be optimized, $\mathcal{B}$ is the available dataset, and $l(\cdot)$ is the loss function associated with the model parameters $\boldsymbol{\theta}$ and data point $\mathbf{b} \in \mathcal{B}$. A typical implementation of the minimization is through iterative update of $\boldsymbol{\theta}$ using

SGD where the model parameters at t-th iteration is given as

$$\boldsymbol{\theta}(t+1) := \boldsymbol{\theta}(t) - \eta\, \mathbf{g}(t). \tag{3.1}$$

Here, η represents the learning rate, $\mathbf{g}(t)$ represents a stochastic gradient at t-th iteration. Let the set of data samples available at n-th user be $\mathcal{B}_n$, then the total available dataset $\mathcal{B} = \cup_{n=1}^{N} \mathcal{B}_n$. In the current setting of FL with distributed users and their respective local datasets $\mathcal{B}_n$, DSGD allows the users to collaboratively train the global model $\boldsymbol{\theta}(t)$ using the aggregated local gradients. At t-th iteration, each user computes a stochastic gradient $\mathbf{g}_n(t)$ using SGD with respect to its local dataset $\mathcal{B}_n$ and the global model $\boldsymbol{\theta}(t)$ and sends it to the server. In DSGD, the server aggregates the local gradients from all the users to compute a stochastic gradient $\mathbf{g}(t)$ to update the global model according to (3.1) where $\mathbf{g}(t) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{g}_n(t)$. The server then broadcasts the updated model $\boldsymbol{\theta}(t+1)$ to all the users through an error-free shared link [45]. Therefore in FL with DSGD, the server requires the average of the local gradients i.e. $\mathbf{g}(t)$ instead of the actual local gradients $\mathbf{g}_n(t) \forall\ n \in [N]$.

Gaussian MAC. We adopt the Gaussian MAC channel [45] to model the wireless medium between the users and the server over which the local gradient estimates at each iteration are transmitted to the server which is characterized as

$$\mathbf{y}(t) = \sum_{n=1}^{N} \mathbf{x}_n(t) + \mathbf{z}(t), \tag{3.2}$$

where $\mathbf{x}_n(t) \in \mathbb{R}^d$ is the length-d signal transmitted by the n-th user, $\mathbf{y}(t)$ is the signal received by the server, and $\mathbf{z}(t)$ is the additive white Gaussian noise (AWGN) at t-th iteration and each entry z_n is i.i.d. according to $\mathcal{N}(z_n; 0, \psi)$. Since we use DSGD for training the global model, the goal of the server is to recover $\mathbf{g}(t) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{g}_n(t)$ from $\mathbf{y}(t)$ and update the global model as in (3.1). At each iteration, every user pre-processes the gradients before transmitting over the MAC and thus $\mathbf{x}_n(t)$ is simply a function of the local gradient $\mathbf{g}_n(t)$. It may not be possible for the server to recover $\mathbf{g}(t)$ perfectly from $\mathbf{y}(t)$ due to the pre-processing by the users and noise added by the channel. Therefore, we obtain a noisy estimate of $\mathbf{g}(t)$ i.e. $\hat{\mathbf{g}}(t)$ using a reconstruction procedure to update the model parameter

$\boldsymbol{\theta}(t+1)$. The algorithm for the described FL training where the distributed users pre-process the gradients before transmitting over the Gaussian MAC to the server where they are reconstructed back is shown in Alg. 5. The procedure for the gradient compression (represented as GradCompress$(\cdot)$) and the reconstruction (represented as GradRecon$(\cdot)$) are described in the following sections and shown in Alg. 6 and 7 respectively.

Gradient sparsity and its sparsification. The gradients of a NN are observed to be skewed with most of the entries close to zero but a few of them are large and significant [101, 104]. Therefore the number of the significant gradient entries is much less than the actual dimension of the gradient vector. The skewed nature of the gradients is supported by Fig. 3.3, where the histogram of the gradient shows that most entries of the gradient are close to zero but a few dominant entries. We employ gradient sparsification at the users to reduce the transmission bandwidth by transmitting only the dominant entries of the gradients to the server. In particular, each user sets all but few elements of the gradient vector that are larger in magnitude than a pre-defined threshold to zero. The threshold is chosen such that $s\%$ of the number of elements in the K-length gradient vector is not transmitted and s is usually a large number close to but less than 100. This s-level sparsification is represented as sparsify$(\cdot, s)$ in Alg. 6. The sparse nature of the gradients and sparsification are demonstrated in Fig. 3.4a and Fig. 3.4b.

Error accumulation. Let the sparsified local gradient be represented as $\mathbf{g}_n^{\text{sp}}(t)$. In order to avoid the information loss of the local gradients in the process of sparsification, the users employ error accumulation where the gradients that are set to zero are accumulated locally and added to the gradient corresponding to the next iteration, which is performed for each iteration. Every user $n \in [N]$ maintains an accumulated error vector denoted by $\boldsymbol{\Delta}_n(t) \in \mathbb{R}^K$ initialized as $\boldsymbol{\Delta}_n(0) = \mathbf{0}$, to keep track of the gradient residuals after sparsification at each iteration t. After the computation of the local gradient $\mathbf{g}_n(t)$, each user updates the local gradient estimate $\mathbf{g}_n^{\text{ec}}(t)$ using the accumulated error as

$$\mathbf{g}_n^{\text{ec}}(t) = \mathbf{g}_n(t) + \boldsymbol{\Delta}_n(t), \forall n \in [N].$$

Algorithm 5 Pseudo algorithm for FL procedure with DSGD

Initialize: $\boldsymbol{\theta}(0), \boldsymbol{\Delta}_n(0) = 0 \forall\ n = [N]$
for $t = 0, \cdots, T-1$ **do**
  Broadcast $\boldsymbol{\theta}(t)$ to all users in $[N]$ i.e. $\boldsymbol{\theta}_n(t) = \boldsymbol{\theta}(t)$
  - **Users do**
  for $n = [N]$ in parallel **do**
    Compute $\mathbf{g}_n(t)$ with respect to $\mathcal{B}_n$ using SGD
    $\boldsymbol{\theta}_n(t+1) = \boldsymbol{\theta}_n(t) - \eta \mathbf{g}_n(t)$
    Compress $\mathbf{g}_n(t)$ for transmission: $\mathbf{x}_n(t) = \text{GradCompress}(\mathbf{g}_n(t), \boldsymbol{\Delta}_n(t))$
  - **Channel does**
    $\mathbf{y}(t) = \sum_{n=1}^{N} \mathbf{x}_n(t) + \mathbf{z}(t)$
  - **Server performs**
    Reconstruct $\hat{\mathbf{g}}^s(t)$ from $\mathbf{y}(t)$: $\hat{\mathbf{g}}(t) = \text{GradRecon}(\mathbf{y}(t))$
    $\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta \frac{1}{N} \hat{\mathbf{g}}^s(t)$

Algorithm 6 Algorithm for gradient compression

procedure GradCompress($\mathbf{g}_n(t), \boldsymbol{\Delta}_n(t)$)
  $\mathbf{g}_n^{\text{ec}}(\boldsymbol{\theta}_t) = \mathbf{g}_n(\boldsymbol{\theta}_t) + \boldsymbol{\Delta}_n(t)$
  $\mathbf{g}_n^{\text{sp}}(\boldsymbol{\theta}_t) = \text{sparsify}(\mathbf{g}_n^{\text{ec}}(\boldsymbol{\theta}_t), s)$
  $\boldsymbol{\Delta}_n(t+1) = \mathbf{g}_n^{\text{ec}}(\boldsymbol{\theta}_t) - \mathbf{g}_n^{\text{sp}}(\boldsymbol{\theta}_t)$
  $\mathbf{x}_n(t) = \mathbf{A}\mathbf{g}_n^{\text{sp}}(\boldsymbol{\theta}_t)$
  return $\mathbf{x}_n(t)$

The accumulated error vector is then updated as $\boldsymbol{\Delta}_n(t+1) = \mathbf{g}_n^{\text{ec}}(t) - \mathbf{g}_n^{\text{sp}}(t)$, which is the difference between the local gradient estimate $\mathbf{g}_n^{\text{ec}}(t)$ and its sparsified version $\mathbf{g}_n^{\text{sp}}(t)$ where $\mathbf{g}_n^{\text{sp}}(t) = \text{sparsify}(\mathbf{g}_n^{\text{ec}}(t), s)$. The error accumulation at each user n is shown in Alg. 6.

3.3.2 Gradient compression

One of the most important challenges in gradient communication is the bandwidth limitation and hence its compression is significantly important. As we have seen, the dimension of the gradients in NN is generally large but the number of significant entries of the gradient is relatively less. Hence, we transmit only the significant non-zero entries of the sparse gradients. However, transmitting the indices of the non-zero entries of the gradients con-

sumes additional bandwidth. To transmit the local sparse gradient estimates over a limited bandwidth channel, the users use a random projection matrix to compress the gradients using CS [45]. At each iteration t, the sparse gradient $\mathbf{g}_n^{\text{sp}}(t) \in \mathbb{R}^K$ is linearly compressed to a d-length vector $\mathbf{x}_n(t)$ using a measurement matrix $\mathbf{A} \in \mathbb{R}^{d\times K}, (d < K)$ according to $\mathbf{x}_n(t) = \mathbf{A}\mathbf{g}_n^{\text{sp}}(t)$. The measurement matrix $\mathbf{A}$ is a pseudo-random matrix where each entry $\mathbf{A}_{ij}$ is i.i.d. according to $\mathcal{N}(\mathbf{A}_{ij}; 0, 1/d)$. We assume that $\mathbf{A}$ is generated and shared among the users and the server before the distributed training starts. Each user then transmits the compressed vector $\mathbf{x}_n(t)$ over the MAC and the server receives

$$\mathbf{y}(t) = \sum_{n=1}^{N} \mathbf{x}_n(t) + \mathbf{z}(t) = \sum_{n=1}^{N} \mathbf{A}\mathbf{g}_n^{\text{sp}}(t) + \mathbf{z}(t) = \mathbf{A}\sum_{n=1}^{N} \mathbf{g}_n^{\text{sp}}(t) + \mathbf{z}(t).$$

Let the gradient-sum $\sum_{n=1}^{N} \mathbf{g}_n^{\text{sp}}(t)$ be represented as $\mathbf{g}^{\text{s}}(t)$, then the gradient transmission over the Gaussian MAC is modeled as (omitting the iteration number)

$$\mathbf{y} = \mathbf{A}\mathbf{g}^{s} + \mathbf{z}. \tag{3.3}$$

With the gradient sparsification and error accumulation, the objective of the server is to recover $\frac{1}{N}\sum_{n=1}^{N} \mathbf{g}_n^{\text{sp}}(t)$ from $\mathbf{y}(t)$ with the knowledge of $\mathbf{A}$. After the local gradients are compressed and transmitted through the Gaussian MAC, the server reconstructs it back to obtain a noisy estimate $\hat{\mathbf{g}}^{s}(t)$ which is then used to update the global model as

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta\frac{1}{N}\hat{\mathbf{g}}^{s}(t).$$

As we can see from the above global model update equation using the noisy estimate of the gradient-sum, the accuracy of the gradient reconstruction at the server affects the performance of the global model. The higher the accuracy of the reconstruction, the better the convergence of the global model. In this chapter, we design a novel algorithm for reconstructing the gradients of the CNN model at the server that leverages properties pertinent specifically to the convolutional gradients which is discussed in detail in the sequel.

Algorithm 7 Algorithm for gradient reconstruction

procedure GradRecon($\mathbf{y}(t)$)

Initialization: Set $i = 1, T_{\max}, \epsilon_{\text{tol}}$. Initialize θ, ϕ, ψ, $\boldsymbol{\lambda}$, $\boldsymbol{\mu}^g(1)$, $\boldsymbol{\sigma}^g(1)$, $\boldsymbol{\mu}^s(0) = \mathbf{0}$

for $i = 1 : 1 : T_{\max}$ **do**

Factor Node update: $\forall\ m = [d]$

$$\mu_m^p(i) = \sum_{n=1}^{K} \mathbf{A}_{mn}\mu_n^g(i) - \sigma_m^p(i)\mu_m^s(t-1), \sigma_m^p(i) = \sum_{n=1}^{K} |\mathbf{A}_{mn}|^2 \sigma_n^g(i)$$

$$\mu_m^w(i) = \mathbb{E}_{W|\mathbf{Y}}\left[w_m|\mathbf{y}; \mu_m^p(i), \sigma_m^p(i), \mathbf{q}_{\text{in}}\right] = \mu_m^p(i) + \frac{\sigma_m^p(i)}{\sigma_m^p(i) + \psi(i)}(y_m - \mu_m^p(i))$$

$$\sigma_m^w(i) = \text{var}_{W|\mathbf{Y}}\left[w_m|\mathbf{y}; \mu_m^p(i), \sigma_m^p(i), \mathbf{q}_{\text{in}}\right] = \frac{\sigma_m^p(i)\psi(i)}{\sigma_m^p(i) + \psi(i)}$$

$$\mu_m^s(i) = \frac{\mu_m^w(i) - \mu_m^p(i)}{\sigma_m^p(i)}, \sigma_m^s(i) = \frac{(1 - \sigma_m^w(i)/\sigma_m^p(i))}{\sigma_m^p(i)}$$

Variable Node update: $\forall\ n = [K]$

$$\mu_n^r(i) = \mu_n^g(i) + \sigma_n^r(i)\sum_{m=1}^{d} \mathbf{A}_{mn}\mu_m^s(i), \sigma_n^r(i) = \sum_{m=1}^{d} |\mathbf{A}_{mn}|^2 \sigma_m^s(i)$$

$$\hat{g}_n^s(i+1) = \mu_n^g(i+1) = \mathbb{E}_{G|\mathbf{Y}}\left[g_n^s|\mathbf{y}; \mu_n^r(i), \sigma_n^r(i), \mathbf{q}_{\text{in}}\right] = \pi_n(i)\gamma_n(i)$$

$$\sigma_n^g(i+1) = \text{var}_{G|\mathbf{Y}}\left[g_n^s|\mathbf{y}; \mu_n^r(i), \sigma_n^r(i), \mathbf{q}_{\text{in}}\right] = \pi_n(i)(\gamma_n^2(i) + \nu_n(i)) - (\mu_n^g(i+1))^2$$

if $\frac{\|\hat{\mathbf{g}}^s(i+1) - \hat{\mathbf{g}}^s(i)\|}{\|\hat{\mathbf{g}}^s(i)\|} \leq \epsilon_{\text{tol}}$ **then, break**

return $\hat{\mathbf{g}}^s(i)$

3.4 Proposed gradient reconstruction approach

In this section, we present a novel reconstruction algorithm (GradRecon($\cdot$)) in Alg. 7, for recovering the gradient-sum at the server according to the described FL procedure in Alg. 5. We first present the motivation and intuition behind the spatial consistency in the convolutional gradients and then present a prior for the convolutional gradients that models this correlation. At the core of the proposed gradient reconstruction method is this novel prior that captures the spatial consistency of the convolutional gradients and a sparse signal reconstruction technique that enables to incorporate any arbitrary prior on the sparse signal

and likelihood model.

3.4.1 Spatial consistency of convolutional gradients

In order to demonstrate the spatial consistency in a convolutional layer, we describe its tensor structure first (refer Fig. 3.2). A (H, W, D, F)-convolutional layer has F filters, each of dimension (H, W, D), where H, W, D is the height, width, depth of each filter and acts on an input producing a gradient tensor $\mathcal{G} \in \mathbb{R}^{H \times W \times D \times F}$. In Fig. 3.2, we demonstrate the formation of vector $\mathbf{g}$, matrix $\mathbf{G}$ representation of $\mathcal{G}$ and the mapping of $\mathbf{g}, \mathbf{G}$ to $\mathcal{G}$ and vice-versa. We define a distance on $\mathbf{G}$ to characterize the correlation between its elements. For this purpose, we use the manhattan distance (using only the coordinates corresponding to H, W) to measure the distance between any two columns of $\mathbf{G}$. Here, $\mathcal{G}(h, w, :, :)$ for some $h \in [H], w \in [W]$ represents the $D \times F$ matrix formed by the gradients at h-th row, w-th column of $\mathcal{G}$. A column of $\mathbf{G}$ is formed by the elements in $\mathcal{G}(h, w, :, :)$ for some $h \in [H], w \in [W]$ and the gradients adjacent (with respect to tensor) to this column are the ones corresponding to $\mathcal{G}(h \pm 1, w, :, :)$, $\mathcal{G}(h, w \pm 1, :, :)$. Therefore, the distance between the columns formed by $\mathcal{G}(h, w, :, :)$ and any of the columns formed by $\mathcal{G}(h \pm 1, w, :, :)$, $\mathcal{G}(h, w \pm 1, :, :)$ is 1. Similarly, the distance between the columns formed by $\mathcal{G}(h, w, :, :)$ and any of the columns formed by $\mathcal{G}(h \pm 2, w, :, :)$, $\mathcal{G}(h, w \pm 2, :, :)$, $\mathcal{G}(h \pm 1, w \pm 1, :, :)$ is 2. The development of the proposed gradient reconstruction approach is strongly motivated from the following empirical observation on the convolutional gradients.

Spatial Domain Consistency. An interesting empirical observation of the convolutional gradients is that in a (H, W, D, F)-convolutional layer, the gradient at the location $(h, w, d, f), h \in [H], w \in [W], d \in [D], f \in [F]$ i.e., $\mathcal{G}(h, w, d, f)$ shows strong similarity with its neighbors. The intuition behind this spatial domain consistency in the convolutional gradients is the shared local connectivity of the convolutional weights in computing the outputs of the current layer [50]. The backpropagation for the computation of convolutional gradients gives that $\mathcal{G}(h, w, d, f)$ is obtained as the cross-correlation of the gradients of the f-th channel of current layer w.r.t its output and d-th channel of the inputs to the current layer [50]. Therefore, the equations for computation of the gradients $\mathcal{G}(h, w, d, f)$ and $\mathcal{G}(h - \mathcal{H}, w - \mathcal{W}, d, f), \mathcal{H} \subseteq [-(H-1), (H-1)], \mathcal{W} \subseteq [-(W-1), (W-1)]$ share some of the

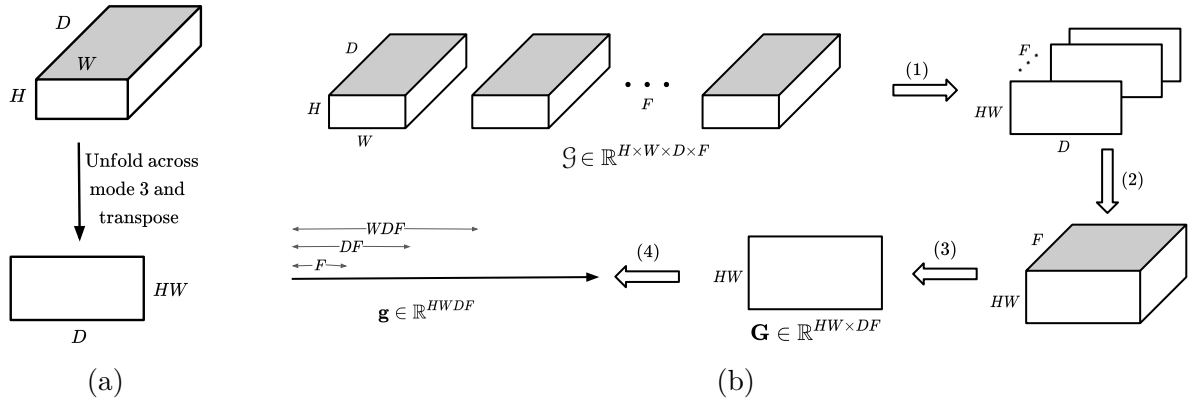


Figure 3.2: (a) Illustration of the natural tensor structure in the gradient generated by a (H, W, D)-convolutional filter of size $H \times W \times D$ and *unfolding* the tensor gradient to obtain a matrix, (b) Demonstration of the natural tensor structure of the gradient $\mathcal{G} \in \mathbb{R}^{H \times W \times D \times F}$ generated by a (H, W, D, F)-convolutional layer: (1) Unfolding of 3D tensor gradients generated by each of the F filters, (2) Stacking the F matrix gradients along the 3rd dimension to form a 3D tensor, (3) Unfolding of the obtained 3D tensor gradient generated by the F filters to form a matrix gradient $\mathbf{G} \in \mathbb{R}^{HW \times DF}$, and (4) Flattening of the obtained matrix gradient $\mathbf{G}$ by concatenating the rows consecutively, to form a vector gradient $\mathbf{g} \in \mathbb{R}^{HWDF}$. (1), (2), & (3) together represent matricization of $\mathcal{G}$.

pixels from the input to the current layer and hence these gradients show strong similarity with each other.

Fig. 3.4a, 3.4b and 3.4c show a $(5, 5, 10, 20)$-convolutional gradient $\mathbf{G} \in \mathbb{R}^{HW \times DF}$, its sparsified version $\mathbf{G}^{\text{sp}}$ and its correlation matrix $\mathbf{R}$ where each entry $\mathbf{R}_{ij}$ shows the pairwise correlation coefficient of i-th and j-th columns of $\mathbf{G}^{\text{sp}}$. Although the distance between columns is measured based on the tensor representation $\mathcal{G}$, we show the correlation matrix of the matrix gradient for illustration purposes. From Fig. 3.4a, 3.4b, we observe that the non-zero entries of $\mathbf{G}^{\text{sp}}$ (entries remaining after sparsification) tend to be present in clusters with high probability i.e., they are strongly correlated with their nearest neighboring elements. For sparse signals with such a clustering pattern, if the nearest neighbors of an element are zero (non-zero), then it will also tend to be zero (non-zero) with high probability. This correlation is supported by the above argument of the convolutional gradient computation using backpropagation. This spatial consistency in the gradients is also corroborated by Fig. 3.4c, where we can see a gradual fading pattern in $\mathbf{R}$ as we move away from each diagonal

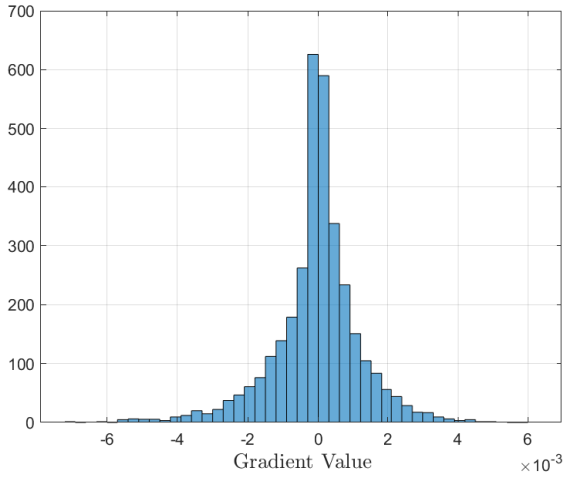


Figure 3.3: Histogram of a sample $(5, 5, 10, 20)$-convolutional gradient $\mathcal{G}$

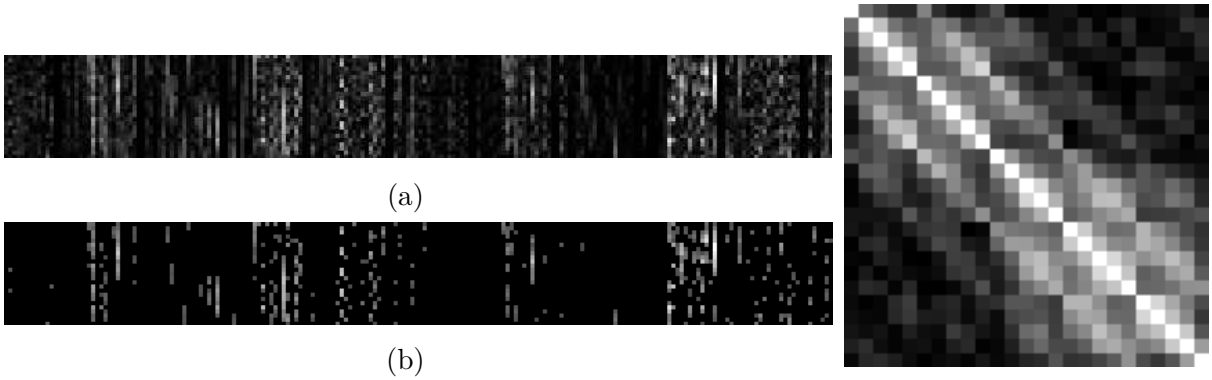
element $\mathbf{R}_{ij}, \forall i = j$. This gradual fading pattern in $\mathbf{R}$ indicates that the correlation of i-th column of $\mathbf{G}$ with j-th column of $\mathbf{G}$ is maximum when j belongs to the index of the nearest neighboring columns of i and the correlation decreases rapidly as the distance between i-th and j-th columns increase. In the next section, we provide a scheme for exploiting this correlation of gradient elements with its adjacent gradients to improve the performance of the global model in the described FL system.

3.4.2 Bayesian modeling of gradients

In this section, we propose a novel prior to model the convolutional gradients that captures the sparsity of the gradients and the aforementioned spatial consistency. The Bayesian modeling of gradients requires the definition of a distribution $p(\mathbf{g}^s)$. We assume that the entries of $\mathbf{g}^s$ are independently distributed i.e.,

$$p(\mathbf{g}^s) := \prod_{n=1}^{K} p(g_n^s)$$

To enforce sparsity from a Bayesian perspective, the gradients are assumed to follow a



(a)

(b)

(c)

Figure 3.4: Illustration of the $(5, 5, 10, 20)$-convolutional gradient $\mathcal{G}$ rearranged into $\mathbf{G} \in \mathbb{R}^{25\times 200}$ from Fig. 3.3 as described in Fig. 3.2b. (a) Gray-scale image representation of $\mathbf{G}$, (b) Gray-scale image representation of $\mathbf{G}^{\text{sp}} = \text{sparsify}(\mathbf{G}, 90)$ i.e., $\mathbf{G}$ sparsified with $s = 90$, and (c) Gray-scale image representation of correlation matrix $\mathbf{R}$ of $\mathbf{G}^{\text{sp}}$.

sparsity promoting prior i.e., a *spike and slab prior* [112] with the joint pdf

$$p(\mathbf{g}^s) = \prod_{n=1}^{K} p(g_n^s) := \prod_{n=1}^{K} [(1-\lambda_n)\delta(g_n^s) + \lambda_n f(g_n^s)]$$

Here $\lambda_n \in (0,1), \forall\ n \in [K]$ is the *sparse ratio* that models the sparsity, i.e., the probability of g_n^s being non-zero, $f(g_n^s) \forall\ n \in [K]$ models the non-zero entries of the sparse gradient $\mathbf{g}^s$, which is assumed to be a Gaussian with mean θ and variance ϕ i.e., $f(g_n^s) = \mathcal{N}(g_n^s; \theta, \phi)$. The joint pdf of the sparse gradient is then obtained as

$$p(\mathbf{g}^s) = \prod_{n=1}^{K} [(1-\lambda_n)\delta(g_n^s) + \lambda_n \mathcal{N}(g_n^s; \theta, \phi)]. \tag{3.4}$$

With the Gaussian MAC model for gradient transmission in (3.2), the exact recovery of $\mathbf{g}^s$ may not be possible due to the added noise. In this chapter, we specifically focus on the Bayesian reconstruction of the convolutional gradients with the above proposed prior. Let $\mathbf{g}^s$ be a convolutional gradient vector, then we target for the estimate $\hat{\mathbf{g}}^s$ that minimizes the mean-squared reconstruction error which is expressed as below. The MMSE estimate of $\mathbf{g}^s$ is the conditional expectation of $\mathbf{g}^s$ relative to the conditional density $p(\mathbf{g}^s|\mathbf{y}; \mathbf{q})$ where

$\mathbf{q} = (\boldsymbol{\lambda}, \theta, \phi) = (\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_K], , \theta, \phi)$ is the set of prior pdf parameters for $\mathbf{g}^s$. Therefore,

$$\hat{\mathbf{g}}^s_{\text{MMSE}} = \arg\min_{\hat{\mathbf{g}}^s} \mathbb{E}(\|\mathbf{g}^s - \hat{\mathbf{g}}^s\|_2^2) = \mathbb{E}\big[\mathbf{g}^s|\mathbf{y}; \mathbf{q}, \psi\big].$$

3.4.3 Proposed algorithm

The Gaussian MAC modeling of the channel between the users and the server gives a Gaussian likelihood model $p(\mathbf{y}|\mathbf{g}^s)$ for the Bayesian formulation of the gradient reconstruction. From (3.3), we have

$$p(\mathbf{y}|\mathbf{g}^s) = (2\pi\psi)^{-\frac{N}{2}} \exp\bigg(-\frac{1}{2\psi} \|\mathbf{y} - \mathbf{A}\mathbf{g}^s\|_2^2 \bigg), \tag{3.5}$$

where ψ is the variance of the Gaussian MAC. The exact computation of the posterior mean $\mathbb{E}\big[\mathbf{g}^s|\mathbf{y}; \mathbf{q}, \psi\big]$ may not be possible due to the complexity in its evaluation because of the coupling of coefficients of $\mathbf{g}^s$ and $\mathbf{y}$ through $\mathbf{A}$. We use the GAMP framework [113] that allows to incorporate any arbitrary priors $p(\mathbf{g}_s|\mathbf{q})$, likelihood models $p(\mathbf{y}|\mathbf{g}_s)$ and provide tractable approximations of the MMSE estimate of the sparse signal using the compressed measurements of the sparse signal. In our approach, we have the prior pdf $p(\mathbf{g}_s|\mathbf{q})$ and likelihood $p(\mathbf{y}|\mathbf{g}_s)$ in (3.4) and (3.5) respectively. We assume that we do not have prior knowledge about the statistics of the gradients i.e., (θ, ϕ), Gaussian MAC i.e., ψ, and the sparsity pattern i.e., $\boldsymbol{\lambda}$ and treat prior parameters $\mathbf{q} = (\boldsymbol{\lambda}, \theta, \phi)$, noise variance ψ as deterministic unknowns. Hence, we need to estimate these parameters in order to obtain the MMSE estimate $\mathbf{g}^s$ using the current prior and likelihood model. The expectation-maximization (EM) based GAMP (EM-GAMP) [114] approach overcomes this limitation by jointly learning the GAMP prior parameters $\mathbf{q}_{\text{in}} = (\mathbf{q}, \psi)$ along with the estimation of $\mathbf{g}^s$.

It is important to note that with the prior in (3.4), we have an individual sparse ratio $\lambda_n, n \in [K]$ as opposed to a common $\lambda_n = \lambda \ \forall n \in [K]$ in [114]. This gives us flexibility in learning the individual sparsity pattern and will be exploited while reconstructing the gradient to extract the spatial domain consistency in the gradients as shown in the sequel.

3.4. Proposed gradient reconstruction approach

The compressed measurement of the sparse gradient-sum can be expressed as

$$\mathbf{y} = \mathbf{w} + \mathbf{z}, \quad \mathbf{w} = \mathbf{A}\mathbf{g}^s.$$

From the Gaussian likelihood $p(\mathbf{y}|\mathbf{g}^s)$, we have $p(\mathbf{y}|\mathbf{w}; \mathbf{q}_{\text{in}}) = \mathcal{N}(y; w, \psi)$. GAMP approximates the true marginal posterior $p(g_n^s|\mathbf{y}; \mu_n^r, \sigma_n^r, \mathbf{q}_{\text{in}})$, $p(w_m|\mathbf{y}; \mu_m^p, \sigma_m^p, \mathbf{q}_{\text{in}})$ by

$$\begin{aligned} p(g_n^s|\mathbf{y}; \mu_n^r, \sigma_n^r, \mathbf{q}_{\text{in}}) &= \frac{p(g_n^s; \mathbf{q}_{\text{in}})\mathcal{N}(g_n^s; \mu_n^r, \sigma_n^r)}{\int_{g^s} p(g^s; \mathbf{q}_{\text{in}})\mathcal{N}(g^s; \mu_n^r, \sigma_n^r)}, \\ p(w_m|\mathbf{y}; \mu_n^p, \sigma_n^p, \mathbf{q}_{\text{in}}) &= \frac{p(y_m|w_m; \mathbf{q}_{\text{in}})\mathcal{N}(w_m; \mu_m^p, \sigma_m^p)}{\int_w p(y_m|w; \mathbf{q}_{\text{in}})\mathcal{N}(w; \mu_n^p, \sigma_n^p)}. \end{aligned}$$

Plugging the assumed prior on the sparse gradients from (3.4) and the Guassian likelihood from (3.5) in $p(g_n^s|\mathbf{y}; \mu_n^r, \sigma_n^r, \mathbf{q}_{\text{in}})$, we obtain the posterior as

$$\begin{aligned} p(g_n^s|\mathbf{y}; \mathbf{q}_{\text{in}}) &= \big((1-\lambda_n)\delta(g_n^s) + \lambda_n\mathcal{N}(g_n^s; \theta, \phi)\big)\frac{\mathcal{N}(g_n^s; \mu_n^r, \sigma_n^r)}{\zeta_n} \\ &= \frac{(1-\lambda_n)\mathcal{N}(0; \mu_n^r, \sigma_n^r)}{\zeta_n}\delta(g_n^s) + \frac{\lambda_n}{\zeta_n}\mathcal{N}(g_n^s; \mu_n^r, \sigma_n^r)\mathcal{N}(g_n^s; \theta, \phi), \end{aligned} \tag{3.6}$$

where

$$\begin{aligned} \zeta_n &= \int_{g_n^s} p(g_n^s; \mathbf{q}_{\text{in}})\mathcal{N}(s; \mu_n^s, \sigma_n^s) \\ &= (1-\lambda_n)\mathcal{N}(0; \mu_n^r, \sigma_n^r) + \lambda_n\mathcal{N}(0; \mu_n^r - \theta, \sigma_n^r + \phi), \end{aligned}$$

is the normalization constant for each $n = [K]$. The posterior is then simplified as

$$p(g_n^s|\mathbf{y}; \mathbf{q}_{\text{in}}) = (1-\pi_n)\delta(g_n^s) + \pi_n\mathcal{N}(g_n^s; \gamma_n, \nu_n), \tag{3.7}$$

where

$$\beta_n = \lambda_n\mathcal{N}(0; \theta, \sigma_n^r + \phi),$$

$$\pi_n = \frac{1}{1+\left(\frac{\beta_n}{(1-\lambda_n)\mathcal{N}(0;\mu_n^r,\sigma_n^r)}\right)^{-1}},$$
$$\gamma_n = \frac{\mu_n^r/\sigma_n^r + \theta/\phi}{1/\sigma_n^r + 1/\phi},$$
$$\nu_n = \frac{1}{1/\sigma_n^r + 1/\phi}.$$

In (3.6), π_n is the posterior sparse ratio i.e., the sparse ratio of the posterior estimate $p(g_n^s|\mathbf{y};\mathbf{q}_{\text{in}})$. The MMSE estimate $\hat{\mathbf{g}}^s_{\text{MMSE}}$ is the posterior mean and hence,

$$\hat{g}_n^s = \mathbb{E}_{G|\mathbf{Y}}\left[g_n^s|\mathbf{y};\mu_n^r,\sigma_n^r,\mathbf{q}_{\text{in}}\right] = \pi_n\gamma_n. \tag{3.8}$$

The complete algorithm for the gradient reconstruction using the GAMP framework is given in Alg. 7. The GAMP parameters $\mathbf{q}_{\text{in}}$ are learnt using the EM algorithm from the measurements $\mathbf{y}$ [114] and are obtained as

$$\psi(i+1) = \frac{1}{d}\sum_{m=1}^{d}(\sigma_m^z(i) + |y_m - \mu_m^z(i)|^2),$$
$$\theta(i+1) = \frac{\sum_{n=1}^K \pi_n(i)\gamma_n(i)}{\sum_{n=1}^K \pi_n(i)},$$
$$\phi(i+1) = \frac{\sum_{n=1}^K \pi_n(i)\big(|\theta(i)-\gamma_n(i)|^2 + \nu_n(i)\big)}{\sum_{n=1}^K \pi_n(i)},$$
$$\lambda_n(i+1) = \pi_n(i). \tag{3.9}$$

3.4.4 Sparse ratio update

Although the EM update equation (3.9) for the sparse ratio $\boldsymbol{\lambda}$ is obtained, it fails to capture the spatial domain consistency in the convolutional gradients. In sparse signals with correlation between the neighboring elements as described in Sec. 3.4.1, if the neighboring elements (g_n^s) are zero (non-zero), then there is a high probability that it will also be zero (non-zero). Inspired by the k-nearest neighbor (k-NN) classification algorithm [115, 116], we capture this inherent structure in the sparsity pattern by defining the sparse ratio update in way that $\lambda_n, n \in [K]$ is the *local average* of the posterior sparse ratio π_m of its nearest

neighboring elements g_m^s with which g_n^s is correlated. Therefore, the proposed update rule for the sparse ratio λ_n is given as

$$\lambda_n(i+1) = \frac{1}{|\mathrm{NN}(n)|} \sum_{m \in \mathrm{NN}(n)} \pi_m(i), \tag{3.10}$$

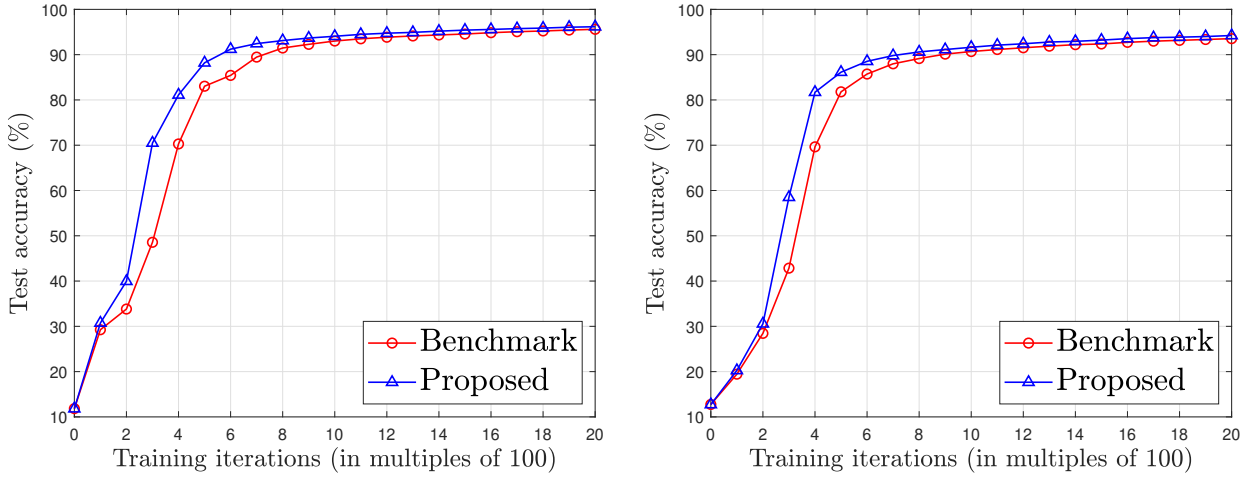
where $\mathrm{NN}(n)$ represents the set of nearest neighboring elements of g_n^s. From Sec. 3.4.1, the spatial consistency in the convolutional gradients is evident from the tensor representation of the gradient $\mathcal{G}$. Alternatively, the sparse ratio update rule can also be expressed as

$$\lambda_{hwdf}(i+1) = \frac{1}{|\mathrm{NN}(h,w,d,f)|} \sum_{\substack{(h',w',d',f') \in \\ \mathrm{NN}(h,w,d,f)}} \pi_{h'w'd'f'}(i), \tag{3.11}$$

where $\mathrm{NN}(h,w,d,f)$ is the set of indices of nearest neighboring elements of $\mathcal{G}(h,w,d,f)$ which is inspired by the spatial consistency in the convolutional gradients and is not a strict choice. An instance of appropriate choice of $\mathrm{NN}(h,w,d,f)$ is $\{(h-1,w,d,f),(h+1,w,d,f),(h,w-1,d,f),(h,w+1,d,f)\}$ where we chose the set $\mathrm{NN}(h,w,d,f)$ as the indices that are at distance of 1 from $\mathcal{G}(h,w,d,f)$. Note that if the neighboring elements set is chosen as the set of all entries in the convolutional gradient, then the proposed algorithm reduces to EM-BG-GAMP algorithm [114], which fails to capture the correlation of the convolutional gradients with its nearest neighboring gradient elements as it averages out the sparse ratio for the entire gradient.

3.5 Experiments

We evaluate the performance of the FL procedure in Alg. 5 with the proposed gradient reconstruction technique in Alg. 7 for the task of digit classification. We run experiments on the MNIST dataset [117] with 60000 training and 10000 test data samples and train a CNN utilizing an SGD optimizer. We use a CNN with three convolutional layers (with ReLu activation layer) with $(H,W,D,F) = (5,5,1,5),(5,5,5,10),(5,5,10,20)$ connected in the same order, followed by a 4×4 maxpooling layer, a fully-connected layer with 320 neurons and final softmax output layer with 10 neurons. We set the SNR of the Gaussian MAC



(a) $(s, N, |\mathcal{B}|, K/d) = (75, 1, 20000, 2)$ (b) $(s, N, |\mathcal{B}|, K/d) = (75, 10, 2000, 2)$

Figure 3.5: Test accuracy vs training iterations with different $(s, N, |\mathcal{B}|, K/d)$.

to 25dB. The performance is measured as the accuracy with respect to the test dataset, called test accuracy, versus training iteration count t. We consider i.i.d. distribution of data across users i.e., $|\mathcal{B}|$ training data samples are selected at random from the dataset and assigned to each user at the beginning of training. At each iteration, users use all the $|\mathcal{B}|$ local data samples to compute their gradient estimates, i.e., the batch size is equal to the size of the local datasets. We compare the performance of the proposed method with the A-DSGD from [45] for different values of $(s, N, |\mathcal{B}|, K/d)$. We observe that the proposed method outperforms the benchmark method [45] in all the cases, which is expected due to the Bayesian modeling of the gradients in the proposed method.

3.6 Summary

In this chapter, we have studied an FL system with distributed wireless users connected to a central server through a Gaussian MAC which implements DSGD to jointly train a CNN at the server. We also utilized techniques like gradient sparsification, error accumulation for gradient communication in this FL framework. We used a well-known compression technique using CS for reducing the dimension of the sparse gradients and then transmit them over the MAC. We identified and demonstrated spatial consistency in the convolutional gradients that

is exploited to design a novel gradient reconstruction algorithm using the GAMP framework through an appropriate prior on the gradients. Through the numerical simulations, we showed that the proposed reconstruction method achieves a faster convergence of the global model compared to the existing counterpart that does not exploit the spatial consistency in the convolutional gradients.

4

Conclusion

4.1 Summary

In this thesis, we presented two novel applications of tensors to ML for wireless communications where the tensor representation of the data can be leveraged to improve the learning task. Although there has been prior literature studying these problems, the introduction of tensors to these applications is novel. We identified and rearranged the concerned data in a meaningful way to obtain a natural tensor representation. We also showed how the tensor structure in the data is beneficial compared to the existing techniques in learning better ML models. It is also important to note that the tools used for leveraging the tensor structure in both cases are different and customized to the applications. Our key contributions in both these works are summarized below.

In Chapter 2, we explored a classical problem of precoder codebook design in FDD FD-MIMO systems. Given a dataset of channel realizations, this problem comes under the purview of the application of ML for PHY layer communications. We identified that the FD-MIMO channel can be meaningfully rearranged to a 3D tensor. Using this tensor representation of the channel, we proposed a precoder that is represented as an element in a product manifold called CPM. This product representation allows us to construct codebooks in the factor manifolds and we show that finding the codebooks in the factor manifolds is equivalent to K-means clustering in the factor GMs with a chordal distance metric. We showed through numerical simulations that learning the proposed tensor-based precoder codebooks has significantly lower complexity than the existing counterparts such as VQ-based methods without compromising much on their performance.

In Chapter 3, we explored gradient communication (compression-reconstruction) for FL with DSGD at the users and the servers to collaboratively train a CNN. We also incorporated

commonly used techniques such as gradient sparsification, error accumulation to aid the gradient communication in FL. We further studied this problem by considering the tensor structure offered by the convolutional gradients and identifying the spatial consistency among neighboring gradient entries across different dimensions of the gradient tensor. We exploited this correlation in the gradients to impose a novel prior for modeling the convolutional gradients and designed a Bayesian gradient reconstruction algorithm by adopting the GAMP framework to capture the sparsity pattern in the gradients. The numerical results showed that the convergence of the CNN model at the server was faster than the existing counterpart which does not exploit the tensor structure of the convolutional gradients, which arises from the fact that the proposed prior appropriately models the sparsity and spatial consistency that stems from the tensor structure of the convolutional gradients.

4.2 Future directions

Given the versatility of tensors and ML, we find various applications where tensors could be useful in solving problems in ML for wireless communications. However, we now briefly discuss future research directions and possible extensions of the two applications presented in this thesis.

A tensor representation of the FD-MIMO channel motivated the design of the low-complexity precoder codebooks, which was presented in Chapter 2. However, we assumed a stationary channel distribution for the available channel dataset. Considering the increasing number of high-speed mobile devices, an interesting follow-up work would be to extend the codebooks to non-stationary FD-MIMO channels. Further, since the codebooks were designed for a narrow-band FD-MIMO system, an interesting extension would be to design such low-complexity precoder codebooks for wide-band MIMO systems where the carrier frequency is a potential additional dimension to the tensor representation of the channel. From the 3GPP perspective, a relevant extension of this work would be developing codebooks for dual-polarized FD-MIMO systems.

Furthermore, in Chapter 3, we demonstrated that a tensor representation of convolutional gradients motivated the novel spike and slab prior, in which the non-zero entries of the sparse gradient are modeled as a Gaussian distribution. From a modeling perspective,

a worthwhile extension of this work is to impose a more flexible prior such as a Gaussian mixture model to model the non-zero entries of the sparse gradient. The gradients of a NN, including CNN, evolve slowly over training iterations which is likely due to steady gradients and reasonable learning rate. A simple follow-up work would be to characterize the correlation between convolutional gradients across training iterations using an appropriate prior. An interesting extension would be to design a prior that captures both the spatial consistency and the temporal correlation in the convolutional gradients together and develop a corresponding Bayesian reconstruction algorithm

Bibliography

[1] J. C. Roh and B. D. Rao, "Design and analysis of MIMO spatial multiplexing systems with quantized feedback," *IEEE Trans. on Signal Process.*, vol. 54, no. 8, pp. 2874–2886, 2006.

[2] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," in *Proc., ITA*, Feb 2019, pp. 1–8.

[3] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[4] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[5] G. Beylkin and M. J. Mohlenkamp, "Algorithms for numerical analysis in high dimensions," *SIAM J. on Scientific Computing*, vol. 26, no. 6, pp. 2133–2159, 2005.

[6] P. Comon, X. Luciani, and A. L. F. de Almeida, "Tensor decompositions, alternating least squares and other tales," *J. of Chemometrics: A J. of the Chemometrics Society*, vol. 23, no. 7-8, pp. 393–405, 2009.

[7] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[8] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. of Mathematics and Physics*, vol. 6, no. 1-4, pp. 164–189, 1927.

[9] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.

[10] L. R. Tucker *et al.*, "The extension of factor analysis to three-dimensional matrices," *Contributions to Mathematical Psychology*, vol. 110119, 1964.

[11] A. Smilde, R. Bro, and P. Geladi, *Multi-way analysis: Applications in the chemical sciences*. John Wiley & Sons, 2005.

[12] R. Bro, "Multiway calibration. Multilinear PLS," *J. of Chemometrics*, vol. 10, no. 1, pp. 47–61, 1996.

[13] R. Yu and Y. Liu, "Learning from multiway data: Simple and efficient tensor regression," in *Int. Conf. on Mach. Learn.* PMLR, 2016, pp. 373–381.

[14] A. H. Phan and A. Cichocki, "Tensor decompositions for feature extraction and classification of high dimensional datasets," *Nonlinear Theory and its Applications, IEICE*, vol. 1, no. 1, pp. 37–68, 2010.

[15] N. D. Sidiropoulos and A. Kyrillidis, "Multi-way compressed sensing for sparse low-rank tensors," *IEEE Signal Process. Letters*, vol. 19, no. 11, pp. 757–760, 2012.

[16] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, "Parallel factor analysis in sensor array processing," *IEEE Trans. on Signal Process.*, vol. 48, no. 8, pp. 2377–2388, 2000.

[17] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. on Signal Process.*, vol. 65, no. 13, pp. 3551–3582, 2017.

[18] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, 2015.

[19] L. H. Lim and P. Comon, "Multiarray signal processing: Tensor decomposition meets compressed sensing," *Comptes Rendus Mecanique*, vol. 338, no. 6, pp. 311–320, 2010.

[20] D. C. Araújo and A. L. de Almeida, "Tensor-based compressed estimation of frequency-selective mmwave MIMO channels," in *IEEE 7th Int. Workshop on Computational Advances in Multi-Sensor Adaptive Process*, 2017, pp. 1–5.

BIBLIOGRAPHY

[21] A. L. F. De Almeida, "Tensor modeling and signal processing for wireless communication systems," Ph.D. dissertation, Université de Nice Sophia Antipolis, 2007.

[22] K. Naskovska, "Advanced tensor based signal processing techniques for wireless communication systems and biomedical signal processing," Ph.D. dissertation, 2019.

[23] M. N. Da Costa, "Tensor space–time coding for MIMO wireless communication systems," Ph.D. dissertation, Université Nice Sophia Antipolis; Universidade estadual de Campinas (Brésil), 2014.

[24] W. Guo, I. Kotsia, and I. Patras, "Tensor learning for regression," *IEEE Trans. on Image Process.*, vol. 21, no. 2, pp. 816–827, 2011.

[25] D. Tao, X. Li, W. Hu, S. Maybank, and X. Wu, "Supervised tensor learning," in *IEEE Int. Conf. on Data Mining*, 2005, pp. 8–pp.

[26] X. Tan, Y. Zhang, S. Tang, J. Shao, F. Wu, and Y. Zhuang, "Logistic tensor regression for classification," in *Int. Conf. on Intell. Science and Intell. Data Eng.* Springer, 2012, pp. 573–581.

[27] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recognition*, vol. 44, no. 7, pp. 1540–1551, 2011.

[28] S. Jegelka, S. Sra, and A. Banerjee, "Approximation algorithms for tensor clustering," in *Int. Conf. on Algorithmic Learn. Theory.* Springer, 2009, pp. 368–383.

[29] A. Shashua, R. Zass, and T. Hazan, "Multi-way clustering using super-symmetric non-negative tensor factorization," in *European Conf. on Computer Vision.* Springer, 2006, pp. 595–608.

[30] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. on Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.

[31] Q. Shi, Y.-M. Cheung, Q. Zhao, and H. Lu, "Feature extraction for incomplete data via low-rank tensor decomposition with feature regularization," *IEEE Trans. on Neural Networks and Learn. Systems*, vol. 30, no. 6, pp. 1803–1817, 2018.

[32] A. Scaglione, P. Stoica, S. Barbarossa, G. B. Giannakis, and H. Sampath, "Optimal designs for space–time linear precoders and decoders," *IEEE Trans. on Signal Process.*, vol. 50, no. 5, pp. 1051–1064, 2002.

[33] D. J. Love and R. W. Heath, "Limited feedback unitary precoding for spatial multiplexing systems," *IEEE Trans. on Inf. Theory*, vol. 51, no. 8, pp. 2967–2976, Aug 2005.

[34] ——, "Limited feedback diversity techniques for correlated channels," *IEEE Trans. on Veh. Tech.*, vol. 55, no. 2, pp. 718–722, Mar 2006.

[35] D. J. Love, R. W. Heath, V. K. N. Lau, D. Gesbert, B. D. Rao, and M. Andrews, "An overview of limited feedback in wireless communication systems," *IEEE J. on Sel. Areas in Commun.*, vol. 26, no. 8, pp. 1341–1365, Oct 2008.

[36] R. E. Bellman, *Adaptive control processes: A guided tour*. Princeton University Press, 2015, vol. 2045.

[37] C. Wen, W. Shih, and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Commun. Letters*, vol. 7, no. 5, pp. 748–751, Oct 2018.

[38] T. Wang, C. Wen, S. Jin, and G. Y. Li, "Deep learning–based CSI feedback approach for time-varying massive MIMO channels," *IEEE Wireless Commun. Letters*, vol. 8, no. 2, pp. 416–419, Apr 2019.

[39] H. B. McMahan, E. Moore, D. Ramage, and B. A. Y Arcas, "Federated learning of deep networks using model averaging," 2016, available online: arXiv/abs/1602.05629.

[40] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "TernGrad: Ternary gradients to reduce communication in distributed deep learning," in *Proc., Advances in Neural Information Processing Systems*, 2017, pp. 1509–1519.

BIBLIOGRAPHY

[41] J. Bernstein, Y. X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non–convex problems," in *International Conference on Machine Learning*. PMLR, 2018, pp. 560–569.

[42] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication–efficient SGD via gradient quantization and encoding," in *Proc., Advances in Neural Information Processing Systems*, 2017, pp. 1709–1720.

[43] D. Jhunjhunwala, A. Gadhikar, G. Joshi, and Y. C. Eldar, "Adaptive quantization of model updates for communication–efficient federated learning," in *Proc., ICASSP*, 2021, pp. 3110–3114.

[44] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1–bit stochastic gradient descent and its application to data–parallel distributed training of speech DNNs," in *Annu. Conf. of the Int. Speech Commun. Assoc.*, 2014.

[45] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. on Signal Process.*, vol. 68, pp. 2155–2169, 2020.

[46] S. Dörner, S. Cammerer, J. Hoydis, and S. T. Brink, "Deep learning based communication over the air," *IEEE J. of Sel. Topics in Signal Process.*, vol. 12, no. 1, pp. 132–143, 2017.

[47] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. on Cognitive Commun. and Netw.*, vol. 3, no. 4, pp. 563–575, 2017.

[48] A. Karpatne *et al.*, "Theory–guided data science: A new paradigm for scientific discovery from data," *IEEE Trans. on Knowledge and Data Eng.*, vol. 29, no. 10, pp. 2318–2331, 2017.

[49] A. Narula, M. J. Lopez, M. D. Trott, and G. W. Wornell, "Efficient use of side information in multiple-antenna data transmission over fading channels," *IEEE J. on Sel. Areas in Commun.*, vol. 16, no. 8, pp. 1423–1436, Oct 1998.

[50] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1, no. 2.

[51] D. J. Love, R. W. Heath, and T. Strohmer, "Grassmannian beamforming for multiple–input multiple–output wireless systems," *IEEE Trans. on Inf. Theory*, vol. 49, no. 10, pp. 2735–2747, Oct 2003.

[52] K. Amiri, D. Shamsi, B. Aazhang, and J. R. Cavallaro, "Adaptive codebook for beamforming in limited feedback MIMO systems," in *Proc., 42nd Annu. Conf. on Inf. Sciences and Systems*, 2008, pp. 994–998.

[53] D. P. McNamara, M. A. Beach, and P. N. Fletcher, "Spatial correlation in indoor MIMO channels," in *Proc., IEEE PIMRC*, vol. 1, 2002, pp. 290–294.

[54] *Spatial channel model for Multiple Input Multiple Output (MIMO) simulations*, 3GPP TR 25.996, 2003.

[55] T. Shuang, T. Koivisto, H. L. Maattanen, K. Pietikainen, T. Roman, and M. Enescu, "Design and evaluation of LTE–Advanced double codebook," in *IEEE 73rd Veh. Technol. Conf.*, 2011, pp. 1–5.

[56] D. Ying, F. W. Vook, T. A. Thomas, D. J. Love, and A. Ghosh, "Kronecker product correlation model and limited feedback codebook design in a 3D channel model," in *Proc., IEEE ICC*, Aug 2014, pp. 5865–5870.

[57] J. Li, X. Su, J. Zeng, Y. Zhao, S. Yu, L. Xiao, and X. Xu, "Codebook design for uniform rectangular arrays of massive antennas," in *Proc., IEEE 77th Veh. Technol. Conf.*, 2013, pp. 1–5.

[58] X. Su, J. Zeng, J. Li, L. Rong, L. Liu, X. Xu, and J. Wang, "Limited feedback precoding for massive MIMO," *Int. J. of Antennas and Propagation*, vol. 2013, Oct 2013.

[59] J. Song, J. Choi, T. Kim, and D. J. Love, "Advanced quantizer designs for FDD–based FD-MIMO systems using uniform planar arrays," *IEEE Trans. on Signal Process.*, vol. 66, no. 14, pp. 3891–3905, 2018.

[60] J. Choi, K. Lee, D. J. Love, T. Kim, and R. W. Heath, "Advanced limited feedback designs for FD-MIMO using uniform planar arrays," in *Proc., IEEE GLOBECOM*, 2015, pp. 1–6.

[61] Q. Yang, M. B. Mashhadi, and D. Gündüz, "Deep convolutional compression for massive MIMO CSI feedback," in *Proc., IEEE Int. Workshop on Mach. Learn. for Signal Process.*, 2019, pp. 1–6.

[62] J. Guo, C. K. Wen, S. Jin, and G. Y. Li, "Convolutional neural network-based multiple-rate compressive sensing for massive MIMO CSI feedback: Design, simulation, and analysis," *IEEE Trans. on Wireless Commun.*, vol. 19, no. 4, pp. 2827–2840, 2020.

[63] D. C. Araújo, A. L. F. de Almeida, J. P. C. L. da Costa, and R. T. de Sousa, "Tensor–based channel estimation for massive MIMO–OFDM systems," *IEEE Access*, vol. 7, pp. 42 133–42 147, 2019.

[64] A. de Baynast, L. De Lathauwer, and B. Aazhang, "Blind PARAFAC receivers for multiple access–multiple antenna systems," in *2003 IEEE 58th Veh. Technol. Conf.*, vol. 2, 2003, pp. 1128–1132.

[65] A. L. F. de Almeida, G. Favier, and J. C. M. Mota, "Space-time multiplexing codes: A tensor modeling approach," in *IEEE Workshop on Signal Process. Advances in Wireless Commun.*, 2006, pp. 1–5.

[66] A. Cichocki, D. Mandic, L. D. Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two–way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, 2015.

[67] Y. M. Lui, "Human gesture recognition on product manifolds," *JMLR*, vol. 13, no. 1, pp. 3297–3321, 2012.

[68] C. H. Tse, K. W. Yip, and T. S. Ng, "Performance tradeoffs between maximum ratio transmission and switched–transmit diversity," in *Proc., IEEE PIMRC*, vol. 2, 2000, pp. 1485–1489.

[69] J. B. Andersen, "Antenna arrays in mobile communications: Gain, diversity, and channel capacity," *IEEE Antennas and Propag. Mag.*, vol. 42, no. 2, pp. 12–16, Apr 2000.

[70] M. K. Simon and M. S. Alouini, *Digital communication over fading channels*. John Wiley & Sons, 2005, vol. 95.

[71] T. M. Cover, *Elements of Information Theory*. John Wiley & Sons, 1999.

[72] D. Kapetanovic and F. Rusek, "A comparison between unitary and non–unitary precoder design for MIMO channels with MMSE detection and limited feedback," in *Proc., IEEE GLOBECOM*, 2010, pp. 1–6.

[73] A. Kapteyn, H. Neudecker, and T. Wansbeek, "An approach ton–mode components analysis," *Psychometrika*, vol. 51, no. 2, pp. 269–275, 1986.

[74] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[75] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. on Matrix Anal. and Appl.*, vol. 20, no. 2, pp. 303–353, 1998.

[76] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Commun.*, vol. 28, no. 1, pp. 84–95, Jan 1980.

[77] B. Mondal, S. Dutta, and R. W. Heath, "Quantization on the Grassmann manifold," *IEEE Trans. on Signal Process.*, vol. 55, no. 8, pp. 4208–4216, Aug 2007.

[78] G. L. Nemhauser and L. A. Wolsey, "Best algorithms for approximating the maximum of a submodular set function," *Mathematics of Operations Res.*, vol. 3, no. 3, pp. 177–188, 1978.

[79] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions–I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[80] J. C. Roh and B. D. Rao, "Transmit beamforming in multiple–antenna systems with finite rate feedback: A VQ-based approach," *IEEE Trans. on Inf. Theory*, vol. 52, no. 3, pp. 1101–1112, 2006.

[81] A. Konar and N. D. Sidiropoulos, "Greed is good: Leveraging submodularity for antenna selection in massive MIMO," in *Proc., IEEE Asilomar Conf. on Signals, Systems, and Computers*, 2017, pp. 1522–1526.

[82] O. Curtef, G. Dirr, and U. Helmke, "Riemannian optimization on tensor products of Grassmann manifolds: Applications to generalized Rayleigh–quotients," *SIAM J. on Matrix Anal. and Appl.*, vol. 33, no. 1, pp. 210–234, 2012.

[83] K. Bhogi, C. Saha, and H. S. Dhillon, "Learning on a Grassmann manifold: CSI quantization for massive MIMO systems," in *Proc., IEEE Asilomar Conf. on Signals, Systems, and Computers*, 2020, pp. 179–186.

[84] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proc., of the IEEE*, vol. 107, no. 11, pp. 2204–2239, 2019.

[85] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Toward an intelligent edge: Wireless communication meets machine learning," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 19–25, 2020.

[86] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intell. and Statist.* PMLR, 2017, pp. 1273–1282.

[87] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, 2020.

[88] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra–reliable low–latency vehicular communications," *IEEE Trans. on Commun.*, vol. 68, no. 2, pp. 1146–1159, 2019.

[89] H. Shiri, J. Park, and M. Bennis, "Communication–efficient massive UAV online path control: Federated learning meets mean-field game theory," 2020, available online: arXiv/abs/2003.04451.

[90] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.

[91] A. Ferdowsi and W. Saad, "Generative adversarial networks for distributed intrusion detection in the internet of things," in *Proc., IEEE GLOBECOM*, 2019, pp. 1–6.

[92] M. Chen, O. Semiari, W. Saad, X. Liu, and C. Yin, "Federated echo state learning for minimizing breaks in presence in wireless virtual reality networks," *IEEE Trans. on Wireless Commun.*, vol. 19, no. 1, pp. 177–191, 2019.

[93] T. Zeng, O. Semiari, M. Mozaffari, M. Chen, W. Saad, and M. Bennis, "Federated learning in the sky: Joint power allocation and scheduling with UAV swarms," in *Proc., IEEE ICC*, 2020, pp. 1–6.

[94] T. Zeng, J. Guo, K. J. Kim, K. Parsons, P. V. Orlik, S. Di Cairano, and W. Saad, "Multi–task federated learning for traffic prediction and its application to route planning," in *IEEE Intell. Vehicles Symp.*

[95] T. Zeng, O. Semiari, M. Chen, W. Saad, and M. Bennis, "Federated learning on the road: Autonomous controller design for connected and autonomous vehicles," 2021, available online: arXiv/abs/2102.03401.

[96] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. on Commun.*, vol. 68, no. 1, pp. 317–333, 2019.

[97] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Distributed learning in wireless networks: Recent progress and future challenges," 2021, available online: arXiv:/abs/2104.02151.

[98] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," 2019, available online: arXiv/abs/1912.04977.

[99] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc., IEEE INFOCOM*, 2019, pp. 1387–1395.

[100] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, available online: arXiv/abs/1610.05492.

[101] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proc., EMNLP*. Association for Computational Linguistics, Sep. 2017, pp. 440–445.

[102] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," 2017, available online: arXiv/abs/1712.01887.

[103] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," 2018, available online: arXiv/abs/1809.07599.

[104] N. Strom, "Scalable distributed DNN training using commodity GPU cloud computing," in *Annu. Conf. of the Int. Speech Commun. Assoc.*, 2015.

[105] H. Wang, S. Sievert, Z. Charles, S. Liu, S. Wright, and D. Papailiopoulos, "ATOMO: Communication–efficient learning via atomic sparsification," 2018, available online: arXiv/abs/1806.04090.

[106] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli, "The convergence of sparsified gradient methods," in *Proc., Advances in Neural Information Processing Systems*, 2018, pp. 5973–5983.

[107] M. M. Amiri and D. Gündüz, "Over-the-air machine learning at the wireless edge," in *Proc., IEEE SPAWC*, 2019, pp. 1–5.

[108] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over–the–air computation," *IEEE Trans. on Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, 2020.

[109] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low–latency federated edge learning," *IEEE Trans. on Wireless Commun.*, vol. 19, no. 1, pp. 491–506, 2019.

[110] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. on Wireless Commun.*, vol. 20, no. 1, pp. 269–283, 2020.

[111] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. on Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, 2020.

[112] A. E. Raftery, D. Madigan, and J. A. Hoeting, "Bayesian model averaging for linear regression models," *J. of the Amer. Statistical Assoc.*, vol. 92, no. 437, pp. 179–191, 1997.

[113] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc., IEEE ISIT*, 2011, pp. 2168–2172.

[114] J. P. Vila and P. Schniter, "Expectation–maximization Gaussian–mixture approximate message passing," *IEEE Trans. on Signal Process.*, vol. 61, no. 19, pp. 4658–4672, 2013.

[115] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. on Inf. Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[116] X. Meng, S. Wu, L. Kuang, J. Lu *et al.*, "Approximate message passing with nearest neighbor sparsity pattern learning," 2016, available online: arxiv.org/abs/1601.00543.

[117] Y. LeCun, "The MNIST database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.