Robust Learning in Wireless Networks: Efficacy of Models and Prior Knowledge in Learning from Small Sample Sets

vorgelegt von
M.Sc.
Daniyal Amir Awan

an der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Giuseppe Caire
Gutachter: Prof. Dr.-Ing. Slawomir Stanczak
Gutachter: Prof. Vincent Poor
Gutachter: Prof. Osvaldo Simeone
Gutachter: Dr. Renato Cavalcante

Tag der wissenschaftlichen Aussprache: 17. Dezember 2020

Berlin 2021

Abstract

This work studies robust learning in dynamic wireless environments. Modern wireless data networks are complex and modeling their behavior accurately is difficult. As a result, machine learning and artificial intelligence have been making significant inroads into wireless networks. State-of-the-art machine learning algorithms (e.g., neural networks) assume a stationary learning environment and they generally require large training sets. However, modern radio access networks (RANs) are dynamic, and by the time a large training set is collected the environment may have changed so much as to render the learning useless. Therefore, in dynamic networks learning frameworks must work with small training sets. Assuming that each training sample is informative, the lack of a large training set results in uncertainty about the underlying phenomenon/function to be learned. In light of these facts, we study "hybrid" learning approaches in which the above-mentioned uncertainty is combated by the inclusion of model based prior knowledge in the proposed learning frameworks. In Chapter 2 we study cell-load approximation in RANs using a small sample set and a robust learning framework armed with model based prior knowledge. To this end, we study the nonlinear load-coupling model and prove some salient properties of cell-load as a function of user rates. We show how this prior knowledge can be used to decrease the uncertainty resulting from a small training set. In Chapter 3 we study robust multiuser detection in dynamic wireless networks in which users transmit sporadically. Though it is known that the optimal multiuser detector is nonlinear, learning this detector using conventional methods requires a large number of training samples. Additionally, all nonlinear detectors are sensitive to small changes in the environment. In modern wireless applications, such as machine-type communications, users transmit sporadically and as a result performance of nonlinear detectors may deteriorate. To address this issue, we propose a novel online learning framework that combines the expressive power of a nonlinear filter with the robustness of a linear filter. The proposed "sum filter" is designed in a reproducing kernel Hilbert space (RKHS) constructed by taking the direct sum of an RKHS associated with a linear kernel and an RKHS associated with a nonlinear kernel. We derive the nonlinear kernel from the multiuser detection model by exploiting the connection between the optimal nonlinear filter and certain RKHSs. Working in RKHSs and, in general, Hilbert spaces allows for low-complexity projection based algorithms which are well-known for their robustness to noise and their numerical stability. In Chapter 4 we use the celebrated projection onto convex sets (POCS) technique to learn probability density functions (pdfs) in a Hilbert space. Here again, we combine a small training sample set with prior knowledge based on general properties of pdfs. We then show how to apply our learning framework to distributed multiuser detection in a cloud RAN network.

Zusammenfassung

Diese Arbeit untersucht robustes Lernen in dynamischen Umgebungen der drahtlosen Datenkommunikation. Moderne drahtlose Datennetze sind komplex und die genaue Modellierung ihres Verhaltens ist schwierig. Infolgedessen haben maschinelles Lernen und künstliche Intelligenz einen bedeutenden Einzug in drahtlose Netzwerke gehalten. Moderne Lernalgorithmen (z.B. neuronale Netze) setzen aber eine stationäre Lernumgebung voraus und erfordern in der Regel große Trainingssätze. Funkzugangsnetze (*RANs*) sind jedoch dynamisch, und wenn ein großer Trainingssatz gesammelt wurde, kann sich die Umgebung so stark verändert haben, dass das gelernte Modell unbrauchbar wird. Daher muss in dynamischen Netzwerken mit kleinen Trainingssätzen gearbeitet werden. Unter der Annahme, dass jeder Trainingsdatenpunkt informativ ist, führt das Fehlen eines großen Trainingssatzes zu Unsicherheit über das zu lernende Phänomen/Funktion. Vor diesem Hintergrund untersucht diese Arbeit "hybride" Lernansätze, bei denen die oben erwähnte Unsicherheit durch die Einbeziehung von modellbasiertem Vorwissen reduziert wird. In Kapitel 2 wird die Approximation der Zellenlast in RANs unter Verwendung eines kleinen Trainingssatzes und einer robusten Lernmethode untersucht, das mit modellbasiertem Vorwissen ausgestattet ist. Zu diesem Zweck wird das nichtlineare Lastkopplungsmodell untersucht und einige hervorstechende Eigenschaften der Funkzellenlast als Funktion der Benutzerraten bewiesen. Es wird gezeigt wie dieses Vorwissen genutzt werden kann, um die aus einem kleinen Trainingssatz resultierende Unsicherheit zu verringern. In Kapitel 3 wird die robuste Mehrbenutzer-Demodulation in dynamischen drahtlosen Netzwerken, in denen Benutzer sporadisch senden, untersucht. Obwohl bekannt ist, dass der optimale Mehrbenutzer-Demodulator nichtlinear ist, erfordert das Erlernen dieses Detektors mit herkömmlichen Methoden einen großen Trainingssatz. Darüber hinaus sind alle nichtlinearen Detektoren empfindlich in Bezug auf kleine Änderungen in ihrer Umgebung. In modernen drahtlosen Anwendungen, wie z.B. in der maschinellen Kommunikation, senden die Benutzer sporadisch und als Folge davon kann sich die Leistung der nichtlinearen Detektors verschlechtern. Um dieses Problem zu lösen, wird eine neuartige Online-Lernmethode vorgeschlagen, das die Ausdruckskraft eines nichtlinearen Filters mit der Robustheit eines linearen Filters kombiniert. Der vorgeschlagene "Summenfunktion" ist in einem reproduzierenden Hilbertraum (RKHS) konstruiert, der aus der direkten Summe eines RKHS, der

mit einem linearen Kernel assoziiert ist, und eines RKHS, der mit einem nichtlinearen Kernel assoziiert ist, besteht. Der nichtlineare Kernel wird aus dem Mehrbenutzermodell abgeleitet, indem eine Zusammenhang zwischen dem optimalen nichtlinearen Filter und gewissen RKHSs ausgenutzt wird. RKHSs und im Allgemeinen Hilberträumen ermöglichen die Konstruktion von auf Projektion basierenden Algorithmen mit geringer Komplexität, die für ihre Robustheit gegenüber Rauschen und ihre numerische Stabilität bekannt sind. In Kapitel 4 wird die berühmte Projektion auf konvexe Mengen (POCS) Methode verwendet, um Wahrscheinlichkeitsdichtefunktionen in einem Hilbertraum zu lernen. Auch hier wird ein kleiner Trainingssatz mit modellbasierten Vorwissen kombiniert, welches auf den allgemeinen Eigenschaften von Wahrscheinlichkeitsdichtefunktionen basiert. Anschließend wird gezeigt, wie diese Lernmethode auf die verteilte Mehrbenutzer-Demodulation in einem *Cloud*-RAN angewendet werden kann.

Acknowledgements

First of all, I want to thank my advisor Prof. Dr.-Ing. Slawomir Stanczak for giving me the opportunity to work with him. I am grateful for the scientific freedom and support during my Ph.D. I am also especially grateful to Dr. Renato Cavalcante for his support and guidance throughout my Ph.D.

I am grateful to Prof. Vincent Poor and Prof. Osvaldo Simeone for serving as referees of this dissertation. Moreover, I would like to thank Prof. Giuseppe Caire for serving as the chairman of the dissertation committee.

I would like to thank my coauthors Prof. Masahiro Yukawa, Dr. Zoran Utkovski, Dr. Qi Liao, Matthias Mehlhose, and Martin Kurras for our collaborations. My colleagues at Heinrich Hertz Institute (HHI) and my former colleagues and friends at Technical University of Berlin (TU Berlin) have always been very helpful and kind. I would like to take this opportunity to thank them.

Contents

1 Introduction

1.1 Motivation

Driven by new applications and advances in electronics, commercial cellular wireless networks have evolved significantly since their humble beginnings in late 1970s. The internet started out being delivered by wired phone and cable networks. But nowadays internet and high-speed data is integral to wireless networks. As the growth of wireless infrastructure allows customers to enjoy ever higher data rates and connectivity, the modeling of these networks is becoming harder. Users are almost always connected to a wireless network, but data transmission in many applications (e.g., internet surfing and video streaming) is bursty in nature and it is initiated randomly. It is well known that, in contrast to voice traffic, data traffic within a small area (e.g., a cell or a sector) is hard to model and predict. Models allow engineers to study network behavior and optimize future behavior according to objectives, such as high data capacity, high connectivity, low transmission delay, and low energy consumption, to name a few. Unfortunately, the evolution of some aspects of wireless networks has been faster than the development of accurate models for them. The reason is that many models are developed based on our understanding of how networks react to user behavior in the real world. However, behavior of users (e.g., initiation of a video stream) is difficult to model, especially in new data applications. The lack of accurate models is galling enough for network providers and engineers to look for model-less data-driven explanation of phenomenon in wireless networks. Luckily, the tools required in this case, i.e., machine learning and artificial intelligence (AI) have seen an unprecedented rise in recent years due to increase in computational power. To some extent, the lack of accurate models is being compensated by the computation power of AI machines and availability of historical network performance data.

Machine learning has been around since the advent of function approximation or parameter estimation by computers. However, recently the terms “machine learning” and AI have been seized by neural networks, especially *deep learning networks.* Deep learning combined with *big data* has allowed engineers to solve forecasting and complex optimization problems in various fields, without necessarily understanding some aspects of these frameworks mathematically. As a result, deep learning is gradually replacing model-based approximation and optimization in many engineering fields. A typical example is that of energy optimization in Google's data centers [EG16]. However, it is generally accepted in the machine learning community that neural networks and deep learning require large sets of offline training data [Mar18]. This data can be in some cases shared and reused, and huge data collection in many applications is not a problem. In fact it has been observed that, for sufficiently large data sets, deep neural networks outperform classical (nonneural network based) approximation algorithms. However, in wireless networks, especially radio access networks (RANs), large data sets may not be available. The reason is that, modern wireless environments can be considered roughly constant only for a short time [typically milliseconds to seconds], which can be all the time available to collect training data, train a machine learning algorithm, and then perform the communication task. If this temporal aspect is not taken into account, then by the time enough samples are available to train existing state-of-the-art learning algorithms, the environment may have changed so drastically as to render the training set useless for the current propagation conditions, even if we ignore the complexity of the training process which can be somewhat compensated by extra computational power.

1.2 Contributions and Outline

This thesis studies "hybrid" learning techniques in the sense that the gap in knowledge about the underlying phenomenon, resulting from the unavailability of a large training set in a dynamic wireless network, is filled by extracting prior knowledge from known models. Such prior knowledge may consist of properties, such as monotonicity, continuity, positivity, and membership to certain sets etc., of the underlying function to be approximated. In other words, we use well-known models only to extract reliable qualitative information and fundamental insights about the phenomenon that we are interested in learning. Note that explicit inclusion of prior knowledge in current learning frameworks is in fact not always encouraged or advisable. This is especially true for deep learning neural networks due to a lack of sufficient mathematical understanding of some aspects of these frameworks [Mar18]. Nevertheless, as assumed in these frameworks, if the size of the data set is sufficiently large then one can reasonably expect that a powerful learning machine can infer this prior knowledge from the data. In fact explicit inclusion of prior knowledge can make neural networks inflexible [Mar18]. However, in situations where a large data set is not available, such as those considered in this thesis, prior knowledge should be included if it is accurate, and if its efficacy in learning is clearly understood.

In the following we present the main contributions of this thesis that are divided into three chapters. In this thesis we aim to be self-contained and we provide mathematical proofs wherever they are necessary. Important aspects of the presented learning techniques are that they are numerically robust, they have a low complexity, and they are well-understood mathematically. From an algorithmic point-of-view, the proposed approximation techniques are deterministic and they are particular cases of classical *bounded error estimation* [Wit68,Sch68,NG95] (also known as *set-membership estimation* [MV91,Cas02] and *robust estimation* [MT85]) (which is presented briefly in Section 1.7). Simulations are performed at the end of each chapter and comparisons with other state-of-art methods are done wherever appropriate.

In **Chapter 2** we study the problem of cell-load approximation in wireless networks. Cell-load forecast is an important part of network optimization because it provides an estimate of how much cell traffic will be seen in a given time period as a function of, e.g., user rate demand. However, learning of the cell-load in RANs has to be performed within a short time period. Therefore, we propose a learning framework that is robust against uncertainties resulting from the need for learning based on a relatively small training sample set. To this end, we incorporate model based prior knowledge about the cell-load in the learning framework. For example, an inherent property of the cell-load is that it is monotonic in downlink (data) rates. To obtain additional prior knowledge, we first study the feasible rate region, i.e., the set of all vectors of user rates that can be supported by the network. We prove that the feasible rate region is compact. Moreover, we show the existence of a Lipschitz function that maps feasible rate vectors to cell-load vectors. With this prior knowledge in hand, we propose a learning framework which has better robustness and accuracy than standard multivariate learning techniques, especially for small training sample sets.

The work presented in Chapter 2 has been partially presented in the conference paper [ACS18] and in its entirety in the journal paper [ACS19].

In **Chapter 3** we tackle the problem of robust multiuser detection that has been a subject of recent research in the context of non-orthogonal multiple access systems (NOMA). It is known that the optimal multiuser detector (in terms of the bit error rate of the desired user) is the maximum a posteriori (MAP) filter. Due to the impracticality of the optimal MAP filter [it requires complete knowledge of user channels and noise power], various suboptimal linear and nonlinear receivers have been proposed in the literature. Nonlinear receivers outperform linear receivers due to their higher resolution, but they lack the robustness of linear receivers in the presence of sporadic interference. We develop an online learning based partially linear (or partially nonlinear) receiver consisting of a linear and a nonlinear component, where the amount of nonlinearity can be adapted based on the performance of the desired user. Since we work with relatively small training sample sets, we study the multiuser detection model to incorporate prior knowledge. In particular, we show that the optimal MAP filter (which is nonlinear) belongs to a certain reproducing kernel Hilbert space (RKHS), and therefore we design the nonlinear component mentioned above in this RKHS. We then extend this RKHS by adding linear functions to it which gives rise to a sum RKHS of partially linear functions. We propose an online learning framework that has low complexity and it does not require any intermediate parameter estimation [e.g., user channels] which is a limitation of most of the previous studies.

The work presented in Chapter 3 has been partially published in the conference paper [ALCYS18] and the book chapter [DCYS19]. The previously unpublished parts in this chapter are under preparation for submission to a journal [ACS21].

In **Chapter 4** we present a novel method to learn probability density functions (pdfs). Our approximation method is based on projection onto convex sets (POCS) technique, which means that our method is numerically robust. The convex sets are based on prior knowledge about pdfs and information extracted from a relatively small sample set. Our proposed method can work with relatively small number of training sample sets and therefore it is suited to real-time applications in dynamic wireless networks. To show the efficacy of our method, we apply our framework to distributed multiuser detection in a cloud-radio access network (CRAN).

The work presented in Chapter 4 has been partially published in the conference paper [ACUS18].

Further Work: The following publications are not part of this thesis.

- The conference publication [ACS16] addresses the problem of minimizing the energy consumption in large-scale ultra-dense networks (UDNs) by means of a joint optimization of RAN and multi-hop wireless backhaul network. The objective is to operate the network with the smallest set of base stations while meeting the quality of service (QoS) requirements of users. We first pose the optimization problem as a convex optimization problem. We use a Lagrangian decomposition method to separate the problem in smaller subproblems, which are then solved using minimax primal-dual optimization in a distributed manner. By using the proposed method both the primal and dual problems can be solved at each base station with minimal information exchange between the neighboring base stations.

- The multiuser detection scheme from Chapter 3 is demonstrated in [MACKS20] in a joint work with Matthias Mehlhose. In a *hardware-in-a-loop* system, we perform comparisons with minimum mean square error (MMSE) and SIC receivers in terms of symbol error rate (SER) and complexity.

- In the coauthored study with Qi Liao [LAS16] we develop an optimization framework for self-organizing networks (SON). The objective is to ensure efficient network operation by a joint optimization of different SON functionalities, which includes capacity, coverage and load balancing. Based on the axiomatic framework of monotone and strictly subhomogeneous functions, we formulate an optimization problem for the uplink and propose a two-step optimization scheme using fixed point iterations: i) per base station antenna tilt optimization and power allocation, and ii) cluster-based base station assignment of users and power allocation. We then consider the downlink, which is more difficult to handle due to coupled variables, and show downlink-uplink duality relationship. As a result, a solution for the downlink is obtained by solving the uplink problem. Simulations show that our approach achieves a good trade-off between coverage, capacity, and load balancing.

Copyright Information

Parts of this thesis have already been published as journal articles, book chapters, or conference proceedings as listed in the publication list in the Appendix. These parts are covered by the copyrights of the respective publications.

1.3 Notation and Abbreviations

We use the following notation and abbreviations:

Notation	
$\mathbb{R}, \mathbb{R}_{\geq 0}, \mathbb{R}_{>0}$	the set of reals, non-negative, and posotive reals, respectively
$\mathbb{Z}_{\geq 0}$	the set of non-negative integers
$\mathbb{N}$	the set of natural numbers (excluding zero)
$\\|\cdot\\|$	Euclidean norm in $\mathbb{R}^m$ or $\mathbb{C}^m$
$\\|\cdot\\|_\infty$	l_∞ norm in $\mathbb{R}^m$
$\overline{N_1, N_2}$	$\{N_1, N_1+1, N_1+2, \ldots, N_2\} \subset \mathbb{N}$
$x \in \mathcal{X}$	x is an element of $\mathcal{X}$
$\mathcal{Y} \subset \mathcal{X}$	$\mathcal{Y}$ is a subset of $\mathcal{X}$
$(x_n)_{n\in\mathbb{N}} \subset \mathcal{X}$	a sequence of elements of $\mathcal{X}$
$\mathcal{B}_{\mathcal{X}}(\mathbf{x}_o, \delta)$	the open-ball of radius $\delta > 0$ centered at $\mathbf{x}_o \in \mathcal{X}$
$\mathcal{N}(\mu, \sigma)$	Gaussian (normal) distribution with mean μ and variance σ
$\mathbf{P}_{\mathcal{C}}(\mathrm{x})$	projection of x onto C
$\mathbb{P}[\mathcal{A}]$	probability of an event $\mathcal{A}$
$\mathrm{card}(\mathcal{X})$	cardinality of $\mathcal{X}$
$\mathrm{span}\ \mathcal{X}$	the linear span of $\mathcal{X}$
$\mathcal{X} \times \mathcal{Y}$	Cartesian product of the sets $\mathcal{X}$ and $\mathcal{Y}$
$\mathcal{X}^n$	nth Cartesian product of $\mathcal{X}$
$\mathbb{E}[\mathbf{X}]$	the expected value of the random variable $\mathbf{X}$
$\min \mathcal{X}$	the smallest element of $\mathcal{X}$
$\max \mathcal{X}$	the largest element of $\mathcal{X}$
$\mathbf{0}$	the all-zero vector in $\mathbb{R}^m$
$(\mathbf{x})_+$	coordinate-wise $\max\{\mathbf{x}, \mathbf{0}\}$, $\mathbf{x} \in \mathbb{R}^m$
$\mathbf{x} \leq \mathbf{y}$	coordinate-wise inequality
$\inf \mathcal{X}$	the infimum of $\mathcal{X}$
$\sup \mathcal{X}$	the supremum of $\mathcal{X}$
$\log(\cdot)$	The base 2 logarithm $\log_2(\cdot)$
$[\mathbf{x}]_i$ or x_i	the i-th entry of a vector $\mathbf{x}$
$[\mathbf{G}]_{i,j}$	the entry in the ith row and jth column of matrix $\mathbf{G}$
$]a, b[$, $[a, b]$	an open and a closed interval, respectively
$]a, b]$, $[a, b[$	half-open intervals, which do not contain a or b, respectively
$:=$	equal by definition

Abbreviations

APSM	adaptive projected subgradient method
AWGN	additive white gaussian noise
BER	bit error rate
BPSK	binary phase-shift keying
LIMF	Lipschitz monotone functions
LTE	long term evolution
MAP	maximum a posteriori
MMSE	minimum mean-squared error
OFDMA	orthogonal frequency division multiple access
POCS	projection onto convex sets
QPSK	quadrature phase-shift keying
RAN	radio access network
RKHS	reproducing kernel Hilbert space
RRM	radio resource management
SNR	signal-to-noise-ratio
SINR	signal-to-interference-plus-noise-ratio
SIC	successive interference cancellation
QP	quadratic program
LP	linear program

1.4 Some Results from Real Analysis

Let $\mathcal{S}$ be a normed vector space equipped with a norm $\|\cdot\|_{\mathcal{S}}$ and its induced metric $d_{\mathcal{S}} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0} : (\mathbf{s}_o, \mathbf{s}) \mapsto \|\mathbf{s}_o - \mathbf{s}\|_{\mathcal{S}}$. We denote by $\mathcal{B}_{\mathcal{S}}(\mathbf{s}_o, \delta) := \{\mathbf{s} \in \mathcal{S} | \|\mathbf{s} - \mathbf{s}_o\|_{\mathcal{S}} < \delta\}$ the open-ball of radius $\delta > 0$ centered at $\mathbf{s}_o \in \mathcal{S}$. A sequence $(\mathbf{s}_n)_{n\in\mathbb{N}} \subset \mathcal{S}$ is said to converge (in norm) to $\mathbf{s} \in \mathcal{S}$ if $\|\mathbf{s}_n - \mathbf{s}\|_{\mathcal{S}} \to 0$ [Lue97, Page 26]. We now define the concepts of *boundedness*, *closedness*, and *compactness*.

Definition 1.1 (*Boundedness, Closedness, and Compactness*)**.** [Lue97, Chapter 2] Consider a set $\mathcal{K}$ in the normed space $(\mathcal{S}, \|\cdot\|_{\mathcal{S}})$.

a). Boundedness: $\mathcal{K}$ is bounded if

$$(\exists L \geq 0)\ (\forall \mathbf{k} \in \mathcal{K})\ \|\mathbf{k}\|_{\mathcal{S}} \leq L.$$

b). Closedness: $\mathcal{K}$ is closed if and only if every convergent sequence $(\mathbf{k}_n)_{n\in\mathbb{N}} \subset \mathcal{K}$ has a limit in $\mathcal{K}$.

c). Compactness: $\mathcal{K}$ is compact if every sequence $(\mathbf{k}_n)_{n\in\mathbb{N}} \subset \mathcal{K}$ has a convergent subsequence with a limit in $\mathcal{K}$.

In this thesis we shall consider the space $C(\mathcal{X}, \mathcal{Y})$ of vector-valued continuous functions mapping $\mathcal{X} \subset \mathbb{R}^N_{>0}$ to $\mathcal{Y} \subset \mathbb{R}^M_{\geq 0}$. For a function $\mathbf{g} \in C(\mathcal{X}, \mathcal{Y})$ its ith component ($i \in \overline{1, M}$) $g_i : \mathcal{X} \to \mathbb{R}_{\geq 0}$ is a scalar continuous function. We equip $C(\mathcal{X}, \mathcal{Y})$ with the uniform norm [Lue97, Page 23]

$$\|\mathbf{g}\|_{C(\mathcal{X})} = \sup_{\mathbf{x}\in\mathcal{X}} \max_{1\leq i\leq M} g_i(\mathbf{x}). \tag{1.1}$$

If $\mathcal{X}$ is compact, then the supremum is attained according to the *extreme value theorem* [Mun00] because the max operation[1] preserves continuity.

Definition 1.2 (*Monotonic Function*)**.** Let $\mathcal{X} \subset \mathbb{R}^N_{>0}$ and $\mathcal{Y} \subset \mathbb{R}^M_{\geq 0}$. A function $\mathbf{f} : \mathcal{X} \to \mathcal{Y}$ is said to be monotonic if

$$(\forall \mathbf{x} \in \mathcal{X})\ (\forall \mathbf{y} \in \mathcal{X})\ \mathbf{x} \leq \mathbf{y} \Rightarrow \mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\mathbf{y}).$$

Definition 1.3 (**L**-*Lipschitz function*)**.** Consider $\mathbf{f} \subset C(\mathcal{X}, \mathcal{Y})$ and a vector $\mathbf{L} := [L_1, L_2, \cdots, L_M]^\intercal \in \mathbb{R}^M_{\geq 0}$. We say that $\mathbf{f}$ is $\mathbf{L}$-Lipschitz on $\mathcal{X}$ if

$$(\forall i \in \overline{1, M})\ (\forall \mathbf{x} \in \mathcal{X})\ (\forall \mathbf{y} \in \mathcal{X})\ |f_i(\mathbf{x}) - f_i(\mathbf{y})| \leq L_i \|\mathbf{x} - \mathbf{y}\|.$$

Definition 1.4 (**L**-*Lipschitz-Monotonic Function*)**.** We say that $\mathbf{f} \subset C(\mathcal{X}, \mathcal{Y})$ belongs to the class of **L**-*Lipschitz-Monotonic Functions* (LIMF) if $\mathbf{f}$ is monotonic and there exists $\mathbf{L} \in \mathbb{R}^M_{\geq 0}$ such that $\mathbf{f}$ is $\mathbf{L}$-Lipschitz.

Note that a function $\mathbf{f} \in C(\mathcal{X}, \mathcal{Y})$ is continuous at $\mathbf{x}_o \in \mathcal{X}$ if given $\epsilon > 0$, there exists $\delta_{\mathbf{x}_o} > 0$ such that $(\forall \mathbf{x} \in \mathcal{B}_{\mathcal{X}}(\mathbf{x}_o, \delta_{\mathbf{x}_o}))$ $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}_o)\| < \epsilon$. The following concept of *equicontinuity* extends the concept of continuity to a collection/set $\mathcal{F} \subset \mathcal{C}(\mathcal{X}, \mathcal{Y})$ of functions.

Definition 1.5 (*Equicontinuity of a Set*)**.** [Mun00, Chapter 7] A function set $\mathcal{F} \subset \mathcal{C}(\mathcal{X}, \mathcal{Y})$ is called equicontinuous at $\mathbf{x}_o \in \mathcal{X}$ if for every $\epsilon > 0$ there exists $\delta_{\mathbf{x}_o} > 0$ such that

$$(\forall \mathbf{x} \in \mathcal{B}_{\mathcal{X}}(\mathbf{x}_o, \delta_{\mathbf{x}_o}))\ (\forall \mathbf{f} \in \mathcal{F})\ \|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}_o)\| < \epsilon.$$

[1] The usage of max in (1.1) is different to the component-wise max in $\max\{\mathbf{x}, \mathbf{0}\}$. The distinction between the two usages shall be clear by the context in which they are used.

Furthermore, if for every $\epsilon > 0$ there exists $\delta > 0$ such that $(\forall \mathbf{x}_o \in \mathcal{X})$ $(\forall \mathbf{x} \in \mathcal{B}_{\mathcal{X}}(\mathbf{x}_o, \delta))$ $(\forall \mathbf{f} \in \mathcal{F})$ $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}_o)\| < \epsilon$, then $\mathcal{F}$ is said to be (uniformly) equicontinuous.

The general concept of compactness in normed vector spaces has been introduced in Definition 1.1. The following Fact, along with Remark 1.1, characterizes compact subsets of $C(\mathcal{X}, \mathcal{Y})$.

Fact 1.1 (Compact subsets of $C(\mathcal{X}, \mathcal{Y})$)**.** *[Bro14] [Mun00, Corollary 45.5] Let $\mathcal{X}$ be compact. Then,*

a). Arzelá-Ascoli's Theorem: Every bounded and equicontinuous sequence $(\mathbf{f}_n)_{n\in\mathbb{N}} \subset C(\mathcal{X}, \mathcal{Y})$ has a convergent subsequence.

b). A set $\mathcal{F} \subset C(\mathcal{X}, \mathcal{Y})$ is compact if it is bounded, equicontinuous, and closed.

Remark 1.1 (Compactness in $\mathbb{R}^m$ and in $C(\mathcal{X}, \mathcal{Y})$). A subset of a finite dimensional Euclidean space is compact *if and only* if it is bounded and closed (see *Heine-Borel Theorem* [Mun00, Theorem 27.3]). However, in $C(\mathcal{X}, \mathcal{Y})$, equicontinuity is required in addition to boundedness and closedness for compactness.

Now, we present the concept of *implicit functions.*

Fact 1.2 (*Implicit function theorem*)**.** *[KP03] Consider sets $\mathcal{X} \subset \mathbb{R}^N$, $\mathcal{Y} \subset \mathbb{R}^M$, and $\mathcal{Z} \subset \mathbb{R}^M$, and a vector-valued continuous function $\mathbf{g} : \mathcal{Y} \times \mathcal{X} \to \mathcal{Z}$. Denote by $(i \in \overline{1, M})$ $g_i : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}$ the ith component of $\mathbf{g}$. Now, assume that $\mathbf{g}$ is continuously differentiable in a neighborhood $(\exists \delta_{\overline{x}}, \delta_{\overline{y}} > 0)$ $\mathcal{B}_{\mathcal{Y}}(\overline{\mathbf{y}}, \delta_{\overline{y}}) \times \mathcal{B}_{\mathcal{X}}(\overline{\mathbf{x}}, \delta_{\overline{x}})$ of a point $(\overline{\mathbf{y}}, \overline{\mathbf{x}}) \in \mathcal{Y} \times \mathcal{X}$, and that $\mathbf{g}(\overline{\mathbf{y}}, \overline{\mathbf{x}}) = \mathbf{0}$. Let the Jacobian of $\mathbf{g}$ with respect to variables $\mathbf{y}$ (i.e., the first argument), denoted by $\boldsymbol{\nabla}^{\mathbf{g}}_{\mathbf{y}} : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}^{M\times M}$ and defined as*

$$\boldsymbol{\nabla}^{\mathbf{g}}_{\mathbf{y}} := \begin{pmatrix} \frac{\partial g_1}{\partial y_1} & \frac{\partial g_1}{\partial y_2} & \cdots & \frac{\partial g_1}{\partial y_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_M}{\partial y_1} & \frac{\partial g_M}{\partial y_2} & \cdots & \frac{\partial g_M}{\partial y_M} \end{pmatrix},$$

be invertible at $(\overline{\mathbf{y}}, \overline{\mathbf{x}})$. Then, there exists a (unique and continuous) "implicit" function $\mathbf{f} : \mathcal{B}_{\mathcal{X}}(\overline{\mathbf{x}}, \delta_{\overline{x}}) \to \mathcal{B}_{\mathcal{Y}}(\overline{\mathbf{y}}, \delta_{\overline{y}})$ such that $(\forall \mathbf{x} \in \mathcal{B}_{\mathcal{X}}(\overline{\mathbf{x}}, \delta_{\overline{x}}))$ $\mathbf{g}(\mathbf{f}(\mathbf{x}), \mathbf{x}) = \mathbf{0}$. Furthermore, $\mathbf{f}$ is continuously differentiable on $\mathcal{B}_{\mathcal{X}}(\overline{\mathbf{x}}, \delta_{\overline{x}})$. The value of the Jacobian of $\mathbf{f}$ is given by

$$(\forall \mathbf{x} \in \mathcal{B}_{\mathcal{X}}(\overline{\mathbf{x}}, \delta_{\overline{x}}))\ \boldsymbol{\nabla}^{\mathbf{f}}_{\mathbf{x}}(\mathbf{x}) = -\big(\boldsymbol{\nabla}^{\mathbf{g}}_{\mathbf{y}}(\mathbf{f}(\mathbf{x}), \mathbf{x})\big)^{-1} \boldsymbol{\nabla}^{\mathbf{g}}_{\mathbf{x}}(\mathbf{f}(\mathbf{x}), \mathbf{x}),$$

where $\nabla_{\mathbf{x}}^{\mathbf{g}} : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}^{M \times N}$ *is the Jacobian of* $\mathbf{g}$ *with respect to variables* $\mathbf{x}$ *(i.e., the second argument) given by*

$$\nabla_{\mathbf{x}}^{\mathbf{g}} := \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_M}{\partial x_1} & \frac{\partial g_M}{\partial x_2} & \cdots & \frac{\partial g_M}{\partial x_N} \end{pmatrix}.$$

1.5 Reproducing Kernel Hilbert Spaces

Reproducing kernel Hilbert spaces (RKHS) have been extensively used in diverse fields such as statistics, probability, signal processing, and machine learning, among others [BTA04, TSY11, STY09, Yuk15a].[2] There exists a large introductory as well as detailed literature on RKHSs; here we only cover aspects that are relevant to this thesis.

Definition 1.6 (Reproducing kernel Hilbert spaces and reproducing kernels (RKHS))**.** Let $\mathcal{U}$ be an arbitrary nonempty set. A Hilbert space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ of real-valued functions $f : \mathcal{U} \to \mathbb{R}$ is called a *reproducing kernel Hilbert space* if and only if there exists a positive symmetric function $\kappa : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ such that:

1. $(\forall x \in \mathcal{U})\ \kappa(\cdot, \boldsymbol{x}) \in \mathcal{H}$; and
2. $(\forall x \in \mathcal{U})(\forall f \in \mathcal{H})\ f(\boldsymbol{x}) = \langle f, \kappa(\cdot, \boldsymbol{x})\rangle_{\mathcal{H}}$ (reproducing property).

The function $\kappa : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ is known as the *reproducing kernel* of $\mathcal{H}$ and it is unique. Strictly speaking, an RKHS should be denoted by the triplet $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}}, \mathcal{U})$, where $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ is given by the kernel of $\mathcal{H}$, unless there is no ambiguity about the domain $\mathcal{U}$. When there is also no ambiguity about the kernel, then we can use simply $\mathcal{H}$.

Remark 1.2 (RKHS Coordinate System)*.* Note that $\mathcal{H}$ in Definition 1.6 is a linear/vector space. Thinking of an arbitrary member $f \in \mathcal{H}$ as an infinite-length vector, property (2) shows that, even if an explicit orthonormal basis for $\mathcal{H}$ may be unknown, the reproducing kernel of $\mathcal{H}$ introduces a special coordinate system which can be used to obtain the coordinate $f(x)$ for every $x \in \mathcal{U}$.

Signal processing in an RKHS entails many operations involving the reproducing kernel. Therefore, we now define these special functions and show how to construct RKHSs using these functions. Since reproducing kernels are the same as *positive definite kernels*, and it follows that every positive definite kernel is associated with a unique RKHS [*Moore-Aronszajn theorem* [Aro50] [BTA04, Ch. 1]], we now formally define positive definite kernels.

[2] In this thesis we deal with real Hilbert spaces of real-valued functions, but Definition 1.6 can be naturally extended to complex Hilbert spaces [BTA04].

Definition 1.7 (Positive Definite Kernel)**.** Let $\mathcal{U}$ be an arbitrary set. We say that $\kappa : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ is a (real) positive definite kernel if the following properties hold:

a). Symmetry: $(\forall \boldsymbol{x} \in \mathcal{U})(\forall \boldsymbol{y} \in \mathcal{U})\ \kappa(\boldsymbol{x}, \boldsymbol{y}) = \kappa(\boldsymbol{y}, \boldsymbol{x})$

b). Non-negativity: $(\forall M \in \mathbb{N})(\forall (\alpha_1, \ldots, \alpha_M) \in \mathbb{R}^M)(\forall (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M) \in \mathcal{U}^M)$

$$\sum_{k=1}^{M} \sum_{j=1}^{M} \alpha_k \alpha_j \kappa(\boldsymbol{x}_k, \boldsymbol{x}_j) \geq 0.$$

An equivalent way of defining kernels is to require $\kappa : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ to satisfy two properties. First, κ must be symmetric. Second, for arbitrary $M \in \mathbb{N}$ and $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M) \in \mathcal{U}^M$, the matrix $\mathbf{K} \in \mathbb{R}^{M \times M}$ with the element in the ith row and jth column given by $[\mathbf{K}]_{i,j} := \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ has to be positive semi-definite.[3].

In this thesis we take $\mathcal{U}$ as a subset of $\mathbb{R}^m$ or $\mathbb{C}^m$ and we deal with only two reproducing kernels: the linear kernel $\kappa_{\mathrm{L}} : \mathcal{U} \times \mathcal{U} \to \mathbb{R} : (\mathbf{u}, \mathbf{v}) \mapsto \Re(\mathbf{u}^{\mathsf{H}} \mathbf{v})$ and the Gaussian kernel $\kappa_{\mathrm{G}} : \mathcal{U} \times \mathcal{U} \to \mathbb{R} : (\mathbf{u}, \mathbf{v}) \mapsto \exp\left(-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2}\right)$ with width $\sigma > 0$.

For a given kernel κ, we can construct a real vector space $\mathcal{H}_0$ with functions of the following type:

$$f : \mathcal{U} \to \mathbb{R} \in \operatorname{span}\{\kappa(\cdot, \boldsymbol{x}) \ : \ \boldsymbol{x} \in \mathcal{U}\} =: \mathcal{H}_0; \tag{1.2}$$

the sum and real scalar multiplication in $\mathcal{H}_0$ are defined in the usual way. To endow $\mathcal{H}_0$ with the structure of an inner-product space, let $(M, N) \in \mathbb{N} \times \mathbb{N}$, $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M) \in \mathcal{U}^M$, $(\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_N) \in \mathcal{U}^N$, $(\alpha_1, \ldots, \alpha_M) \in \mathbb{R}^M$, and $(\beta_1, \ldots, \beta_N) \in \mathbb{R}^N$ be arbitrary. Now, consider the functions $f : \mathcal{U} \to \mathbb{R} : \boldsymbol{x} \mapsto \sum_{k=1}^{M} \alpha_k\ \kappa(\boldsymbol{x}, \boldsymbol{x}_k)$ and $g : \mathcal{U} \to \mathbb{R} : \boldsymbol{x} \mapsto \sum_{j=1}^{N} \beta_j\ \kappa(\boldsymbol{x}, \boldsymbol{x}'_j)$. Clearly, both $f = \sum_{k=1}^{M} \alpha_k\ \kappa(\cdot, \boldsymbol{x}_k)$ and $g = \sum_{j=1}^{M} \beta_j\ \kappa(\cdot, \boldsymbol{x}'_j)$ can be seen as arbitrary members of $\mathcal{H}_0$, and we can consider the inner-product

$$\langle f, g \rangle_{\mathcal{H}_0} := \sum_{k=1}^{M} \sum_{j=1}^{N} \alpha_k \beta_j \kappa(\boldsymbol{x}_k, \boldsymbol{x}'_j),$$

which induces the norm $\|\cdot\|_{\mathcal{H}_0} = \langle \cdot, \cdot \rangle^{1/2}$. Note that $(\mathcal{H}_0, \langle \cdot, \cdot \rangle_{\mathcal{H}_0})$ is a pre-Hilbert space that is not necessarily complete (i.e., Cauchy sequences in $\mathcal{H}_0$ may not have a limit in $\mathcal{H}_0$). However, $(\mathcal{H}_0, \langle \cdot, \cdot \rangle_{\mathcal{H}_0})$ can be completed[4] to become an RKHS $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}} = \langle \cdot, \cdot \rangle_{\mathcal{H}_0})$ whose

[3] Some authors further divide kernels into positive definite kernels and positive semi-definite kernels. However, here we do not make this distinction.

[4] Let $\mathbb{R}^{\mathcal{U}}$ be the space of all functions $f : \mathcal{U} \to \mathbb{R}$. Roughly speaking, completion means that we add to $\mathcal{H}_0 \subset \mathbb{R}^{\mathcal{U}}$ all the functions in $\mathbb{R}^{\mathcal{U}}$ that are infinitesimally close to $\mathcal{H}_0$.

members include functions of the type in (1.2). Fact 1.3 shows the remarkable property of RKHSs that convergence in the RKHS-norm implies point-wise convergence.

Fact 1.3 (Uniform Approximation in an RKHS)**.** *[BTA04] Let $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ be an RKHS defined over a set $\mathcal{U}$. Then for any sequence $(f_n)_{n\in\mathbb{N}} \subset \mathcal{H}$ and a function $f^* \in \mathcal{H}$, if $\|f_n - f^*\|_{\mathcal{H}} \to 0$ as $n \to \infty$, then $f_n(x) = f^*(x)$ as $n \to \infty$ for every $x \in \mathcal{U}$.*

1.5.1 Sum Spaces of Reproducing Kernel Hilbert Spaces

We now briefly summarize the ideas originally proposed in [Yuk15a]. Suppose that we have the task of learning an unknown nonlinear function $f : \mathcal{U} \to \mathbb{R}$ (where $\mathcal{U} \subset \mathbb{R}^N$) that can be decomposed into Q distinct components, such as high and low frequency components, linear and nonlinear components, and periodic and aperiodic components. If each of these components can be well approximated by members of one of the Q RKHSs $(\mathcal{H}_1, \langle\cdot,\cdot\rangle_{\mathcal{H}_1}), \ldots, (\mathcal{H}_Q, \langle\cdot,\cdot\rangle_{\mathcal{H}_Q})$, we can naturally assume that the unknown function f is a member of the *sum space*

$$\mathcal{H} := \left\{ \sum_{q\in\mathcal{Q}} f_q \ : \ (\forall q \in \mathcal{Q}) \ f_q \in \mathcal{H}_q \right\},$$

where $\mathcal{Q} = \overline{1,Q}$. For fixed (strictly) positive weights $[w_1, \ldots, w_Q] =: \boldsymbol{w}$, we can equip the space $\mathcal{H}^+$ with the (weighted) norm

$$(\forall f \in \mathcal{H}) \ \|f\|^2_{\mathcal{H},\boldsymbol{w}} := \min \left\{ \sum_{q\in\mathcal{Q}} w_q^{-1} \|f_q\|^2_{\mathcal{H}_q} \ : \ f = \sum_{q\in\mathcal{Q}} f_q, \ (\forall q \in \mathcal{Q}) \ f_q \in \mathcal{H}_q \right\}, \quad (1.3)$$

and it can be shown that the resulting normed space is an RKHS $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H},\boldsymbol{w}})$ associated with the reproducing kernel $\kappa := \sum_{q\in\mathcal{Q}} w_q \ \kappa_q$ [BTA04, Page 24] [Aro50, Yuk15a].

In applications, without imposing any additional structure on $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H},\boldsymbol{w}})$, we note that the decomposition $f = \sum_{q\in\mathcal{Q}} f_q$ above is not necessarily unique, which in turn makes the computation of the norm in (1.3) challenging. One notable exception for the non-uniqueness of the decomposition is the case where the sum space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H},\boldsymbol{w}})$ is constructed with RKHSs $(\mathcal{H}_1, \langle\cdot,\cdot\rangle_{\mathcal{H}_1}), ..., (\mathcal{H}_Q, \langle\cdot,\cdot\rangle_{\mathcal{H}_Q})$ satisfying $\mathcal{H}_j \cap \mathcal{H}_q = \{0\}$ if $j \neq q$. In this case, we have

$$(\forall f \in \mathcal{H}) \ \|f\|^2_{\mathcal{H},\boldsymbol{w}} = \sum_{q\in\mathcal{Q}} w_q^{-1} \ \|f_q\|^2_{\mathcal{H}_q} \quad (1.4)$$

and

$$(\forall f \in \mathcal{H})(\forall g \in \mathcal{H})\, \langle f, g \rangle_{\mathcal{H},\boldsymbol{w}} = \sum_{q \in \mathcal{Q}} w_q^{-1} \;\; \langle f_q, g_q \rangle_{\mathcal{H}_q} . \tag{1.5}$$

From a practical perspective, with these sum spaces, algorithms can perform many operations by simply considering the Hilbert spaces $(\mathcal{H}_1, \langle \cdot, \cdot \rangle_{\mathcal{H}_1})$, ..., $(\mathcal{H}_Q, \langle \cdot, \cdot \rangle_{\mathcal{H}_Q})$ independently and by summing the results. By doing so, hard-to-solve optimization problems, such as those required for the evaluation of norms in (1.3) are avoided.

1.6 Projections onto Closed-Convex Sets

Let $\mathcal{H}$ be a real Hilbert space with an inner-product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and an induced norm $\|f\|^2_{\mathcal{H}} = \langle f, f \rangle$. For every $x \in \mathcal{H}$, the projection $\mathbf{P}_{\mathcal{C}}(x) : \mathcal{H} \to \mathcal{C}$ onto a nonempty closed convex set $\mathcal{C} \subset \mathcal{H}$ is the solution to the problem:

$$\inf_{y \in \mathcal{C}} \|x - y\|_{\mathcal{H}} .$$

which implies that if $y^* \in \mathcal{C}$ is a solution to the above problem, then we have

$$(\forall y \in \mathcal{C}) \;\; \|x - y^*\|_{\mathcal{H}} \leq \|x - y\|_{\mathcal{H}} ,$$

In other words, y^* is the *best approximation* of $x \in \mathcal{H}$ from $\mathcal{C}$. Moreover, $\mathbf{P}_{\mathcal{C}}(x)$ always exists and it is unique [Lue97, Theorem. 1, Chapter. 3.12]. Note that the above projection/best approximation is defined only for Hilbert spaces and it is not valid for general normed spaces.

Next, we show projections onto some simple closed-convex sets used in this thesis.

1.6.1 Projection onto a Hyperplane

Let $0 \neq a \in \mathcal{H}$ and $c \in \mathbb{R}$. A hyperplane $\mathcal{P} \subset \mathcal{H}$ is the closed-convex set

$$\mathcal{P} := \{ h \in \mathcal{H} \;:\; \langle h, a \rangle_{\mathcal{H}} = c \} ,$$

and for any function $f \in \mathcal{H}$, its projection onto $\mathcal{P}$, denoted by $\mathbf{P}_{\mathcal{P}}(f) : \mathcal{H} \to \mathcal{P}$, is given as [TSY11]

$$\mathbf{P}_{\mathcal{P}}(f) = f - \frac{\langle f, a \rangle_{\mathcal{H}} - c}{\|a\|^2} \; a.$$

1.6.2 Projection onto a Hyperslab

A hyperslab can be thought of as the space between two hyperplanes. Let $0 \neq a \in \mathcal{H}$ and $(b, c) \in \mathbb{R}^2$. A hyperslab $\mathcal{S} \subset \mathcal{H}$ is the closed-convex set

$$\mathcal{S} := \{h \in \mathcal{H} \ : \ b \leq \langle h, a \rangle_{\mathcal{H}} \leq c\},$$

and for any function $f \in \mathcal{H}$, its projection onto $\mathcal{S}$, denoted by $\mathbf{P}_{\mathcal{S}}(f) : \mathcal{H} \to \mathcal{S}$, is given as [TSY11]

$$\mathbf{P}_{\mathcal{S}}(f) := \begin{cases} f - \frac{\langle f, a \rangle_{\mathcal{H}} - c}{\|a\|^2} a, & \text{if } \langle f, a \rangle_{\mathcal{H}} > c, \\ f - \frac{\langle f, a \rangle_{\mathcal{H}} - b}{\|a\|^2} a, & \text{if } \langle f, a \rangle_{\mathcal{H}} < b, \\ f, & \text{otherwise.} \end{cases}$$

1.7 Bounded Error Estimation

Consider a data set $\mathcal{D} = \{(y_i, x_i) \in \mathbb{R}^2\}_{i=1}^n$. Suppose there exists a function $f^\star : \mathbb{R} \to \mathbb{R}$ such that $y_i = f^\star(x_i) + n_i$ are seen as n noisy observations of $f^\star$, where $n_i \in \mathbb{R}$ models noise. Function approximation entails obtaining an approximation $g^\star$ of $f^\star$ using $\mathcal{D}$, such that $g^\star(x)$ estimates $f^\star(x)$ optimally (in some sense).

In function approximation literature, there are various "philosophies" regarding what constitutes an optimal approximation and how to go about obtaining one. Two well-known types of function approximation are: *stochastic/statistical estimation* and *bounded error estimation.* Statistical estimation assumes that y_i and x_i are generated by an underlying joint probability distribution and the noise sequence $(n_i)_{i \in \overline{1,N}}$ is generated by a known probability distribution, e.g., a white Gaussian noise distribution. One then considers a *regression model* $y = f^\star(x) + n$, where y, x, and n are now random. In this case, a popular notion of optimality is that an optimal $g^\star$ minimizes the mean squared-error (MSE) between $g^\star(x)$ and y. Under certain assumptions on the involved probability distributions, the optimal approximation is given by $g^\star : x \mapsto \mathbb{E}[Y|X = x]$, where the expectation is taken over the conditional distribution $p(y|x)$. Other popular techniques include maximum likelihood (ML), maximum a posteriori (MAP), and general Bayesian inference. In general, statistical estimation tends to target the average performance of the estimator. However, in general, the involved probability distributions are unknown and their estimation is complex, requiring large data sets.

In *bounded error estimation* [Wit68, Sch68, NG95] (also referred to as *set-membership estimation* [MV91, Cas02] and *robust estimation* [MT85]), noise is assumed to be unknown but bounded in a given norm. Note that this includes the case when the observations are noiseless provided that no probabilities are attached to y_i and x_i. In contrast to the stochastic estimation theory, bounded error estimation is deterministic in the sense that y, x, and n above are deterministic and $g^\star(x)$ gives a direct estimate $y^\star$ of y. Of course the approximation in this case is heavily dependent on the data set $\mathcal{D}$ since the prior statistical information regarding the variables is missing. Therefore, one begins by adding deterministic prior knowledge to the framework. First, we assume that $f^\star \in \mathcal{H}$, where $\mathcal{H}$ is as yet arbitrary and it is referred to as the *problem element set* or *Hypothesis space*, i.e., all members of $\mathcal{H}$ are possible candidates to be $g^\star$. Then we work with two pieces of information:

1. Prior knowledge about $f^\star$ that helps us identify $\mathcal{H}$. In some cases we can restrict $f^\star$ to smaller subsets of $\mathcal{H}$ as shown in Figure 1.1.

2. A sample set $\mathcal{D}$ and an (in some cases) an assumption on the worst-case noise bound that helps us restrict $f^\star$ further to a sufficiently small $\mathcal{F} \subset \mathcal{H}$. The set $\mathcal{F}$ is referred to as the *feasible solution set.*

There are various optimality criteria and algorithms in bounded error estimation literature based on how one obtains the approximation $g^\star$ from the set $\mathcal{F}$. Since $\mathcal{F}$ contains all feasible solutions to the approximation problem, any element $g \in \mathcal{F}$ can be reasonably seen as an approximation of $f^\star$ because $f^\star \in \mathcal{F}$. Among point-wise algorithms, i.e., algorithms that return a single $g^\star \in \mathcal{F}$, the *central algorithms* and the *projection algorithms* are very popular. In general, bounded error estimation seeks to minimize the worst-case approximation error under uncertainty, and in this sense it is more “conservative” as compared to its statistical counterpart. Though this conservatism can be seen as a drawback, in dynamic systems knowledge about underlying probability distributions is hard to come by, and their estimation requires large data sets and complexity.

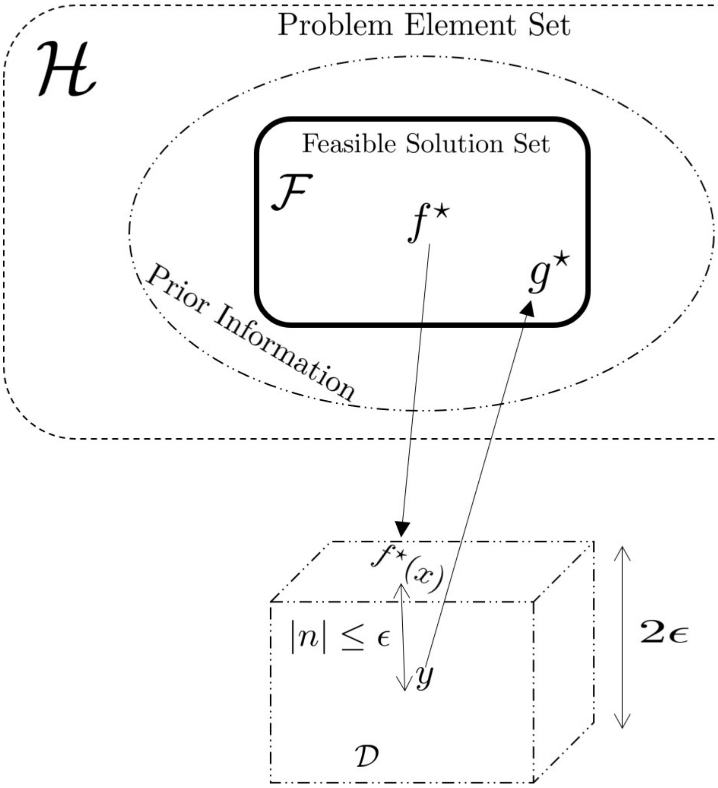


Figure 1.1: Bounded Error Estimation: The problem element set $\mathcal{H}$, the sample set $\mathcal{D}$, and the prior knowledge are used to construct the set $\mathcal{F}$ to which $f^\star$ belongs. An approximation algorithm is then used to obtain $g^* \in \mathcal{F}$.

2 Robust Cell-Load Approximation

2.1 Introduction

5G networks are based on orthogonal frequency-division multiple access (OFDMA). Due to inter-cell interference, radio resource management (RRM) and performance optimization in these networks are challenging. In fact, many RRM problems in OFDMA-based networks, such as small-scale optimal assignment of time-frequency resource blocks and powers to users, have been shown to be NP-hard [WSEA04]. Recent research has therefore focused on the development of frameworks that capture the essence of OFDMA-based networks, while leading to a tractable problem formulation. An example of such a framework is the non-linear load-coupling model proposed in [Sio12, FF12, MK10]. In this framework the *cell-load* at a base station is the fraction of time-frequency resource blocks that are used to support downlink data rates (henceforth simply rates). With this model, and given some power budget that can be used for transmission, one can estimate the cell-load required at each base station to support given rates.

The study in [HYS14] shows the intuitive result that the cell-load is monotonic in rates. The interference coupling between cells implies that increasing the rates in an arbitrary cell increases the cell-load at each base station, which also increases the inter-cell interference.[1] So, it is important for a base station to have a reliable forecast of the cell-load before serving higher rate demands from its associated users. Therefore, cell-load learning can be used to make radio resource management and self-organizing-network (SON) algorithms more reliable and efficient.

Cell-load learning is also a vital part of energy saving mechanisms in radio access networks (RANs). For instance in [SF12], the value of the cell-load is used as an input to a simple heuristic algorithm that switches off base station antennas when the cell-load is low. Large gains in energy savings are reported with minimal effect on the cell sum throughput. The same concept can be used in the case of virtual base station formations in cloud RANs [WTT$^+$16]. In these virtual systems some power-hungry components of a RAN (digital signal processors, line cards, fronthaul, etc.) are virtualized in a central location, and these components can be allocated on-demand to cells according to the cell-load.

[1] For brevity, we assume that cells are not mutually orthogonal.

Cell switch-off (also called carrier switch-off) is a vital energy saving mechanism which relies heavily on robust approximation of the cell-load in various frequency bands in a base station site comprising many cells. Therefore, given RAN data traffic (or rates) predictions, the corresponding cell-load forecasts can enable us to proactively manage network components for energy savings.

2.1.1 The Need for Robust Cell-Load Learning

Note that even though the load-coupling model has been shown to work sufficiently well in predicting the cell-load in some scenarios [FF12, MNK+07, She15], models are only idealizations and in general they do not capture all the intricacies of dynamic wireless environments. Therefore, our objective is to directly learn the underlying function that maps user rates to cell-load values given a training sample set consisting of rate vectors and the corresponding measured cell-load vectors. To improve the learning process, we use the load-coupling model to study some salient aspects of the relationship between rates and the cell-load. We use these aspects as prior knowledge in the learning process.

Compared to the core network, the RAN data traffic is volatile and it shows irregular patterns throughout a day because of the unpredictable nature of user activity and relatively fast changes in the network topology [CPS+13]. Therefore, the underlying statistics (i.e., the joint probability distribution) of rates and the corresponding cell-load values, which are part of the so-called "learning environment", can be assumed to remain constant for only a short time. This implies that a training sample set must be acquired during this short time before the environment changes, since otherwise the sample set can be rendered useless for predicting future cell-load values. However, in general, the smaller the sample set, the larger the uncertainty about the underlying phenomenon, which makes large prediction errors on unseen rates more probable.

In uncertain situations we need robust learning methods that provide a guaranteed worst-case performance under uncertainty. The objective of this study is to develop such a robust learning framework. Our method is optimal in the sense that it minimizes the worst-case or maximum error of approximation which is a classical robust optimization problem [see, e.g., [Suk92,TW80,GW59,Cal14]]. This means that no matter how small the training sample set is, we are guaranteed the best worst-case error. Our method involves only low-complexity and stable mathematical operations and its theoretical properties are very well understood. The above mentioned optimization problem is solved by explicitly incorporating prior knowledge regarding the Lipschitz continuity of the function to be approximated. By incorporating additional prior knowledge concerning monotonicity of the function, we further reduce the worst-case error.

We point out that our framework is different to many modern conventional machine learning frameworks that target mean or average performance rather than the worst-case performance we consider in this study. The performance of many current complex learning methods, such as deep neural networks (DNNs), is often dependent on the availability of a large training (or pre-training) sample set. Including prior knowledge in these frameworks to reduce the reliance on large training sets is not easy, and it is often discouraged [Mar18]. Even if some prior knowledge could be enforced in neural networks (as in [DRG17]), it is theoretically unclear whether (or how) this enables neural networks to learn better. This makes DNNs ill-suited to our setting because we consider learning with very small training sample sets.

2.1.2 Related Work

The load-coupling model [Sio12, FF12, MK10] is commonly used when designing networks according to the long-term evolution (LTE) standard. Recently it has also attracted attention in the context of 5G networks [YYL+18]. More specifically, the load-coupling model has been used in various optimization frameworks dealing with different aspects of network design including data offloading [HYS14], proportional fairness [GECS+16], energy optimization [ACS16, PCS16, RSF14], and load balancing [SY12]. In the context of energy savings, and by using the theory of *implicit functions* [KP03], the study in [RSF14] shows that there exists a continuously differentiable function relating user associations with the base stations to the cell-load. In contrast to [RSF14], the user association is assumed to be fixed in this study; we study the relationship between downlink rates and the cell-load and we incorporate this prior knowledge in our learning framework. Previous studies dealing with cell-load estimation, for instance, in the context of data offloading [HYS14] and maximizing the scaling-up factor of traffic demand [SY15], have used load coupling model driven methods that require information about channel gains, powers, etc.. Most of these methods employ iterative algorithms to estimate the cell-load for given downlink rates and other parameters by exploiting the fact that the cell-load is the fixed point of the *standard interference mapping* [Yat95] that is constructed using the network information. In contrast, we directly learn the underlying function that maps feasible rates, i.e., downlink data rates that can be supported by the network, to the observed cell-load in the network using a sample training set and prior knowledge. Our framework, therefore, does not require information about powers, channels, etc.

Incorporating prior knowledge in machine learning algorithms for multivariate data[2] with arbitrary dimensions is difficult, and most of the well-known algorithms either do not

[2] Multivariate data in this context means that the input argument (or domain) of the function to be approximated has an arbitrary dimension.

preserve the "shape" (i.e., known properties such as monotonicity, continuity, etc.) of the underlying function or they become too complex for high-dimensional data [Bel05]. An inherent property of the cell-load is that it is monotonic in rates. The study in [Kot16] shows that monotonicity is difficult to incorporate in popular online learning methods even in the case of univariate data. In [Bel05] the author proposes a shape preserving multivariate approximation of scalar monotonic functions that are also Lipschitz. The author shows that Lipschitz continuity of the function to be approximated allows for computing tight upper and lower bounds on the function values. Using these bounds one can obtain an optimal solution in the sense that this solution minimizes a worst-case error of approximation [Suk92, TW80, GW59]. Furthermore, the approximation preserves both the monotonicity and the Lipschitz continuity of the underlying function.

2.1.3 Contribution

This chapter deals with the problem of learning cell-load in RANs as a function of downlink rates given a relatively small training sample set. The assumption of small training sample sets is crucial because modern RAN networks do not permit a long observation and sample acquisition period [see Section 2.1.1]. To cope with this limitation, we propose a robust learning framework that guarantees a minimum worst-case error of approximation. To achieve robustness we incorporate prior knowledge about the cell-load and its relationship with rates. We show that the incorporation of prior knowledge enables us to provide explicit tight bounds that cannot be achieved by using a sample set alone, no matter how large the sample set is.

We now summarize the main contributions of this chapter. We study the feasible rate region which is defined as the set of all rates that can be supported by the network. We shows that the rate region is compact. Morever, we show that there exists a function that maps rates to the cell-load and this function is monotonic and Lipschitz continuous over the feasible rate region. With this prior knowledge in hand, we perform robust learning of the cell-load by using the framework of *minimax approximation* [Suk92, TW80, GW59]. In contrast to [Bel05], where the main concern is to preserve the monotonicity, we show theoretically and by experiments that including the prior knowledge regarding monotonicity results in reduced uncertainty. Our machine learning framework does not require network information such as powers and channel gains in contrast to traditional cell-load approximation methods. The guaranteed performance of our framework with small sample sets makes it suitable in such scenarios where other learning frameworks such as DNNs cannot be applied. We perform simulations in the network simulator NS3 to demonstrate the performance of the algorithm in a realistic cellular wireless network.

Table 2.1: List of Variables

Description	**Symbol**
Number of base stations	M
Number of users	N
Set of base stations	$\mathcal{M} = \{1, 2, \ldots, M\}$
Set of users	$\mathcal{N} = \{1, 2, \ldots, N\}$
Set of users for base station i	$\mathcal{N}(i)$
Rate of user j	$r_j \in \mathbb{R}_{>0}$
Minimum user rate vector	$r_{\min} \in \mathbb{R}_{>0}$
Device SNR between base station i and user j	γ_{ij}
Number of resource blocks	$R \in \mathbb{N}$
Bandwidth of each resource block	$B \in \mathbb{R}_{>0}$
Cell-load	$\boldsymbol{\rho} \in \mathbb{R}^M_{\geq 0}$
Load mapping	$\mathbf{q} : \mathbb{R}^M_{\geq 0} \times \mathbb{R}^N_{>0} \to \mathbb{R}^M_{\geq 0}$
Base station transmit power	$\mathbf{p} \in \mathbb{R}^M_{>0}$
Path-loss between base station i and user j	$G_{i,j} \in \mathbb{R}_{>0}$
Space of continuous functions from X to $\mathcal{Y}$	$C(\mathcal{X}, \mathcal{Y})$
Lipschitz constant	$\mathbf{L} \in \mathbb{R}^M_{\geq 0}$
Euclidean open-ball centered at $\mathbf{x} \in \mathcal{X}$	$\mathcal{B}_{\mathcal{X}}(\mathbf{x}, \delta)$
Network coherence time	$T_{\text{net}} \in \mathbb{R}_{>0}$
Sample acquisition time	$T_{\text{obv}} \in \mathbb{R}_{>0}$
Sample average time	$T_{\text{avg}} \in \mathbb{R}_{>0}$
Sample set size	$K \in \mathbb{N}$

Finally, we compare our framework with standard multivariate learning techniques and show that our method outperforms these techniques for small sample sizes.

2.2 The Load Coupling Model

We consider an urban cellular base station deployment consisting of $M \in \mathbb{N}$ base stations and $N \in \mathbb{N}$ users. We consider the downlink and we denote by $r_j \in \mathbb{R}_{>0}$ the rate of user $j \in \overline{1, N}$ per unit time. We collect the rates of all users in a vector $\mathbf{r} := [r_1, r_2, \cdots, r_N]^\intercal \in \mathbb{R}^N_{>0}$.

We now present the load-coupling model proposed in [Sio12, HYS14], which has been shown to be sufficiently accurate in certain scenarios in practice [FF12, MNK$^+$07, She15]. This model is based on the fact that time-frequency resources available at a base station are divided into physical resource blocks to facilitate resource allocation. The cell-load (at a base station) is defined to be the fraction of available resource blocks that are allocated to support the rates of the users associated with the base station. Resource blocks are allocated to users based on their rates and channel qualities given in terms of their average signal-to-interference-plus-noise ratios (SINRs). In the following we denote by $\mathcal{M} := \{1, 2, \ldots, M\}$ and $\mathcal{N} := \{1, 2, \ldots, N\}$ the set of base stations and users, respectively, and we denote by $\mathcal{N}(i)$ the set of users associated with base station $i \in \mathcal{M}$.

Consider the case where base station $i \in \mathcal{M}$ is serving user $j \in \mathcal{N}(i)$ and denote by $G_{i,j}$ the path-loss between base station i and user j. The load-based SINR model represents the inter-cell interference from base station $k \in \mathcal{M} \setminus \{i\}$ as the product $p_k G_{k,j} \rho_k \geq 0$, where p_k is the fixed transmit power of base station k per resource block, and where $0 < \rho_k \leq 1$ denotes the cell-load at base station k [FF12].[3] With this model in hand, the network layer (averaged) SINR of the wireless link between base station i and user j is expressed as [Sio12, HYS14]

$$\gamma_{ij}(\boldsymbol{\rho}) = \frac{p_i G_{i,j}}{\sum_{k \in \mathcal{M} \setminus \{i\}} p_k G_{k,j} \rho_k + \sigma^2}, \tag{2.1}$$

where $\boldsymbol{\rho} := [\rho_1, \rho_2, ..., \rho_M]^\intercal \in \mathbb{R}^N_{>0}$ is the vector of cell-load values at all base stations in the network and where σ^2 denotes noise power. Note that the denominator in (2.1) provides an interpretation of the cell-load as the probability of inter-cell interference from base station k [Sio12]. For further details of the model including its strengths and weaknesses see [Sio12, HYS14].

Let $R \in \mathbb{N}$ be the total number of resource blocks available at the base station, each with bandwidth $B \in \mathbb{R}_{>0}$. Given SINR $\gamma_{ij}(\boldsymbol{\rho})$, we assume that base station i can reliably transmit at a rate $r^s_{ij} = B \log(1 + \gamma_{ij}(\boldsymbol{\rho}))$ per resource block to user j. Thus, to "support" the rate r_j, base station i has to allocate $\rho_{ij} = \frac{r_j}{r^s_{ij}}$ resource blocks to user j. Summing the resource block consumption over all $\mathcal{N}(i)$, we obtain the cell-load (in terms of total resource consumption) of base station $i \in \overline{1, M}$

$$\rho_i = \frac{1}{RB} \sum_{j \in \mathcal{N}(i)} \frac{r_j}{\log(1 + \gamma_{ij}(\boldsymbol{\rho}))}. \tag{2.2}$$

Note that, we can express the right-hand side of (2.2) for the entire network as a vector-valued mapping

$$\begin{array}{rcl} \mathbf{q} : \mathbb{R}^M_{\geq 0} \times \mathbb{R}^N_{>0} & \to & \mathbb{R}^M_{>0} \\ (\boldsymbol{\rho}, \mathbf{r}) & \mapsto & \begin{bmatrix} \frac{1}{RB} \sum_{j \in \mathcal{N}(1)} \frac{r_j}{\log(1+\gamma_{ij}(\boldsymbol{\rho}))} \\ \vdots \\ \frac{1}{RB} \sum_{j \in \mathcal{N}(M)} \frac{r_j}{\log(1+\gamma_{ij}(\boldsymbol{\rho}))} \end{bmatrix}, \end{array}$$

[3] Note that the cell-load at an "active" base station is always non-zero, and $\rho_i = 0$ implies that base station $i \in \overline{1, M}$ is inactive. On the other hand, $\rho_i = 1$, in the literature, refers to the case where "worst-case/maximum" interference is caused by base station i to other base stations.

which we refer to as the load mapping. Given $\bar{\mathbf{r}} \in \mathbb{R}^N_{>0}$, it follows from (2.2) that the cell-load vector is the solution (if it exists) to the fixed point problem: *Find* $\boldsymbol{\rho}^* = [\rho_1^*, \rho_2^*, ..., \rho_M^*]^\intercal \in \mathbb{R}^M_{\geq 0}$ *such that:*

$$\boldsymbol{\rho}^* = \mathbf{q}(\boldsymbol{\rho}^*, \bar{\mathbf{r}}). \tag{2.3}$$

Since the cell-load is defined as a fraction of the available resources at the base station, a rate vector is feasible (i.e., there are sufficient resource blocks available at all base stations to support rate of every user) if the solution (if it exists) to (2.3) satisfies $\boldsymbol{\rho}^* \leq \mathbf{1}$. For a given supported $\bar{\mathbf{r}} \in \mathbb{R}^N_{>0}$, the solution to (2.3) can be obtained by iterative fixed point algorithms as long as the network information (path-losses, powers, user association, etc. in (2.2)) required by these algorithms is available. In more detail, given $\mathbf{r} \in \mathbb{R}^N_{>0}$, the mapping

$$\Gamma_{\mathbf{r}} : \mathbb{R}^M_{\geq 0} \to \mathbb{R}^M_{>0} : \boldsymbol{\rho} \mapsto \mathbf{q}(\boldsymbol{\rho}, \mathbf{r})$$

is a *positive concave mapping*, so it also belongs to the class of *standard interference functions* [CSS16, Yat95]. Therefore, the following holds:

Fact 2.1 (*The unique fixed point solution*)**.** *[Yat95] Suppose the rate vector* $\bar{\mathbf{r}} \in \mathbb{R}^N_{>0}$ *is feasible, then the solution set of* (2.3) *given by*

$$\mathrm{Fix}(\Gamma_{\bar{\mathbf{r}}}) := \left\{\boldsymbol{\rho}^* \in \mathbb{R}^M_{\geq 0} \mid \mathbf{0} < \Gamma_{\bar{\mathbf{r}}}(\boldsymbol{\rho}^*) = \boldsymbol{\rho}^* \leq \mathbf{1}\right\}$$

contains one fixed point.

As mentioned previously in Section 2.1.3, we incorporate prior knowledge about the cell-load in our learning framework presented in Section 2.4 to ensure robust learning. To this end, Fact 2.2 presents an important property of the cell-load, namely its monotonicity in the rate vector:

Fact 2.2. *[HYS14, Theorem 2] Consider any two feasible rate vectors* $\mathbf{r}^k, \mathbf{r}^j \in \mathcal{R}$ *and the corresponding fixed points* $\boldsymbol{\rho}^j \in \mathrm{Fix}(\Gamma_{\mathbf{r}^j}) \neq \emptyset$ *and* $\boldsymbol{\rho}^k \in \mathrm{Fix}(\Gamma_{\mathbf{r}^k}) \neq \emptyset$*. Then*

$$\mathbf{r}^j \geq \mathbf{r}^k \implies \boldsymbol{\rho}^j \geq \boldsymbol{\rho}^k.$$

In the next section we define and study the feasible rate region, which is the set of all rates supported by the network.

2.3 Properties of the Feasible Rate Region

In light of Fact 2.1 and Fact 2.2, and given the minimum feasible rate vector $\mathbf{r}_{\min} \in \mathbb{R}^N_{>0}$ (e.g., corresponding to the lowest order *modulation and coding scheme* in the network) that induces the cell-load $\boldsymbol{\rho}_{\min} \in \mathbb{R}^M_{>0}$, we are now in a position to define the feasible rate region and the set of cell-load vectors over this set.

Definition 2.1 (*Feasible Rate Region and the Cell Load Set*)**.** The feasible rate region is defined as

$$\mathcal{R} := \{\mathbf{r} \geq \mathbf{r}_{\min} \in \mathbb{R}^N_{>0} \mid (\exists\ \boldsymbol{\rho}^* \in \mathrm{Fix}(\Gamma_{\mathbf{r}}))\ , \boldsymbol{\rho}_{\min} \leq \boldsymbol{\rho}^* \leq \mathbf{1}\}.$$

Similarly, the feasible cell-load set is given by the set of fixed points ([see Fact 2.1]

$$\mathcal{L} := \left\{\boldsymbol{\rho} \in \mathbb{R}^M_{>0} \mid (\exists\ \mathbf{r}^* \in \mathcal{R})\ , \boldsymbol{\rho}_{\min} \leq \Gamma_{\mathbf{r}^*}(\boldsymbol{\rho}) = \boldsymbol{\rho} \leq \mathbf{1}\right\}.$$

In the following we extend the prior knowledge in our learning framework by studying the feasible rate region $\mathcal{R} \in \mathbb{R}^N_{>0}$ in Definition 2.1. In particular, we show in Theorem 2.1 that $\mathcal{R}$ is compact. The compactness of $\mathcal{R}$ is also required for our results in Section 2.4.

Note that $\mathcal{R}$ is bounded from below by $\mathbf{r}_{\min} \in \mathbb{R}^N_{>0}$. Since power, bandwidth, and the total number of resource blocks are fixed in (2.1) and (2.2), and because the cell-load is monotonic in the user rate vector by Fact 2.2, arbitrarily large user rates cannot be supported. We state this fact formally in Lemma 2.1 for completeness. We use this result to prove compactness of $\mathcal{R}$ in Theorem 2.1.

Lemma 2.1. *The feasible rate region is bounded.*

Proof. The set $\mathcal{R}$ is clearly bounded from below. Suppose the set $\mathcal{R}$ is unbounded from above, then there exists at least one unbounded sequence $(\mathbf{r}_n)_{n\in\mathbb{N}} \subset \mathcal{R}$. This implies that at least one component of the vector $\mathbf{r}_n$ grows unboundedly. Let us denote the sequence of this component by $(r_{l,n})_{n\in\mathbb{N}}$ and let the corresponding user be associated with BS $i \in \mathcal{M}$. Every unbounded sequence has an increasing subsequence that diverges to $+\infty$. Let us extract such a subsequence and denote it by $(r_{l,k})_{k\in\mathbb{K}\subset\mathbb{N}}$. Likewise, denote by $(\rho_{i,k})_{k\in\mathbb{K}\subset\mathbb{N}}$ the ith component of the subsequence $(\boldsymbol{\rho}_k)_{k\in\mathbb{K}\subset\mathbb{N}}$, where $\boldsymbol{\rho}_k \in \mathrm{Fix}(\Gamma_{\boldsymbol{r}_k})$. It can be verified that, for fixed $\mathbf{p}$ and bandwidth resources (RB) in (2.2), we have that $\rho_{i,k} = \frac{1}{RB}\sum_{j\in\mathcal{N}(i)} \frac{r_{j,k}}{\log(1+\gamma_{ij}(\mathbf{p},\boldsymbol{\rho}_k))} \geq \frac{1}{RB}\sum_{j\in\mathcal{N}(i)} \frac{r_{j,k}}{\log(1+\gamma_{ij}(\mathbf{p},\mathbf{0}))} \geq \frac{1}{RB}\frac{r_{l,k}}{\log(1+\gamma_{ij}(\mathbf{p},\mathbf{0}))} > 0$, where $\rho_{i,k}$ and $r_{j,k}$ are the ith and jth component of vectors $\boldsymbol{\rho}_k$ and $\mathbf{r}_k$, respectively. Now, note that the lower bound $\frac{1}{RB}\frac{r_{l,k}}{\log(1+\gamma_{ij}(\mathbf{p},\mathbf{0}))}$ grows unboundedly as $r_{l,k} \to \infty$, which in particular implies that $\lim_{k\to\infty} \rho_{i,k} = \infty$. However, this contradicts the fact that, by our definition of feasibility, $(\forall k \in \mathbb{K})\ 0 \leq \rho_{i,k} \leq 1$. Therefore, we conclude that the feasible set $\mathcal{X}$ does not contain any unbounded sequence, so $\mathcal{R}$ is bounded. □

We now present the main result of this section.

Theorem 2.1. *The feasible rate region is compact.*

Proof. Recall from Definition 1.1(b) that a subset of a normed space is closed *if and only if* it contains all of its limit points. We denote by clo($\mathcal{R}$) the *closure* of $\mathcal{R}$ in Definition 2.1, which is the smallest closed set in $\mathbb{R}^N_{>0}$ containing $\mathcal{R}$. Similarly, denote by clo($\mathcal{L}$) the closure of $\mathcal{L}$ in Definition 2.1. Consider an arbitrary sequence $(\mathbf{r}_n, \boldsymbol{\rho}_n)_{n\in\mathbb{N}} \subset \mathcal{R} \times \mathcal{L}$, of tuples consisting of feasible rate vectors and the corresponding cell-load vectors. Suppose $(\mathbf{r}_n, \boldsymbol{\rho}_n) \to (\overline{\mathbf{r}}, \overline{\boldsymbol{\rho}}) \in \text{clo}(\mathcal{R}) \ \times \ \text{clo}(\mathcal{L})$. From (2.3) it follows that, given $\mathbf{r}_n$, $\boldsymbol{\rho}_n$ must be the solution to the fixed point problem with the load mapping $\mathbf{q}$. Therefore, we have

$$(\forall n \in \mathbb{N}) \ \boldsymbol{\rho}_{\min} \leq \boldsymbol{\rho}_n = \mathbf{q}(\boldsymbol{\rho}_n, \mathbf{r}_n) \leq \mathbf{1}.$$

Now, since $\mathbf{q}$ is continuous, we have

$$\begin{aligned} \boldsymbol{\rho}_{\min} &\leq \lim_{n\in\mathbb{N}} \boldsymbol{\rho}_n = \lim_{n\in\mathbb{N}} \mathbf{q}(\boldsymbol{\rho}_n, \mathbf{r}_n) \leq \mathbf{1} \\ \boldsymbol{\rho}_{\min} &\leq \overline{\boldsymbol{\rho}} = \mathbf{q}(\overline{\boldsymbol{\rho}}, \overline{\mathbf{r}}) \leq \mathbf{1} \end{aligned}$$

which implies that $(\overline{\mathbf{r}}, \overline{\boldsymbol{\rho}}) \in \mathcal{R} \times \mathcal{L}$. Thus, every convergent sequence in $\mathcal{R}$ has its limit in $\mathcal{R}$ which implies that $\mathcal{R}$ is closed. Now, according to Lemma 2.1, $\mathcal{R}$ is bounded and recall from Remark 1.1 that every bounded and closed subset of a finite dimensional Euclidean space is compact. □

2.4 Robust Approximation of Cell-Load

Building upon the results from the previous section we formulate the robust learning of cell-load. Note that the cell-load is modeled by the load-coupling model in (2.2). This means that given the network information required by the model, we can calculate the value of the modeled cell-load. However, as mentioned in Section 2.1.1, dynamic wireless networks are in general difficult to model accurately. Therefore, in the following we present a framework to directly estimate the cell-load values in networks that may not follow the cell-load model accurately. We use the cell-load model in this study only to extract some useful prior knowledge. In addition to the monotonicity of the cell-load and the compactness of the feasible rate region $\mathcal{R}$ established in Theorem 2.1, we show in Theorem 2.2 that the function that maps rates to cell-load is continuously differentiable and therefore Lipschitz continuous on $\mathcal{R}$. The Lipschitz continuity is then exploited to solve our robust optimization problem formulated in the following.

Let $\mathcal{D} = \{(\mathbf{r}^k, \boldsymbol{\rho}^k := \mathbf{f}^*(\mathbf{r}^k)) \in \mathcal{R} \times \mathcal{L}, k \in \overline{1, K}\}$ be a sample set of rates and their corresponding cell-load values, where $\mathbf{f}^* : \mathcal{R} \to \mathcal{L}$ is assumed to be a continuous but unknown function, and where $\mathcal{R}$ and $\mathcal{L}$ are defined in Definition 2.1. In the following we denote by $C(\mathcal{R}, \mathcal{L})$ the space of vector-valued continuous functions mapping $\mathcal{R}$ to $\mathcal{L}$, equipped with the norm defined in (1.1).

Our objective is to learn a function $\mathbf{g}^*$ that approximates $\mathbf{f}^*(\mathbf{r})$ for any $\mathbf{r} \in \mathcal{R}$ which is a classical problem considered in, for example, [GW59, Suk92, TW80]. As mentioned in Section 2.1.3 we are interested in a robust approximation of $\mathbf{f}^*$. To this end, we consider the minimax optimization problem that leads to robust solutions under uncertainties:

Problem 2.1. *[Suk92, Bel72] Given $\mathcal{D} = \{(\mathbf{r}^k, \boldsymbol{\rho}^k) \in \mathcal{R} \times \mathcal{L}, k \in \overline{1, K}\}$, find $\mathbf{g}^* \in C(\mathcal{R}, \mathbb{R}^M_{\geq 0})$ that minimizes the worst-case error*[4]

$$E_w : C(\mathcal{R}, \mathbb{R}^M_{\geq 0}) \to \mathbb{R} : \mathbf{g} \mapsto \sup_{\mathbf{f} \in S} \|\mathbf{f} - \mathbf{g}\|_{C(\mathcal{R})}, \tag{2.4}$$

where $S := \{\mathbf{f} \in C(\mathcal{R}, \mathcal{L}) \mid (\forall k \in \overline{1, K})\, \mathbf{f}(\mathbf{r}^k) = \boldsymbol{\rho}^k\}$.

It is known that Problem 2.1 can be solved by restricting $\mathbf{f}^*$ to a compact subset of $C(\mathcal{R}, \mathcal{L})$ and computing finite tight upper and lower bounds on the values $(\forall \mathbf{r} \in \mathcal{R})$ $\mathbf{f}^*(\mathbf{r})$ [GW59, Bel06, Bel05]. Note that if the only information available about $\mathbf{f}^*$ is that it satisfies the interpolation constraints in Problem 2.1, then computing tight bounds on unseen function values $\mathbf{f}^*(\mathbf{r})$ is not possible no matter how large the sample set $\mathcal{D}$ is. However, if we impose an additional restriction on $\mathbf{f}^*$ that satisfies certain properties [GW59], then we can obtain tight bounds such that

$$\boldsymbol{\sigma}_{\mathrm{l}}(\mathbf{r}) \leq \mathbf{f}^*(\mathbf{r}) \leq \boldsymbol{\sigma}_{\mathrm{u}}(\mathbf{r}),$$

where the bounds $\boldsymbol{\sigma}_{\mathrm{l}}(\mathbf{r})$ and $\boldsymbol{\sigma}_{\mathrm{u}}(\mathbf{r})$ can be computed explicitly. An optimal estimation $\mathbf{g}^*(\mathbf{r})$ of $\mathbf{f}^*(\mathbf{r})$ is simply given by the *central algorithm* [Suk92, Bel72]

$$\mathbf{g}^*(\mathbf{r}) = \frac{\boldsymbol{\sigma}_{\mathrm{l}}(\mathbf{r}) + \boldsymbol{\sigma}_{\mathrm{u}}(\mathbf{r})}{2},$$

and the magnitude of uncertainty $\frac{|\boldsymbol{\sigma}_{\mathrm{u}}(\mathbf{r}) - \boldsymbol{\sigma}_{\mathrm{l}}(\mathbf{r})|}{2}$ is minimal [Cal14]. Note that the central algorithm is one of the point-wise estimators discussed in *bounded error estimation* in Section 1.7. Therefore, no matter how small the sample set $\mathcal{D}$ is we are guaranteed the minimum worst-case error (2.4) for a given $\mathcal{D}$. In other words, the approximation is robust against the uncertainty resulting from small sample sets.

[4] The worst-case error is finite because the value of the cell-load cannot exceed 1.

It is known that Lipschitz functions in $C(\mathcal{R}, \mathcal{L})$ satisfy the required properties mentioned above, where the Lipschitz continuity plays the role of the nonlinear restriction mentioned above [see, e.g., [Bel06, Bel05, Suk92]]. Furthermore, the bounds $\boldsymbol{\sigma}_\mathrm{l}$ and $\boldsymbol{\sigma}_\mathrm{u}$ can be computed easily for Lipschitz functions. To exploit these facts, we show in Theorem 2.2 that $\mathbf{f}^*$ belongs to the class of $\mathbf{L}$-Lipschitz-Monotone Functions (LIMF) [see Definition 1.4]. Moreover, Proposition 2.1 shows that this class is a compact subset of $C(\mathcal{R}, \mathcal{L})$. We then use these facts to solve Problem 2.1. The computation of the bounds $\boldsymbol{\sigma}_\mathrm{l}(\mathbf{r})$ and $\boldsymbol{\sigma}_\mathrm{u}(\mathbf{r})$ is presented in Fact 2.3.

In the following we denote by $\widetilde{\mathcal{R}} \subset \mathbb{R}^N_{>0}$ the set of all rate vectors (not necessarily feasible/supported) for which there exists a fixed point solution of (2.3), i.e., $\widetilde{\mathcal{R}} := \{\overline{\mathbf{r}} \in \mathbb{R}^N_{>0} \,|(\exists\, \overline{\boldsymbol{\rho}} \in \mathbb{R}^M_{>0})\ \overline{\boldsymbol{\rho}} = \mathbf{q}(\overline{\boldsymbol{\rho}}, \overline{\mathbf{r}})\}$. So we have $\mathcal{R} \subset \widetilde{\mathcal{R}}$.

Theorem 2.2. *Consider the load mapping* $\mathbf{q} : \mathbb{R}^M_{\geq 0} \times \mathbb{R}^N_{>0} \to \mathbb{R}^M_{>0}$ *in* (2.3).

a). There exists a continuously differentiable function $\mathbf{f}^{imp} : \widetilde{\mathcal{R}} \to \mathbb{R}^M_{>0}$ *such that* $(\forall \overline{\mathbf{r}} \in \widetilde{\mathcal{R}})\ \mathbf{f}^{imp}(\overline{\mathbf{r}}) = \overline{\boldsymbol{\rho}} = \mathbf{q}(\overline{\boldsymbol{\rho}}, \overline{\mathbf{r}})$.

b). The restriction of $\mathbf{f}^{imp}$ *to the feasible rate region* $\mathcal{R} \subset \widetilde{\mathcal{R}}$ *is a LIMF function.*

Proof. a). From the uniqueness of the fixed point solution of (2.3) it follows that, for two solution pairs $(\overline{\boldsymbol{\rho}_1}, \overline{\mathbf{r}}_1)$ and $(\overline{\boldsymbol{\rho}_2}, \overline{\mathbf{r}}_2)$, if $\overline{\boldsymbol{\rho}}_1 \neq \overline{\boldsymbol{\rho}}_2$, then we must have $\overline{\mathbf{r}}_1 \neq \overline{\mathbf{r}}_2$. Thus, there exists a function $\mathbf{f}^{\mathrm{imp}} : \widetilde{\mathcal{R}} \to \mathbb{R}^M_{>0} : \overline{\mathbf{r}} \mapsto \mathbf{f}^{\mathrm{imp}}(\overline{\mathbf{r}}) = \mathbf{q}(\mathbf{f}^{\mathrm{imp}}(\overline{\mathbf{r}}), \overline{\mathbf{r}})$ that maps every feasible rate vector to a unique fixed point. We now show that $\mathbf{f}^{\mathrm{imp}}$ is continuously differentiable on $\widetilde{\mathcal{R}}$.

Consider the function $\mathbf{g} : \mathbb{R}^N_{>0} \times \mathbb{R}^M_{>0} \to \mathbb{R}^M$ defined as

$$\mathbf{g}(\mathbf{r}, \boldsymbol{\rho}) := \boldsymbol{\rho} - \mathbf{q}(\boldsymbol{\rho}, \mathbf{r}),$$

where $\mathbf{q}$ is the load mapping in (2.3). We have

$$(\forall \overline{\boldsymbol{\rho}} \in \mathbb{R}^M_{>0})\ (\forall \overline{\mathbf{r}} \in \widetilde{\mathcal{R}})\ \overline{\boldsymbol{\rho}} = \mathbf{f}^{\mathrm{imp}}(\overline{\mathbf{r}}) \iff \mathbf{g}(\overline{\mathbf{r}}, \overline{\boldsymbol{\rho}}) = \mathbf{0}.$$

We now show that $\mathbf{g}$ is continuously differentiable, and the Jacobian matrix $\boldsymbol{\nabla}^{\mathbf{g}}_{\boldsymbol{\rho}}(\overline{\mathbf{r}}, \overline{\boldsymbol{\rho}})$ is non-singular (invertible), on $\widetilde{\mathcal{R}} \times \mathbb{R}^M_{>0}$ [see Fact 1.2]. To show that $\mathbf{g}$ is continuously differentiable, we show that the Jacobians $\boldsymbol{\nabla}^{\mathbf{g}}_{\mathbf{r}}$ and $\boldsymbol{\nabla}^{\mathbf{g}}_{\boldsymbol{\rho}}$ are continuous. The two Jacobians are given in Section 2.7.1 and Section 2.7.2, respectively, and it can be verified that they are continuous. The invertibility of $\boldsymbol{\nabla}^{\mathbf{g}}_{\boldsymbol{\rho}}(\overline{\mathbf{r}}, \overline{\boldsymbol{\rho}})$ is shown in Section 2.7.3. Therefore, according to Fact 1.2, $\mathbf{f}^{\mathrm{imp}}$ is continuously differentiable.

b). According to part (a) and Fact 1.2, the Jacobian $\boldsymbol{\nabla}_{\mathbf{r}}^{\mathbf{f}^{\text{imp}}}$ is continuous on $\widetilde{\mathcal{R}}$. Denote by $\mathbf{f} : \mathcal{R} \to \mathcal{L}$ and $\boldsymbol{\nabla}_{\mathbf{r}}^{\mathbf{f}}$, the restriction of $\mathbf{f}^{\text{imp}}$ and $\boldsymbol{\nabla}_{\mathbf{r}}^{\mathbf{f}^{\text{imp}}}$, respectively, to the set of feasible rate vectors $\mathcal{R} \subset \widetilde{\mathcal{R}}$. Since $\mathcal{R}$ is compact according to Theorem 2.1, $\boldsymbol{\nabla}_{\mathbf{r}}^{\mathbf{f}}$ is bounded on $\mathcal{R}$ according to the *extreme value theorem* [Mun00] which implies that $\exists \mathbf{L} \in \mathbb{R}_{\geq 0}^{M}$ such that $\mathbf{f}$ is $\mathbf{L}$-Lipschitz on $\mathcal{R}$. Moreover, by Fact 2.2, $\mathbf{f}$ is monotonic on $\mathcal{R}$, so $\mathbf{f}$ is a LIMF function [see Definition 1.4].

□

In the following we denote by $\mathcal{F} \subset C(\mathcal{R}, \mathcal{L})$ the class of LIMF functions $\mathbf{f} : \mathcal{R} \to \mathcal{L}$ with a given $\mathbf{L} \in \mathbb{R}_{\geq 0}^{M}$ [see Definition 1.4]. Before we proceed further, we obtain the following important result:

Proposition 2.1. *The class $\mathcal{F} \subset C(\mathcal{R}, \mathcal{L})$ of LIMF functions, with a given $\mathbf{L} = [L_1, L_2, \cdots, L_M]^{\intercal} \in \mathbb{R}_{\geq 0}^{M}$, is compact.*

Proof. The class $\mathcal{F} \subset C(\mathcal{R}, \mathcal{L})$ satisfies the following properties:

a). *Boundedness*: $\mathcal{F}$ is bounded because $(\forall \mathbf{f} \in \mathcal{F})$ $\|\mathbf{f}\|_{C(\mathcal{R})} \leq 1$.

b). *Equicontinuity*: Since $\mathcal{F}$ is a set of $\mathbf{L}$-Lipschitz functions, $\mathcal{F}$ is an equicontinuous subset of $C(\mathcal{R}, \mathcal{L})$ [see Lemma 2.2 in Section 2.7.4) for a proof].

c). *Closedness*: The class $\mathcal{F}$ can be written as $\mathcal{F} = \mathcal{F}^{\text{Lip}} \bigcap \mathcal{F}^{\text{mon}}$, where $\mathcal{F}^{\text{Lip}}$ and $\mathcal{F}^{\text{mon}}$ are the sets of $\mathbf{L}$-Lipschitz functions and continuous monotone functions, respectively, in $C(\mathcal{R}, \mathcal{L})$. Recall that the intersection of two closed sets is closed. Therefore, it is sufficient to show that $\mathcal{F}^{\text{Lip}}$ and $\mathcal{F}^{\text{mon}}$ are closed sets. For completeness, we show in Lemma 2.3 in Section 2.7.5 that $\mathcal{F}^{\text{mon}}$ and $\mathcal{F}^{\text{Lip}}$ are closed sets.

The proposition now follows from Fact 1.1. □

2.4.1 Minimax Optimal Approximation

We are now in a position to incorporate the prior information obtained in previous sections into Problem 2.1. Moreover, we formally state the robust learning problem considered in this chapter as an optimization problem.

Definition 2.2 (*Minimax Optimal Approximation*)**.** *Let $\mathcal{D} = \{(\mathbf{r}^k, \boldsymbol{\rho}^k) \in \mathcal{R} \times \mathcal{L}\}_{k=1}^{K}$ be a sample set and assume that $(\forall k \in \overline{1, K})$ $\boldsymbol{\rho}^k := \mathbf{f}^*(\mathbf{x}^k)$ are values generated by an unknown function $(\mathcal{F} \ni)$ $\mathbf{f}^* : \mathcal{R} \to \mathcal{L}$, where $\mathcal{F} \subset C(\mathcal{R}, \mathcal{L})$ is a set of LIMF functions with a given $\mathbf{L} \in \mathbb{R}_{\geq 0}^{M}$. The robust learning problem can be then stated as follows:*

Problem 2.2. *[Suk92, Bel05, Bel72] Find* $\mathbf{g}^* \in \mathcal{F}$ *such that*

$$\mathbf{g}^* \in \underset{\mathbf{g}\in\mathcal{F}}{\arg\min}\ E_{max}(\mathbf{g}),$$

where $E_{max}(\mathbf{g}) := \max_{\mathbf{f}\in\mathcal{F}} \|\mathbf{f}-\mathbf{g}\|_{C(\mathcal{R})}$, *such that* $(\forall k \in \overline{1,K})\ \mathbf{f}(\mathbf{r}^k) = \mathbf{g}(\mathbf{r}^k) = \boldsymbol{\rho}^k$.

Note that a solution to Problem 2.2 is "shape preserving" because $\mathbf{g}^* \in \mathcal{F}$, i.e., the approximation preserves the Lipschitz continuity and monotonicity. We show in Proposition 2.2 that enforcing such shape preservation (based on prior information) results in less uncertainty compared to the case where this prior information is omitted. To this end, we first need to show how to solve Problem 2.2.

We use the framework in [Bel05] to obtain a solution to Problem 2.2. The following fact summarizes the important properties of an optimal solution obtained based on this framework.

Fact 2.3. *[Bel05] Let* $\mathcal{D} = \{(\mathbf{r}^k, \boldsymbol{\rho}^k) \in \mathcal{R} \times \mathcal{L}\}_{k=1}^{K}$ *be a dataset generated by an unknown function* $\mathbf{f}^* \in \mathcal{F}$, *where* $\mathcal{F}$ *is the set of LIMF functions with the same* $\mathbf{L} := [L_1, L_2, \cdots, L_M]^\intercal \in \mathbb{R}^M_{\geq 0}$. *Then, the following holds:*

a). A minimax optimal approximation $\mathbf{g}^*$ *of* $\mathbf{f}^* \in \mathcal{F}$ *can be constructed component-wise by*

$$(\forall i \in \overline{1,M})\ (\forall \mathbf{r} \in \mathcal{R})\ g_i^*(\mathbf{r}) = \frac{\sigma_l^i(\mathbf{r}) + \sigma_u^i(\mathbf{r})}{2}, \tag{2.5}$$

where $\sigma_l^i(\mathbf{r}) = \max_k\{\rho_i^k - L_i\|(\mathbf{r}^k - \mathbf{r})_+\|\}$, $\sigma_u^i(\mathbf{r}) = \min_k\{\rho_i^k + L_i\|(\mathbf{r} - \mathbf{r}^k)_+\|\}$, *and* $L_i \in \mathbb{R}_{\geq 0}$ *is the Lipschitz constant of the ith component* f_i^* *of* $\mathbf{f}^*$.

b). The approximation preserves the $\mathbf{L}$*-Lipschitz continuity and monotonicity, i.e.,* $\mathbf{g}^*$ *is* $\mathbf{L}$*-Lipschitz and monotonic.*

c). $\mathbf{g}^*$ *interpolates the sample set* $\mathcal{D}$.

Remark 2.1 (Prior Knowledge Decreases Uncertainty)*.* The study [Bel05] is concerned with shape preserving approximation and it does not consider learning from a small sample set. However, we show in Proposition 2.2 that (except for one particular case) excluding prior information regarding monotonicity worsens at least one of the bounds in Fact 2.3(a) during generalization on unseen data and this therefore increases uncertainty. We also evaluate this fact empirically in Section 2.6.2 in a realistic wireless network.

The lower and upper bounds without monotonicity constraints in Fact 2.3 are given by $(i \in \overline{1,M})$ $\eta_{\mathrm{l}}^i(\mathbf{r}) = \max_k\{\rho_i^k - L_i\|\mathbf{r}^k - \mathbf{r}\|\}$ and $\eta_{\mathrm{u}}^i(\mathbf{r}) = \min_k\{\rho_i^k + L_i\|\mathbf{r} - \mathbf{r}^k\|\}$. Let $\mathrm{U}_{\mathrm{mon}}(\mathbf{r}) := \frac{|\sigma_{\mathrm{u}}^i(\mathbf{r}) - \sigma_{\mathrm{l}}^i(\mathbf{r})|}{2}$ denote the magnitude of uncertainty calculated from the bounds in Fact 2.3, and let $\mathrm{U}(\mathbf{r}) := \frac{|\eta_{\mathrm{u}}^i(\mathbf{r}) - \eta_{\mathrm{l}}^i(\mathbf{r})|}{2}$ denote the magnitude of uncertainty without monotonicity in the framework.

Proposition 2.2. *Let* $\mathbf{r} \notin \mathcal{D} = \{(\mathbf{r}^k, \boldsymbol{\rho}^k) \in \mathcal{R} \times \mathcal{L}\}_{k=1}^K$*, where* $\mathcal{D}$ *is the data set in Fact 2.3. Then* $U_{mon}(\mathbf{r}) \leq U(\mathbf{r})$ *if*

a). ($\exists\ k^* \in argmax_k\{\rho_i^k - L_i\|(\mathbf{r}^k - \mathbf{r})\|\}$*)* $\mathbf{r}^{k^*} \geq \mathbf{r}$*, and*

b). ($\exists\ j^* \in argmin_j\{\rho_i^j + L_i\|(\mathbf{r} - \mathbf{r}^j)\|\}$*)* $\mathbf{r}^{j^*} \leq \mathbf{r}$*.*

If a) and b) are not satisfied simultaneously then $U_{mon}(\mathbf{r}) < U(\mathbf{r})$*.*

Proof. Consider two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}_{\geq 0}^N$ such that $\mathbf{x} \neq \mathbf{y}$. If $\mathbf{x} \geq \mathbf{y}$, then $\|(\mathbf{x} - \mathbf{y})_+\| = \|(\mathbf{x} - \mathbf{y})\|$ and $\|(\mathbf{y} - \mathbf{x})_+\| < \|(\mathbf{x} - \mathbf{y})\|$. Similarly, if $\mathbf{x} \leq \mathbf{y}$, then $\|(\mathbf{x} - \mathbf{y})_+\| < \|(\mathbf{x} - \mathbf{y})\|$ and $\|(\mathbf{y} - \mathbf{x})_+\| = \|(\mathbf{x} - \mathbf{y})\|$. If $\mathbf{x}$ and $\mathbf{y}$ are incomparable then $\|(\mathbf{y} - \mathbf{x})_+\| < \|(\mathbf{x} - \mathbf{y})\|$ and also $\|(\mathbf{x} - \mathbf{y})_+\| < \|(\mathbf{x} - \mathbf{y})\|$.

Now, if conditions a) and b) are satisfied simultaneously, then (by condition a)) for the lower bound we have

$$\begin{aligned}
\eta_{\mathrm{l}}^i(\mathbf{r}) &= \{\rho_i^{k^*} - L_i\|(\mathbf{r}^{k^*} - \mathbf{r})\|\} \\
&= \{\rho_i^{k^*} - L_i\|(\mathbf{r}^{k^*} - \mathbf{r})_+\|\} \\
&\leq \max_k\{\rho_i^k - L_i\|(\mathbf{r}^k - \mathbf{r})_+\|\} \\
&= \sigma_{\mathrm{l}}^i(\mathbf{r}).
\end{aligned}$$

Similarly, (by condition b)) $\sigma_{\mathrm{u}}^i(\mathbf{r}) \leq \eta_{\mathrm{u}}^i(\mathbf{r})$. This proves the first claim of the proposition. Now suppose condition a) is violated, i.e., either $\mathbf{r}^{k^*} \leq \mathbf{r}$ or $\mathbf{r}^{k^*}$ and $\mathbf{r}$ are incomparable, then from the above discussion

$$\begin{aligned}
\eta_{\mathrm{l}}^i(\mathbf{r}) &= \{\rho_i^{k^*} - L_i\|(\mathbf{r}^{k^*} - \mathbf{r})\|\} \\
&< \{\rho_i^{k^*} - L_i\|(\mathbf{r}^{k^*} - \mathbf{r})_+\|\} \\
&\leq \max_k\{\rho_i^k - L_i\|(\mathbf{r}^k - \mathbf{r})_+\|\} \\
&= \sigma_{\mathrm{l}}^i(\mathbf{r}).
\end{aligned}$$

Similarly, if condition b) is violated, $\sigma_{\mathrm{u}}^i(\mathbf{r}) < \eta_{\mathrm{u}}^i(\mathbf{r})$ and the second claim follows. □

The consequence of Proposition 2.2 is that $\mathrm{U}_{\mathrm{mon}}(\mathbf{r}) < \mathrm{U}(\mathbf{r})$ whenever $\mathbf{r}$ violates either of the two conditions in Proposition 2.2. Therefore, including prior information regarding monotonicity provably improves generalization on unseen data in our framework.

2.4.2 Complexity

The complexity of the closed-form computation (2.5) is linear in the sample size K, i.e., the complexity is $O(K)$. Since we consider small sample sizes, this computation is fast. Moreover, (2.5) has a natural distributed form because $(\forall i \in \overline{1, M})$ $g_i(\mathbf{r})$ can be computed independently. Therefore, the complexity is independent of the number of base stations M.

2.5 Implementation in a Wireless Network

We have shown in Theorem 2.2 that there exists an implicit function $(\forall i \in \overline{1, M})$ $f_i : \mathcal{R} \to]0, 1]$ mapping every $\mathbf{r} \in \mathcal{R}$ to a cell-load value $\rho_i \in]0, 1]$ at base station i. Furthermore, Fact 2.3 shows that given a sample set $\mathcal{D}(i) = \{(\mathbf{r}^k, f_i(\mathbf{r}^k)) \in \mathcal{R} \times]0, 1]\}_{k=1}^{K}$ (at base station i) and the knowledge of the Lipschitz constant L_i, we can easily approximate the cell-load value $f_i(\mathbf{r})$ for $\mathbf{r} \notin \mathcal{D}(i)$. In this section we show how to implement our framework in an OFDMA-based wireless cellular network. To this end, we first look at how to calculate the cell-load, and then we show how to obtain an appropriate sample set at a base station.

2.5.1 Cell-load Calculation

In OFDMA-based networks, such as LTE networks, time is divided into fixed length *slots* [NS3]. During each slot, if a base station is active, it transmits to one or more users on a block of frequencies in its cell. Therefore, users are allocated slots in time and bandwidth in frequency according to their rate requirements. A slot together with its bandwidth is commonly referred to as a *physical resource block*. To calculate the cell-load, we record the fraction of the total available physical resource blocks allocated by a base station on average during a total time period of $T_{\text{avg}} > 0$, where T_{avg} is a design parameter.

2.5.2 Obtaining a Sample Set

We denote by $T_{\text{net}} > 0$ the network coherence time during which the environment (network topology, channels, rate distribution, etc.) is assumed to be constant. Let $T_{\text{obv}} < T_{\text{net}}$ denote the sample observation time. We divide T_{obv} in $K \in \mathbb{N}$ time windows of duration T_{avg} each as shown in Figure 2.1. To obtain a sample set $\mathcal{D}(i) = \{(\mathbf{r}^k, \rho_i^k = f_i(\mathbf{r}^k))\}_{k=1}^{K}$ at each base station $i \in \overline{1, M}$, the cell-load values $\rho_i^k = f_i(\mathbf{r}^k)$ can be calculated as in Section 2.5.1 for each time window $k \in \overline{1, K}$. The base stations can exchange the rate values of users associated with them with other base stations to obtain the rate vectors $\mathbf{r}^k$.

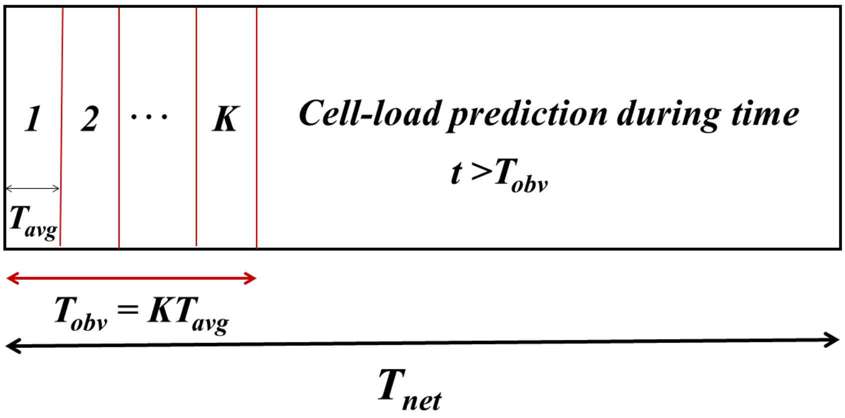


Figure 2.1: Learning Timeline: During each slot $k \in \overline{1,K}$ of length T_{avg} we obtain a sample $(\mathbf{r}^k, \rho_i^k)$ by observing the proportion of resource blocks consumed to support rate $\mathbf{r}^k$ on average during T_{avg}.

In the following we assume that a sample set $\mathcal{D}(i) = \{(\mathbf{r}^k, \rho_i^k = f_i(\mathbf{r}^k)) \in \mathcal{R} \times]0,1]\}_{k=1}^K$, is available at time $t = T_{\text{obv}}$ at base station $i \in \overline{1,M}$. We also omit the index i since the same procedure is carried out at each base station.

2.5.3 Obtaining a Compatible Sample Set

Note that the cell-load values calculated in a real network do not follow the cell-load model exactly. In more detail, instead of the sample set $\mathcal{D} = \{(\mathbf{r}^k, \rho^k = f(\mathbf{r}^k))\}_{k=1}^K$, we assume that an inaccurate sample set $\mathcal{D}^{\text{error}} = \{(\mathbf{r}^k, y^k = f(\mathbf{r}^k) + \epsilon(\mathbf{r}^k))\}_{k=1}^K$ is available; $\epsilon(\mathbf{r}^k) \geq 0$ is the inaccuracy/error which is bounded for every $\mathbf{r}$.[5] As a consequence, for a given value of the Lipschitz constant $L \in \mathbb{R}_{\geq 0}$, $\mathcal{D}^{\text{error}}$ may not be compatible with the monotonicity of f. Therefore, and if required, it must be smoothed to obtain a compatible set. Furthermore, in practice the prior information about the Lipschitz constant L is often unavailable, so its value must be estimated from the set $\mathcal{D}^{\text{error}}$. In more detail, we first estimate the Lipschitz constant by $\tilde{L} := \max_{k \neq j} \frac{|y^k - y^j|}{\|\mathbf{r}^k - \mathbf{r}^j\|}$ [Str73].[6] Given an estimate $\tilde{L}$ of the Lipschitz constant, we perform monotone-smoothing of $\mathcal{D}^{\text{error}}$. Given an estimate $\tilde{L}$

[5] Our approximation framework is a special case of bounded error estimation/robust set-membership estimation [MT85, MV91] which was developed for scenarios where the inaccuracy is unknown but bounded.

[6] There exist more sophisticated methods of estimating the Lipschitz constant such as the method proposed in [Bel05]. But these methods are not the focus of this study and they add substantial complexity to the algorithm.

Algorithm 1 Cell-load Learning at each Base Station

```
→ Initialization
    • Fix $K > 0$ and $T_{\text{avg}} > 0$.
→ Sample Acquisition (while $t < T_{\text{obv}}$)
    • Exchange user rate with other base stations.
    • Observe the sample set $\mathcal{D}^{\text{noise}} = \{(\mathbf{r}^k, y^k = f(\mathbf{r}^k) + \epsilon(\mathbf{r}^k))\}_{k=1}^K$ (Section 2.5.2).
→ Training (at $t = T_{\text{obv}}$)
    • Perform the estimation of $L$ (Section 2.5.3).
    • Perform data smoothing to obtain a compatible $\mathcal{D}^{\text{com}} = \{(\mathbf{r}^k, \tilde{\rho}^k)\}_{k=1}^K$ (Section 2.5.3).
→ On-Demand Prediction (at $t > T_{\text{obv}}$)
    • Given a new rate vector $\mathbf{r} \in \mathcal{R}$, perform the computation (2.5) in Fact 2.3
          $g(\mathbf{r}) = \frac{1}{2}(\max_k\{\tilde{\rho}^k - L\|(\mathbf{r}^k - \mathbf{r})_+\|\}) + \frac{1}{2}(\min_k\{\tilde{\rho}^k + L\|(\mathbf{r} - \mathbf{r}^k)_+\|\})$.
```

of the Lipschitz constant, we can now consider the monotone-smoothing problem which is formulated as a standard convex optimization problem.

The author in [Bel05] has shown that a sample set $\mathcal{D}^{\text{com}} := \{(\mathbf{r}^k, \tilde{\rho}^k)\}_{k=1}^K$ is compatible with the monotonicity if and only if it satisfies the following set of linear constraints [Bel05, Proposition 4.1]

$$(\forall k \in \overline{1,K})\ (\forall j \in \overline{1,K})\ \tilde{\rho}^k - \tilde{\rho}^j \leq \tilde{L}\|(\mathbf{r}^k - \mathbf{r}^j)_+\|. \tag{2.6}$$

Given the measured sample set $\mathcal{D}^{\text{noise}} = \{(\mathbf{r}^k, y^k)\}_{k=1}^K$, we look for a compatible set $\mathcal{D}^{\text{com}} = \{(\mathbf{r}^k, \tilde{\rho}^k)\}_{k=1}^K$ (that satisfies (2.6)) that is closest to $\mathcal{D}^{\text{noise}}$ in the $\|\cdot\|_1$ sense. In more detail, let $\mathbf{y} = [y^1, \cdots, y^K]^\intercal$ and $\tilde{\boldsymbol{\rho}} = [\tilde{\rho}^1, \cdots, \tilde{\rho}^K]^\intercal$, then we minimize

$$\|\mathbf{y} - \tilde{\boldsymbol{\rho}}\|_1 = \sum_{k=1}^{K} |\tilde{\rho}^k - y^k|. \tag{2.7}$$

We now formalize this problem as a standard linear program (LP) which can be solved easily by any standard convex solver. Denote the kth residual in (2.7) by $q^k := \tilde{\rho}^k - y^k$ and split q^k into two parts q_+^k and q_-^k such that $q^k = q_+^k - q_-^k$. Substituting $(\forall l \in \overline{1,K})$ $q^l + y^l$ for $\tilde{\rho}^l$ into (2.6) and (2.7), the monotone-smoothing problem can be written as an

LP [Bel05]

$$\begin{aligned} \underset{q_+^k, q_-^k \geq 0}{\text{minimize}} \quad & \sum_{k=1}^{K} |q^k| \\ \text{subject to} \quad & (\forall k \in \overline{1, K})\ (\forall j \in \overline{1, K}) \\ & q^k - q^j \leq y^j - y^k + \tilde{L} \|(\mathbf{r}^k - \mathbf{r}^j)_+\|, \end{aligned} \tag{2.8}$$

where $|q^k| = q_+^k + q_-^k$, and where $q_+^k, q_-^k \geq 0$ are the optimization variables. The smoothed compatible values follow from $\tilde{\rho}^k = y^k + q^k$.

Note that the complexity of the above LP, among other things, increases with the number of constraints $(K \times (K-1))$. Since we consider small sample sizes K and the constraint matrix, with rows given by (2.8), is sparse, the above LP can be solved fast with standard convex solvers that exploit sparsity [LPV]. Therefore, the complexity of the smoothing step, which is performed only once after sample acquisition, is not of a practical concern.

2.5.4 Algorithm

The robust cell-load learning algorithm is presented in Algorithm 1. Note that Algorithm 1 can be executed independently in parallel at each base station. The *Sample Acquisition* step corresponds to the acquisition of the training sample set as explained in Section 2.5.2, whereas *Training* refers to Lipschitz constant estimation and the data smoothing process as presented in Section 2.5.3. The *On-Demand Prediction* refers to the approximation of the cell-load value for a new rate vector during time period $T_{\text{net}} - T_{\text{obv}}$ [also see Figure 2.1].

2.6 Numerical Evaluation

In this section we evaluate the robust learning framework presented in Section 2.4.1 by simulation. To evaluate the learning techniques in a realistic cellular network, simulations are performed in the network simulator (NS3) [NS3]. We focus on the following aspects in this numerical evaluation:

1. We only use the load-coupling model in this study to establish some prior knowledge about the cell-load in a real cellular network. We show in the simulations that our learning framework is able to predict the cell-load sufficiently accurately in a realistic cellular network in NS3. This is significant because models are only idealizations, and they may not capture the true behavior of cellular networks.

2. We have shown in Proposition 2.2 that including prior knowledge decreases the uncertainty. We demonstrate this by comparing our learning framework with full prior knowledge with the case in which the prior information regarding the monotonicity of the cell-load with respect to rate is not included in the framework.

3. Finally, we compare our method to standard multivariate regression techniques. We show the effect of sample size K and the size of the network (i.e., the number of users N and base stations M) on the quality of approximation.

In the next section, we present the LTE simulation framework in NS3.

2.6.1 Network Simulator (NS3) and Scenario

We perform simulation in NS3 using the LTE model, the details of which can be found in [NS3]. The load-coupling model is evaluated in the LTE downlink in certain scenarios in [She15]. Briefly, NS3 is a well-known discrete-event network simulator widely used in educational research and industry due to its accuracy in simulating computer networks such as LTE. The granularity of the LTE model in NS3 is up to the resource block level which allows for accurate packet scheduling and calculation of inter-cell interference. We chose the round robin scheduler at the MAC layer. The reason is that the fairness inherent in the simple cyclic scheduling is more likely to ensure that the minimum data rate requirement of all users are met, which may not be the case with other more complex scheduling algorithms [DPS14]. The modulation and coding scheme and the resource block allocation are chosen based on the wide-band channel quality indicator (CQI). The CQI is calculated based on the average received SINR. An example simulation topology is shown in Figure 2.2.

Users and base stations are distributed uniformly in the service area of 200 × 200 meters. We perform simulations for $M \in \{3, 5, 6, 7, 8, 9, 10\}$ base stations with $N \in \{30, 50, 60, 70, 80, 90, 100\}$ users. Users are associated with the base station to which they have the lowest path-loss. To generate training and test data, the data rates are distributed uniformaly between 0.1×10^6 bits/s and 1×10^6 bits/s. The important simulation parameters are shown in Table 2.2. Other parameters were chosen as default in NS3. The simulation time was chosen to be 1 second which is equal to the length T_{avg} of each averaging time slot/window in Figure 2.1 and Algorithm 1. The cell-load values are calculated according to Section 2.5.1.

2.6.2 Results

We now present our numerical results. We use Algorithm 1 to perform the robust learning of cell-load proposed in this study. We present the results for cell-load learning at a

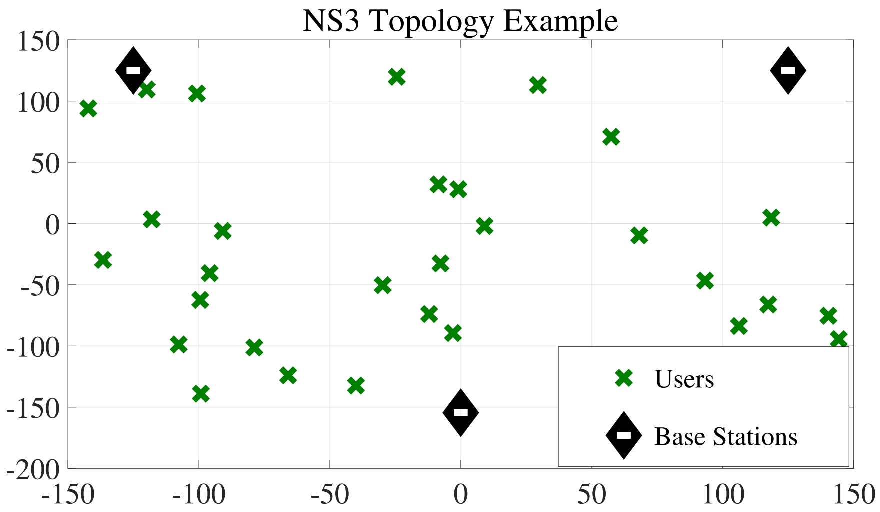


Figure 2.2: NS3 Topology Example: There are 30 users and 3 base stations. Users are associated to base stations with the least path-loss. Simulations are performed for the base station located at $(0, -150)$.

single base station. To obtain reliable statistics we consider 50 topologies (with different user locations, base station locations, and user associations) for each value of N and we let $M = N/10$. Note that scaling the number of base stations with an increase in the number of users is necessary to ensure that rate requirements of users are met. The objective of the simulation is to observe the effect of sample size and the network size on the approximation. For each fixed topology, we perform 100 experiments for each value of $K \in \{10, 20, \dots, 100\}$. During each experiment, a sample set $\mathcal{D}^{\text{error}} = \{(\mathbf{r}^k, y^k)\}_{k=1}^{K}$ is generated independently at random and the *Training Step* is performed in Algorithm 2 to obtain a compatible training sample set $\mathcal{D}^{\text{com}}$. Validation/prediction is performed for an independent test sample set of size 1000 with rate vectors $\mathbf{r} \notin \mathcal{D}^{\text{com}}$. All results are averaged over 100 experiments and then over 50 topologies to obtain reliable statistics.

Effect of Prior Information

In this section we compare our framework's performance with and without the prior information regarding the monotonicity of the cell-road with respect to rate [see Remark 2.1]. For this simulation we consider $M = 3$ and $N = 30$. Note that the objective of this rather theoretical comparison is to confirm the result of Proposition 2.2 in a realistic simulation. This comparison is performed with an ideal Lipschitz constant L^{ideal} that can be obtained by using the method in Section 2.5.3 but by using both the training sample set and the test sample set. This way L^{ideal} is a good approximation of the true Lipschitz constant. We chose an ideal Lipschitz constant because in this section we want to focus only on

Table 2.2: NS3 Simulation Parameters

Description	**Value**
Number of base stations M	3
Number of users N	30
Base station height	30 m
User height	1.5 m
Noise figure base station	5 dB
Noise figure user	9 dB
Base station power	46 dBm
Min/Max user rate	$0.1 \times 10^6/1 \times 10^6$ bits/s
Simulation area	150×150 m
Simulation time	1 s
Total bandwidth	10 MHz
Total number of resource blocks	50
Path-loss model	Log-Distance Propagation Loss
SRS periodicity	80×10^{-3} s
Internet application	On-Off with Ipv4

the effect of including prior knowledge regarding monotonicity of the cell-load in rate in a realistic cellular network, and this requires an accurate calculation of function bounds in Section 2.4.1. However, the comparison with *state-of-art* techniques in Section 2.6.2, which is of a more practical significance, is performed with the Lipschitz constant that is estimated from only the training data set.

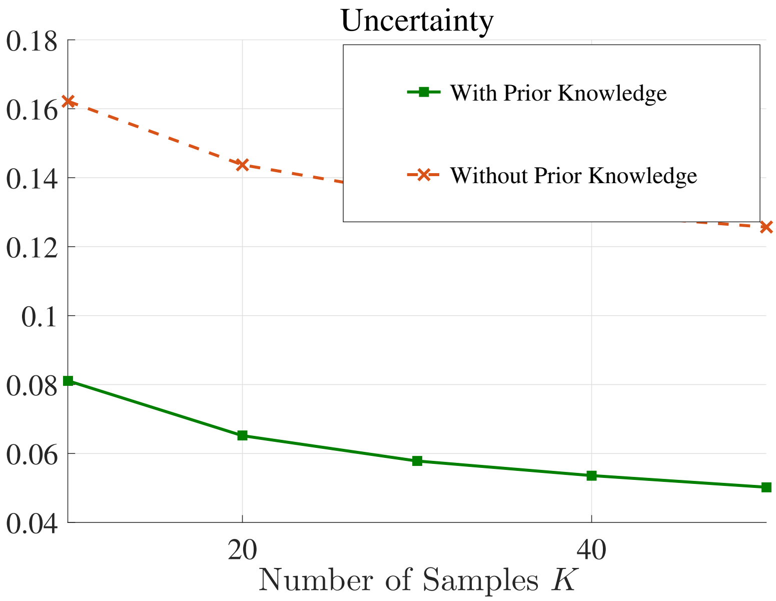


Figure 2.3: We compare the performance of our framework with the case where prior knowledge about the monotonicity of the cell-load has not been considered.

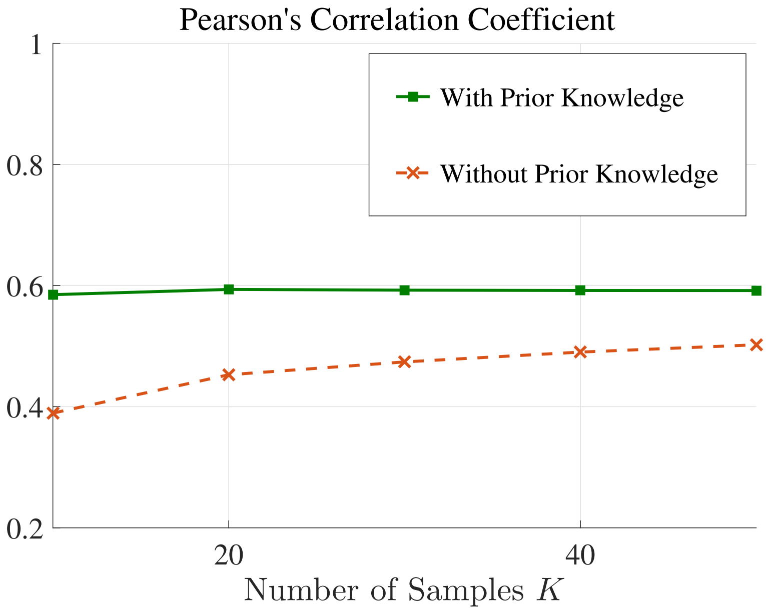


Figure 2.4: We compare the performance of LIMF learning framework with the case where prior knowledge about the monotonicity of the cell-load has not been considered.

We perform the comparison in terms of two metrics, namely the *magnitude of uncertainty* given as $\frac{|\sigma_\mathrm{u}(\mathbf{r})-\sigma_\mathrm{l}(\mathbf{r})|}{2}$ [see Section 2.4.1], where the rate $\mathbf{r}$ is a test sample point and $\sigma_\mathrm{u}(\mathbf{r})$ and $\sigma_\mathrm{l}(\mathbf{r})$ are upper and lower bounds, and the *correlation* with test sample set that we measure in terms of the popular *Pearson's correlation coefficient*.

The results are shown in Figure 2.3 and Figure 2.4. Figure 2.3 shows that uncertainty about the cell-load values decreases with the increasing training sample set size K in both cases. However, we observe that the prior information regarding the monotonicity always results in less uncertainty than the case where monotonicity of the cell-load is ignored. The results are therefore of a theoretical significance and they justify the inclusion of monotonicity as part of the prior knowledge in the framework [see Remark 2.1]. The same effect is seen in Figure 2.4 where we can clearly see that the case with all prior information included in the framework results in more correlation with the test sample set.

Comparison with State-of-Art Techniques

In this section we compare our learning framework with some low-complexity state-of-art techniques for various training sample and network sizes. Throughout this section, we estimate L from the available training sample set. We compare our method with four multivariate techniques, namely the state-of-art methods *Gaussian process regression* (GPR) and *ensemble learning with random forests* (ERF), and the simple *2-nearest neighbor interpolation*. These techniques are well-known for their ability to approximate continuous functions defined over compact sets well which is the case with the cell-load function. Note that, in addition to the state-of-art methods, it is important to compare the performance with a simple method such as the *2-nearest neighbor interpolation* to highlight the difficulty of learning with small sample sets. As we mentioned before, we consider very small sizes which rules out frameworks such as neural networks.

Figure 2.5 shows a comparison of (linear) *Pearson's* correlation coefficient, which is a popular measure of the strength and direction of the linear relationship between the predicted and the real test values, for an increasing sample size and fixed number of users $N = 30$. In particular, we use this coefficient as a measure of the "quality" of approximation. A high positive value of *Pearson's* correlation coefficient means that the predictions made by the learning method have a strong linear relationship with the test sample set. Figure 2.6 shows the maximum or worst-case error encountered while predicting on the test sample set for an increasing sample size K and fixed number of users $N = 30$. The maximum error is more suitable for comparing the robustness of the approximation techniques than some other popular error metrics because it shows that all error residuals remain below this level. Therefore, the maximum error is a reasonable substitute for the maximum error of approximation in (2.4) which we cannot compute directly.

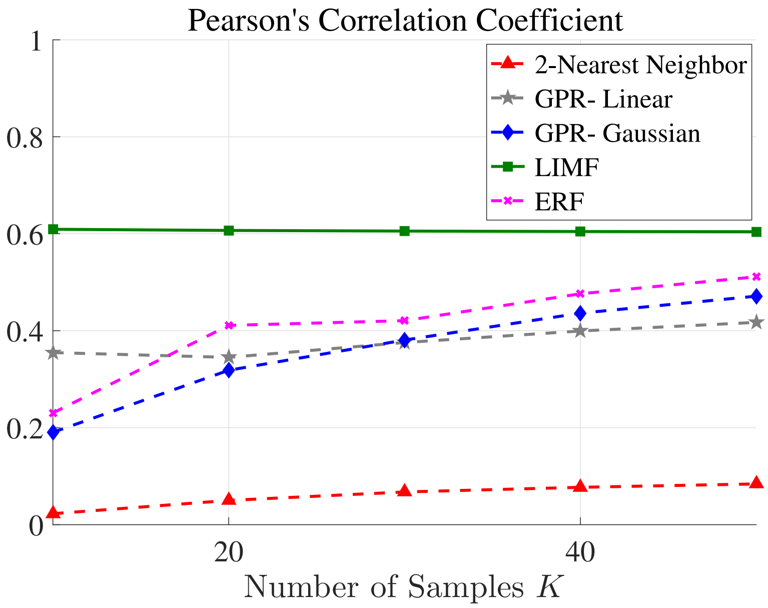


Figure 2.5: We compare the 5 techniques in terms of the linear correlation between predictions and true values for increasing K.

It is important to analyze maximum error and correlation together to better understand the comparison between our learning framework and other techniques. We observe that even for an inexact value of Lipschitz constant L, our method outperforms other techniques. An interesting observation is the fact that the GPR method (with the Gaussian function) and ERF show a relatively good error performance in Figure 2.6 but a considerably smaller correlation in Figure 2.5 than our method for small sampze sizes $K < 30$. This is because of the fact that our method incorporates prior knowledge about the cell-load and other methods do not. The poorest performance is seen in the case of the *2-nearest neighbor interpolation* whose performance improves slowly with increasing sample size. Clearly, this shows that we do not have enough samples to perform such a simple interpolation.

Finally, Figure 2.7 and Figure 2.8 show the effect of network size (in terms of number of users N) on the performance of all techniques for a small and fixed sample size of $K = 20$. We see that there is a gradual degradation of performance for all techniques but our method outperforms others. In particular, we observe in Figure 2.7 that the GPR with Gaussian function performs poorly due to insufficient training.

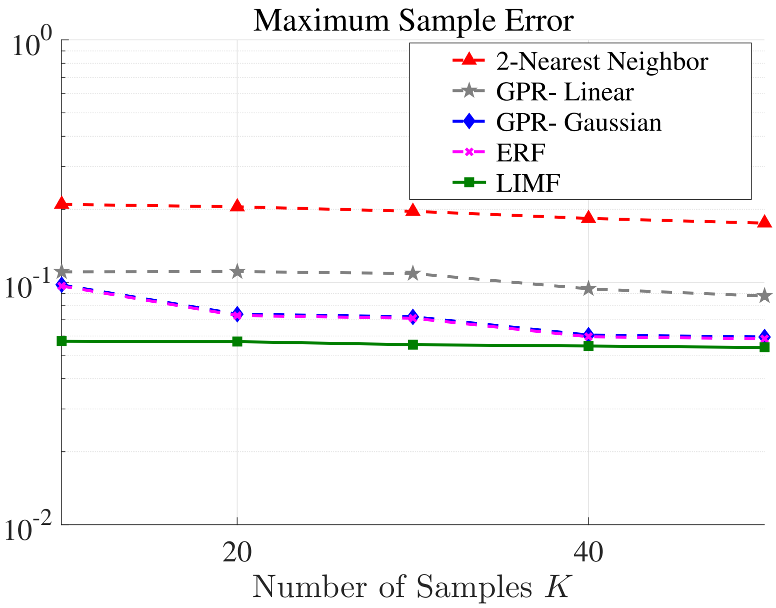


Figure 2.6: We compare the 5 techniques in terms of the maximum error between predictions and true values for increasing K.

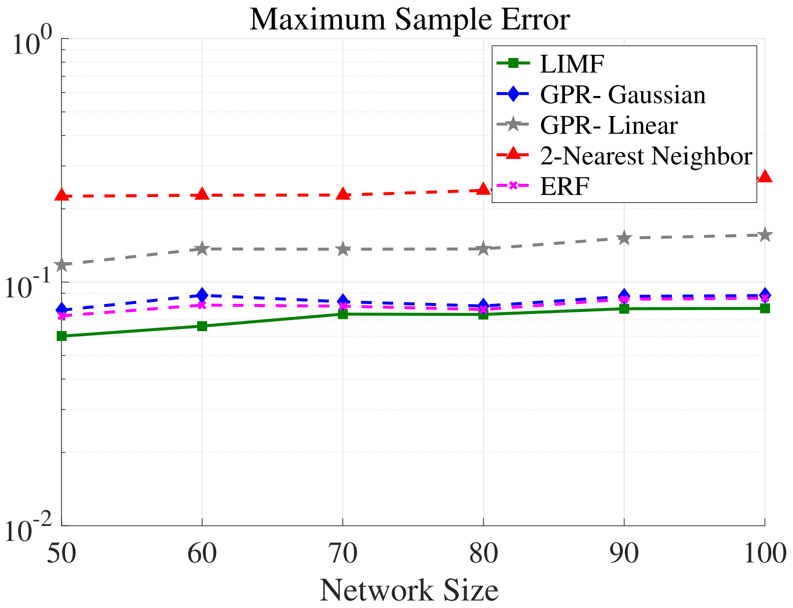


Figure 2.8: We compare the the 5 techniques in terms of the maximum error between predictions and true values for increasing network size.

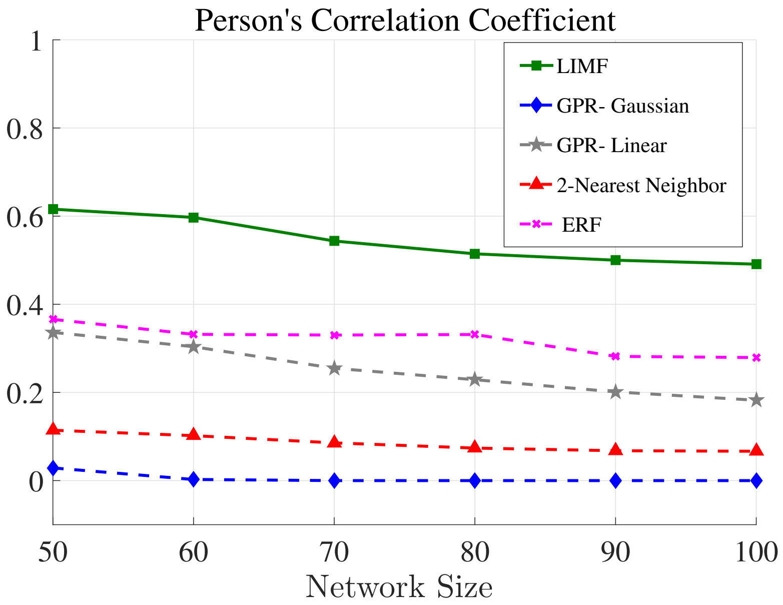


Figure 2.7: We compare the the 5 techniques in terms of the linear correlation between predictions and true values for increasing network size.

Table 2.3: Training Time Comparison on a standard PC

Technique	Average Training Time
LIMF	10×10^{-3} seconds
Nearest Neighbor	not applicable
GPR	80×10^{-3} seconds
ERF	60×10^{-3} seconds

2.7 Supplementary Material and Proofs

2.7.1 First Jacobian of $\mathbf{g}$

The entry $[\boldsymbol{\nabla}^{\mathbf{g}}_{\mathbf{r}}(\mathbf{r},\boldsymbol{\rho})]_{i,j}$ of the $M \times N$ Jacobian $\boldsymbol{\nabla}^{\mathbf{g}}_{\mathbf{r}}(\mathbf{r},\boldsymbol{\rho})$ is given by

$$[\boldsymbol{\nabla}^{\mathbf{g}}_{\mathbf{r}}(\mathbf{r},\boldsymbol{\rho})]_{i,j} = \begin{cases} -\frac{1}{RB\log(1+\gamma_{ij})}, & \text{if } j \in \mathcal{N}(i) \\ 0, & \text{otherwise} \end{cases}$$

where $\gamma_{ij} := \frac{p_i G_{i,j}}{\sum_{k\in\mathcal{M}\setminus\{i\}} p_k G_{k,j}\rho_k + \sigma^2}$.

2.7.2 Second Jacobian of $\mathbf{g}$

The entry $[\boldsymbol{\nabla}^{\mathbf{g}}_{\boldsymbol{\rho}}(\mathbf{r}, \boldsymbol{\rho})]_{i,k}$ of the $M \times M$ Jacobian $\boldsymbol{\nabla}^{\mathbf{g}}_{\boldsymbol{\rho}}(\mathbf{r}, \boldsymbol{\rho})$ is given by

$$[\boldsymbol{\nabla}^{\mathbf{g}}_{\boldsymbol{\rho}}(\mathbf{r}, \boldsymbol{\rho})]_{i,k} = \begin{cases} -\sum_{j \in \mathcal{N}(i)} \ln(2) \frac{r_j}{RB} \frac{\frac{p_i G_{i,j}}{p_k G_{k,j}}}{\ln^2(1+\gamma_{i,j})(\gamma_{i,j}^{-2} + \gamma_{i,j}^{-1})}, & \text{if } i \neq k \\ 1, & \text{if } i = k \end{cases}$$

where $\gamma_{ij} := \frac{p_i G_{i,j}}{\sum_{k \in \mathcal{M} \setminus \{i\}} p_k G_{k,j} \rho_k + \sigma^2}$.

2.7.3 Invertibility of the Jacobian

We follow the analysis in [RSF14] which exploits the sufficient conditions for invertibility of a generalized diagonal dominant matrix [BP94] on the whole domain. In more detail, we show that the matrix $\boldsymbol{\nabla}^{\mathbf{g}}_{\boldsymbol{\rho}}(\mathbf{r}, \boldsymbol{\rho})$ is invertible because it is an invertible generalized diagonal dominant matrix. For any $\boldsymbol{\rho} \in \mathbb{R}^M_{>0}$

$$[\boldsymbol{\nabla}^{\mathbf{g}}_{\boldsymbol{\rho}}(\mathbf{r}, \boldsymbol{\rho})]_i \boldsymbol{\rho} = \rho_i - \sum_{j \in \mathcal{N}(i)} \frac{r_j}{RB \log(1+\gamma_{i,j})} \times \frac{\frac{\sum_{k \in \mathcal{M} \setminus \{i\}} \rho_k p_k G_{k,j}}{p_i G_{i,j}}}{\ln(1+\gamma_{i,j})(\gamma_{i,j}^{-2} + \gamma_{i,j}^{-1})},$$

where $[\boldsymbol{\nabla}^{\mathbf{g}}_{\boldsymbol{\rho}}(\mathbf{r}, \boldsymbol{\rho})]_i$ is the ith row of $\boldsymbol{\nabla}^{\mathbf{g}}_{\boldsymbol{\rho}}(\mathbf{r}, \boldsymbol{\rho})$.

Now, since $\frac{\sum_{k \in \mathcal{M} \setminus \{i\}} \rho_k p_k G_{k,j}}{p_i G_{i,j}} < \frac{\sum_{k \in \mathcal{M} \setminus \{i\}} \rho_k p_k G_{k,j} + \sigma^2}{p_i G_{i,j}} = \gamma_{i,j}^{-1}$ and $\ln(1+\gamma_{i,j})(\gamma_{i,j}^{-2} + \gamma_{i,j}^{-1}) > \gamma_{i,j}^{-1}$ [RSF14], we have

$$\sum_{j \in \mathcal{N}(i)} \frac{r_j}{RB \log(1+\gamma_{i,j})} \times \frac{\frac{\sum_{k \in \mathcal{M} \setminus \{i\}} \rho_k p_k G_{k,j}}{p_i G_{i,j}}}{\ln(1+\gamma_{i,j})(\gamma_{i,j}^{-2} + \gamma_{i,j}^{-1})} < \sum_{j \in \mathcal{N}(i)} \frac{r_j}{RB \log(1+\gamma_{i,j})} = \rho_i$$

which implies that $[\boldsymbol{\nabla}^{\mathbf{g}}_{\boldsymbol{\rho}}(\mathbf{r}, \boldsymbol{\rho})]_i \boldsymbol{\rho} > 0$. Since the off-diagonal entries are all non-positive and diagonal entries are all non-negative, $\boldsymbol{\nabla}^{\mathbf{g}}_{\boldsymbol{\rho}}(\mathbf{r}, \boldsymbol{\rho})$ satisfies the sufficient conditions for it to be an invertible generalized diagonal dominant matrix [RSF14, BP94].

2.7.4 Equicontinuity of $\mathbf{L}$-Lipschitz functions

Lemma 2.2. *Let $\mathcal{F} \subset C(\mathcal{R}, \mathcal{L})$ denote the set of $\mathbf{L}$-Lipschitz functions with $\mathbf{L} := [L_1, L_2, \cdots, L_M]^\intercal \in \mathbb{R}^M_{\geq 0}$. The set $\mathcal{F}$ is an equicontinuous subset of $C(\mathcal{R}, \mathcal{L})$.*

Proof. Since each component of $\mathbf{f} \in \mathcal{F}$ is Lipschitz on $\mathcal{R} \subset \mathbb{R}^N_{>0}$, we have that

$$(\forall \mathbf{x}, \mathbf{y} \in \mathcal{R})\ (\forall i \in \overline{1, M})\ \ |f_i(\mathbf{x}) - f_i(\mathbf{y})| \leq L_i \|\mathbf{x} - \mathbf{y}\|\,.$$

Define $L_{\max} := \max_{i \in \overline{1,M}} L_i$ and note that

$$(\forall \mathbf{x}, \mathbf{y} \in \mathcal{R})\ \|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\|_\infty \leq L_{\max} \|\mathbf{x} - \mathbf{y}\|\,.$$

From the equivalence of norms in finite dimensional normed spaces it follows that $(\exists C > 0)$ such that

$$\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| \leq C\ \|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\|_\infty \leq C\ L_{\max} \|\mathbf{x} - \mathbf{y}\|\,. \tag{2.9}$$

Given $\epsilon > 0$ and for every $\mathbf{x}_o \in \mathcal{R}$, choose $\delta := \frac{\epsilon}{L_{\max} C}$ as the radius of $B_{\mathcal{R}}(\mathbf{x}_o, \delta)$. We have from (2.9) that

$$\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}_o)\| \leq C\, L_{\max} \|\mathbf{x} - \mathbf{x_o}\| < \epsilon, \tag{2.10}$$

whenever $\|\mathbf{x} - \mathbf{x_o}\| < \delta$. We have shown that δ can be chosen independently of $\mathbf{x}_o$. Now since (2.10) holds for every $\mathbf{f} \in \mathcal{F}$, the proof is complete. □

2.7.5 Closedness of $\mathcal{F}^{\text{mon}}$ and $\mathcal{F}^{\text{Lip}}$

Lemma 2.3. *Consider the space $C(\mathcal{X}, \mathcal{Y})$.*

a). The set of monotonic functions $\mathcal{F}^{mon}$ in $C(\mathcal{X}, \mathcal{Y})$ is closed.

b). The set of **L***-Lipschitz functions $\mathcal{F}^{Lip}$ in $C(\mathcal{X}, \mathcal{Y})$ is closed.*

Proof. a). Let $(\mathbf{f}_n)_{n\in\mathbb{N}} \subset \mathcal{F}^{\text{mon}} \subset C(\mathcal{R}, \mathcal{L})$ be an arbitrary convergent sequence of continuous monotone functions converging to some $\mathbf{g} \in C(\mathcal{R}, \mathcal{L})$. Then from Definition 1.2, and the fact that inequalities are preserved in the limit, it follows that:

$$\begin{aligned}
(\forall \mathbf{x}, \mathbf{y} \in \mathcal{R})\ \mathbf{x} \leq \mathbf{y} &\implies (\forall n \in \mathbb{N})\ \mathbf{f}_n(\mathbf{x}) \leq \mathbf{f}_n(\mathbf{y}) \\
(\forall \mathbf{x}, \mathbf{y} \in \mathcal{R})\ \mathbf{x} \leq \mathbf{y} &\implies \lim_{n\to\infty} \mathbf{f}_n(\mathbf{x}) \leq \lim_{n\to\infty} \mathbf{f}_n(\mathbf{y}) \\
(\forall \mathbf{x}, \mathbf{y} \in \mathcal{R})\ \mathbf{x} \leq \mathbf{y} &\implies \mathbf{g}(\mathbf{x}) \leq \mathbf{g}(\mathbf{y}),
\end{aligned}$$

which means that $\mathbf{g} \in \mathcal{F}^{\text{mon}}$. Since $(\mathbf{f}_n)_{n\in\mathbb{N}}$ was chosen arbitrarily, the above holds for every sequence in $\mathcal{F}^{\text{mon}}$ showing that $\mathcal{F}^{\text{mon}}$ is closed.

b). Following the same idea as above, we show that the limit function $\mathbf{g} \in C(\mathcal{R}, \mathcal{L})$ of an arbitrary sequence $(\mathbf{f}_n^{\text{Lip}})_{n\in\mathbb{N}} \subset \mathcal{F}^{\text{Lip}} \subset C(\mathcal{R}, \mathcal{L})$ is Lipschitz with the same

$\mathbf{L}$, i.e., $\mathbf{g} \in \mathcal{F}^{\mathrm{Lip}}$ also. Note that $\|\mathbf{f}_n^{\mathrm{Lip}} - \mathbf{g}\|_{C(\mathcal{R})} \to 0$ if and only if $(\forall i \in \overline{1, M})$ $\|f_{in}^{\mathrm{Lip}} - g_i\|_{C(\mathcal{R})} \to 0$. Therefore, it suffices to show that $(i \in \overline{1, M})$ g_i, the limit of the sequence $(f_{in}^{\mathrm{Lip}})_{n \in \mathbb{N}}$, is Lipschitz with L_i, the ith component of $\mathbf{L}$.

Now, since $f_{in}^{\mathrm{Lip}} \to f_i$ uniformly, for some $\epsilon > 0$ there exists $N_1^\epsilon \in \mathbb{N}$ such that $(\forall \mathbf{x} \in \mathcal{R})$ $|f_i(\mathbf{x}) - f_{iN_1^\epsilon}^{\mathrm{Lip}}(\mathbf{x})| < \epsilon$ which implies that there exists $N^\epsilon > N_1^\epsilon$ such that $(\forall \mathbf{x} \in \mathcal{R})$ $|f_i(\mathbf{x}) - f_{iN^\epsilon}^{\mathrm{Lip}}(\mathbf{x})| < \epsilon/2$. Then,

$$\begin{aligned}(\forall \mathbf{x} \in \mathcal{R})\ (\forall \mathbf{y} \in \mathcal{R})\ |f_i(\mathbf{x}) - f_i(\mathbf{y})| &= |f_i(\mathbf{x}) + f_{iN^\epsilon}^{\mathrm{Lip}}(\mathbf{x}) \\ &\quad - f_{iN^\epsilon}^{\mathrm{Lip}}(\mathbf{x}) + f_{iN^\epsilon}^{\mathrm{Lip}}(\mathbf{y}) \\ &\quad - f_{iN^\epsilon}^{\mathrm{Lip}}(\mathbf{y}) - f_i(\mathbf{y})| \\ &< \epsilon/2 + \epsilon/2 + L_i\|\mathbf{x} - \mathbf{y}\| \\ &= \epsilon + L_i\|\mathbf{x} - \mathbf{y}\|.\end{aligned}$$

Since the above holds for all $\epsilon > 0$, it follows that

$$(\forall \mathbf{x} \in \mathcal{R})\ (\forall \mathbf{y} \in \mathcal{R})\ |f_i(\mathbf{x}) - f_i(\mathbf{y})| \leq L_i\|\mathbf{x} - \mathbf{y}\|.$$

□

3 Robust Multiuser Detection

3.1 Introduction

Solutions to problems concerning multiuser communication in wireless systems are typically based on models that require, channel state information, knowledge of interference patterns, and information about the phase of desired users, to name a few examples. As a result, before data communication over the wireless channel, receivers traditionally estimate many model parameters. However, this approach has two major drawbacks that can severely impair the performance of communication systems. First, perfect estimation of user parameters is impossible in general because of noise, a limited number of training samples, and the complexity of the algorithms. Second, models themselves are only idealizations and they are based on simplifying assumptions about the wireless environment. It is often unclear how well they can capture the true behavior of real systems, especially in modern dynamic wireless networks.

To mitigate the above handicaps of model based receivers, recently machine learning based alternatives have been proposed. The idea is to replace some building blocks of conventional receivers by learning algorithms in order to reduce the number of assumptions required by the models and the complexity of estimation. However, the resulting reduction in model knowledge brings many technical challenges. In particular, some state-of-the-art learning tools, such as those based on neural networks, require large training sets and a long training time [Mar18]. However, in the physical layer, channels or their statistics can be considered roughly constant only for few milliseconds, which can be all the time available to collect training pilots, train a machine learning algorithm, and then perform the communication task. If this temporal aspect is not taken into account, then by the time enough samples are available to train existing state-of-the-art algorithms, the environment may have changed so drastically as to render the learning useless for current propagation conditions. As a result, learning techniques must work with small training sets and they have to deal with the uncertainty resulting from small training sets.

The above-mentioned uncertainty can be reduced by combining model based prior knowledge about the function (in this case a good multiuser receiver) to be approximated with knowledge obtained from a given training sample set (see e.g., [MIM$^+$18, ACS19, DRG17]). For example, it is known that the optimal multiuser detector (which minimizes the bit error rate (BER) of the desired user) is the nonlinear maximum a posterior (MAP) filter [Ver98, MP94, CHW04]. Ideally, in a dynamic environment we would like to approximate the MAP filter with a relatively small number of training samples employing, preferably, low-complexity learning techniques. However, in modern wireless networks, such as massive machine-type communications (mMTC), users transmit sporadically. In these systems, unlike linear filters, nonlinear receive filters suffer from the lack of robustness against sporadic interference [STY09]. In order to achieve a balance between performance and robustness, we develop online learning based partially linear receiver consisting of a nonlinear and a linear component, where the nonlinear component is an approximation of the maximum a posterior (MAP) filter. Before we discuss our contributions in detail, we discuss some related work in the next section.

3.1.1 Related Work: Machine Learning Receivers

It is known that the optimal multiuser detector (which minimizes the BER of the desired user) is the nonlinear maximum a posterior (MAP) filter [Ver98,MP94,CHW04]. Since the optimal MAP filter has a complexity that is exponential in the number of users, several studies have considered either suboptimal linear receivers or suboptimal machine learning based nonlinear receivers. We shall discuss three conventional receivers including the optimal MAP receiver in Section 3.2. Here, we briefly discuss nonlinear receivers that are based on nonlinear neural networks. The use of neural networks in multiuser communication goes back to [APO92], where the authors have designed a *multilayer perceptron* for detection in CDMA systems. The authors show that this neural network can match the performance of the optimal MAP filter. Neural networks based on radial basis function (RBF) networks have also been proposed based on the fact that the optimal MAP filter has an RBF structure [MP94, CHW04]. However, these techniques approximate the optimal MAP filter by estimating the parameters (e.g., the centers for the involved RBF functions) required for its implementation. For other techniques, involving neural networks [both deep and shallow networks] see [IN07] and references therein. To summarize, the main limitations of these studies is that a large number of samples are generally needed to obtain acceptable performance and, because these receivers are nonlinear, they are not robust against sporadic interference and other small variations in the environment.

Traditional offline learning techniques acquire a training sample set and they approximate the underlying true function by minimizing an empirical loss function defined over the training sample set. The quality of approximation depends on, of course, how well the approximation generalizes on unseen data. In online learning, which is more suited to real-time applications, the sample set is acquired in a sequential manner and therefore a loss function over the entire sample set cannot be defined. In this case, we need online adaptive algorithms that minimize time-changing and sequentially arriving loss functions. Adaptive learning algorithms in reproducing kernel Hilbert Spaces (RKHSs) have been applied to the multiuser detection problem in [TSY11, STY09]. These studies design a nonlinear receive filter in an infinite-dimensional RKHS using a criteria/loss function that is related to the BER of the desired user. The algorithm is an online version of the celebrated *Polyak's subgradient algorithm.* To achieve robustness against sporadic interference, the algorithm uses prior knowledge about the angles of arrival of the desired users. The authors show that this method outperforms a conventional linear method. No comparison with any nonlinear technique is presented, but the algorithm requires relatively small number of samples to achieve a good BER. In contrast, we do not assume any knowledge about user channels or angles of arrival because their estimation is prone to errors. .

In the remainder of this chapter, to align our terminology with that of the studies [STY09, TSY11], we will use the words “filter” and “function” interchangeably.

3.1.2 Contribution

We consider a challenging scenario in which the number of antennas available at the base station may be smaller than the number of active users. As an example, such a situation is likely to arise in the mMTC use-case in 5G and beyond networks. Our proposed method is an online partially linear receive filter that learns to detect symbols of a desired user without requiring any intermediate user parameter estimation (e.g., channel estimation), and it shows good robustness against sudden changes of the wireless environment. Our work is mainly inspired by the studies [STY09, TSY11] in which a nonlinear multiuser detection filter is designed in an infinite-dimensional RKHS. These studies show that working in certain RKHSs is particularly suited to real-time nonlinear adaptive filtering applications because the approximation can be carried out using low-complexity online algorithms. Moreover, the particular deterministic projection based learning approach tends to work well with a relatively small number of training samples. Since user parameter estimation is prone to errors, in contrast to [STY09, TSY11], rather than assuming the knowledge of the angles of arrival of the desired users, we provide our design with additional robustness by considering partially linear filters, as proposed in [Yuk15a] in a different application domain. In multiuser systems, such as power-domain NOMA systems, users of interest are multiplexed in the signal-to-noise ratio (SNR) domain. This means that the desired user set can be seen as having weak users, strong users, and intermediate users according to their respective SNRs. In general, if a strong user is to be detected, then a linear filter should suffice, while for the weak user we may need nonlinearity because this user suffers from excessive multi-access interference. Based on this intuition, we design a

Table 3.1: List of Variables

Description	Symbol
Number of Antennas	$M \in \mathbb{N}$
Number of users	$K \in \mathbb{N}$
Convex set at time $t \in \mathbb{N}$	C_t
Hilbert Space with inner-product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$	$(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$
Gaussian kernel with width $\sigma > 0$	$\kappa_{\mathrm{G}}^{\sigma}$
RKHS of Gaussian kernel with width $\sigma > 0$	$\mathcal{H}_{\mathrm{G}}^{\sigma}$
RKHS of Gaussian kernel with weight w_{G}	$\mathcal{H}_{\mathrm{G},w_{\mathrm{G}}}^{\sigma}$
Linear kernel	κ_{L}
RKHS of Linear kernel with weight w_{L}	$\mathcal{H}_{\mathrm{L},w_{\mathrm{L}}}$
Channel coherence time	$T_{\text{block}} \in \mathbb{N}$
Sample acquisition time	$T_{\text{train}} \in \mathbb{N}$

partially linear filter which consists of weighted linear and nonlinear components. Note that, generally, the more nonlinear a receiver becomes, the less robust such a receiver is against sporadic interference. Therefore, for strong users highly nonlinear filters may be an overkill. In our framework the nonlinearity of the aggregate filter can, in priciple, be adapted according to the situation of the desired user (also see [**?**] which has demonstrated our design in a hardware-in-a-loop system), e.g., the SNR or BER performance of the desired user.

In Section 3.2 we present the multiuser detection system model and we look at various conventional detection techniques including the optimal MAP filter. In Section 3.3 we design the above mentioned nonlinear component of the aggregate filter. To add model based knowledge to our design, we show how to adaptively learn a nonlinear filter that mimics the optimal MAP filter using a low-complexity adaptive learning algorithm. Previous studies have shown that the optimal MAP filter has a form of a Gaussian RBF network. We show that the optimal MAP filter belongs to certain RKHSs. The significance of this fact is that it allows us to use a low-complexity algorithm to obtain a good approximation of the above-mentioned nonlinear filter, using a relatively small training sample set, directly without any intermediate user parameter estimation. In Section 3.4 we present our aggregate (partially linear) filter design with weighted linear and nonlinear components by extending the RKHS of nonlinear functions from Section 3.3 to now also include linear functions. We also show how the complexity and memory requirements of the aggregate filter can be reduced to make it suitable for real-time online applications. In Section 3.5 we simulate various aspects of the filter design and we compare the performance with some conventional techniques.

3.2 Multiuser Detection

Consider the multiuser uplink shown in Figure 3.9, where a base station with $M \in \mathbb{N}$ antennas receives signals from $K \in \mathbb{N}$ simultaneously transmitting single-antenna users. We assume that the received baseband signal (with symbol-rate sampling) at the base station at (symbol) time $t \in \mathbb{N}$ is given by [see, e.g., [CHW04, TV05]]

$$\mathbf{r}(t) := \sum_{k=1}^{K} \sqrt{p_k(t)} b_k(t) \mathbf{h}_k(t) + \mathbf{n}(t) \in \mathbb{C}^M, \tag{3.1}$$

where $p_k(t) \in]0\ \infty[$, $b_k(t) \in \mathbb{C}$, and $\mathbf{h}_k(t) \in \mathbb{C}^M$ are the power, the modulation symbol, and the channel, respectively, for user $k \in \overline{1, K}$, and where $\mathbf{n}(t) \in \mathbb{C}^M$ denotes additive noise. Traditionally, one assumes that the noise is additive white Gaussian noise (AWGN) with zero mean and variance $\mathbb{E}[\mathbf{n}(t)\mathbf{n}(t)^{\mathsf{H}}] = 2\sigma_n^2\mathbf{I}$ with $\mathbf{I}$ the $M \times M$ identity matrix.

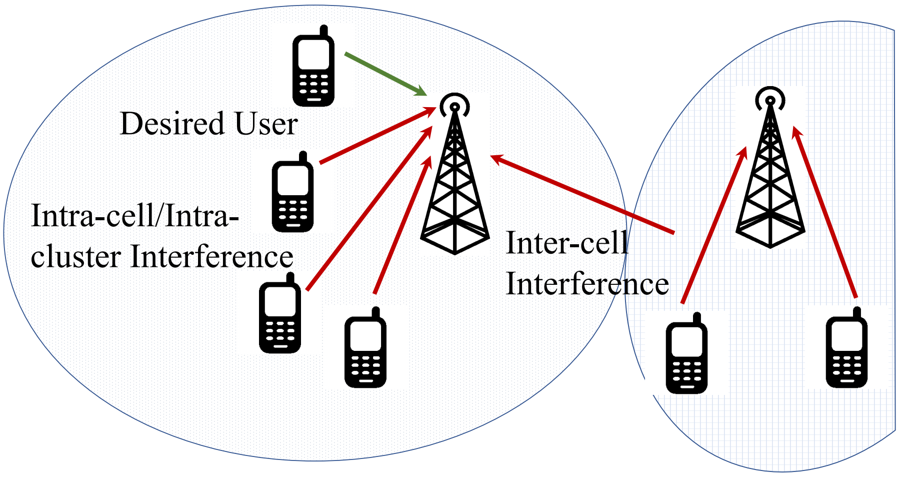


Figure 3.1: Multiuser uplink: The received base band signal $\mathbf{r}(t)$ consists of the desired signal and noise plus interference from other users in the same cell and also users from other cells.

3.2.1 Optimal Multiuser Detection

Without loss of generality, assume in the following that the multiuser receiver at the base station wants to detect the data of user $k = 1$. For simplicity, we make the common assumption that $(\forall t \in \mathbb{N})$ $(\forall k \in \overline{1, K})$ $\mathbf{h}_k(t) := \mathbf{h}_k$ and $p_k(t) := p_k$, i.e., the channels and powers of users remain constant. Let $\mathbf{P} := [\sqrt{p_1}\mathbf{h}_1, \sqrt{p_2}\mathbf{h}_2, \cdots, \sqrt{p_K}\mathbf{h}_K] \in \mathbb{C}^{M \times K}$ and $\mathbf{b}(t) := [b_1(t), b_2(t), \cdots, b_K(t)]^{\intercal} \in \mathbb{C}^K$. Then, we can rewrite (3.1) as [see the *matrix* *channel model* [Hon08, Section 1.2]]

$$\begin{aligned}\mathbf{r}(t) &= \mathbf{P}\mathbf{b}(t) + \mathbf{n}(t).\\ &:= \bar{\mathbf{r}}(t) + \mathbf{n}(t),\end{aligned} \tag{3.2}$$

where $\bar{\mathbf{r}}(t) := \mathbf{P}\mathbf{b}(t)$ is the noiseless multiuser signal. In the following, for simplicity, we assume that the system modulation scheme is BPSK, i.e., $(\forall k \in \overline{1, K})\ b_k(t) = \pm 1$.[1]

Note that $\bar{\mathbf{r}}(t)$ in (3.2) belongs to a finite set. In more detail, since $(\forall k \in \overline{1, K})\ b_k(t) \in \{+1, -1\}$, the cardinality of the finite set from which $\mathbf{b}(t) = [b_1(t), b_2(t), \cdots, b_K(t)]^\intercal$ takes its values is $N_{\text{mod}} := 2^K$. Let $\mathbf{b}_q$ denote the qth possible value of $\mathbf{b}(t)$, i.e., $(\forall t \in \mathbb{N})$ $(\exists q \in \overline{1, N_{\text{mod}}})\ \mathbf{b}_q = \mathbf{b}(t)$, then $\bar{\mathbf{r}}(t)$ belongs to the finite set

$$\tilde{\mathcal{X}} := \{\bar{\mathbf{r}}_q = \mathbf{P}\mathbf{b}_q,\ 1 \leq q \leq N_{\text{mod}}\} \subset \mathbb{C}^M,\ \operatorname{card}(\tilde{\mathcal{X}}) = N_{\text{mod}}. \tag{3.3}$$

Given a received signal $\mathbf{r}(t)$, the (single-user) optimal multiuser receiver performs the MAP decision based detection of $b_1(t)$. This detection is optimal in the sense of minimizing the BER of user 1. The MAP receiver consists of the MAP receive filter followed by a hard-decision. In more detail, the MAP receive filter $f^\star$ is given by [see, e.g., [Ver98, CHW04]]

$$(\forall t \in \mathbb{N})\ f^\star(\mathbf{r}(t)) := \sum_{q=1}^{N_{\text{mod}}} v_q \exp\left(\frac{\|\mathbf{r}(t) - \bar{\mathbf{r}}_q\|^2}{2\sigma_n^2}\right), \tag{3.4}$$

where

$$(\forall q \in \overline{1, N_{\text{mod}}})\ v_q = \frac{\operatorname{sgn}(b_{q_1})}{N_{\text{mod}}(2\pi\sigma_n^2)^M}, \tag{3.5}$$

and where b_{q_1} is the component of $\mathbf{b}_q$ corresponding to user 1. The hard-decision is given by

$$(\forall t \in \mathbb{N})\ \tilde{b}_1(t) := \begin{cases} +1, & f^\star(\mathbf{r}(t)) \geq 0,\\ -1, & f^\star(\mathbf{r}(t)) < 0.\end{cases} \tag{3.6}$$

3.2.2 Suboptimal Multiuser Detection

In the previous section we discussed the optimal MAP filter that achieves the best performance for each user in terms of the BER under AWGN assumption. However, the optimal MAP filter requires the knowledge of the center set $\tilde{\mathcal{X}}$ in (3.3). Obviously, this exact knowledge can be replaced by an estimation to obtain suboptimal performance. However, for a large number of users the complexity of this receiver is impractical because

[1] The modulation symbols (and the centers in (3.3)) are assumed to be equiprobable.

the cardinality of $\tilde{\mathcal{X}}$ increase exponentially with number of users K. Therefore, in the following we discuss some suboptimal receivers that are commonly considered instead of the optimal MAP filter. We once again assume AWGN and equiprobable BPSK signaling in the following.

Minimum Mean-Squared Error (MMSE) Receiver

The (linear) minimum mean-squared error (MMSE) filter is a widely used suboptimal receiver due to its relatively low complexity. The MMSE receiver consists of an MMSE filter followed by the hard-decision in (3.6). A common assumption when using the MMSE filter is that $M > K$, i.e., the number of receive antennas exceeds the number of users, and that there exists sufficient disparity among user channels such that they can be linearly separated.

With the received signal defined in (3.2), the linear MMSE filter given by [CHW04,TV05]

$$f^{\text{MMSE}}(\mathbf{r}(t)) := \mathbf{w}^{\mathsf{H}}\mathbf{r}(t) = \mathbf{w}^{\mathsf{H}}\bar{\mathbf{r}}(t) + \mathbf{w}^{\mathsf{H}}\mathbf{n}(t), \tag{3.7}$$

where

$$\mathbf{w} = (\mathbf{P}\mathbf{P}^{\mathsf{H}} + 2\sigma_n^2\mathbf{I})^{-1}\ \mathbf{p}_1\ \in \mathbb{C}^M, \tag{3.8}$$

with $\mathbf{p}_1 \in \mathbb{C}^M$ the first column of $\mathbf{P}$ in (3.2), minimizes the mean squared error

$$\mathbb{E}\left[|b_1(t) - f^{\text{mmse}}(\mathbf{r}(t))|^2\right].$$

Note that, given the signal-plus-noise covariance matrix $\mathbf{P}\mathbf{P}^{\mathsf{H}} + 2\sigma_n^2\mathbf{I}$ and $\mathbf{p}_1$, the MMSE filtering is simply a matrix inversion and multiplication operation. In practice, the matrix $\mathbf{P}\mathbf{P}^{\mathsf{H}} + 2\sigma_n^2\mathbf{I}$ is replaced by its empirical estimate obtained by using training pilots.

Remark 3.1 (Robustness of Linear Filters). It can be easily seen by (3.7) that (due to the linearity of the MMSE filtering) if a user $k \neq 1$ leaves the system, the SINR of user $k = 1$ improves. In other words, traditional linear receive filters are robust against sporadically transmitting interference sources. However, this does not hold in general for the nonlinear optimal MAP filter. In fact, all nonlinear receivers, besides being generally more complex than linear receivers, lack the robustness of their linear counterparts in dynamic environments.

Successive Interference Cancellation (SIC) Receiver

In contrast to the MMSE receiver, the MMSE with (symbol-level) successive interference cancellation (SIC) receiver (also known as the MMSE-SIC receiver) performs joint successive detection of all users in the decreasing order of their received SNRs.

In more detail, the users are ordered according to their received SNRs. The SIC procedure is illustrated in Figure 3.2 for up to the first 3 users in the SIC order. The modulation symbol of the user with the best SNR is detected first using the MMSE filter on the received signal $\mathbf{r}(t)$ in (3.1), and by treating interference from other users as noise. For every subsequent user in the SIC order, the interference contributions from all previous users are subtracted from the received signal. Then the MMSE filtering is performed on the residual received signal by treating the interference from remaining users as noise.

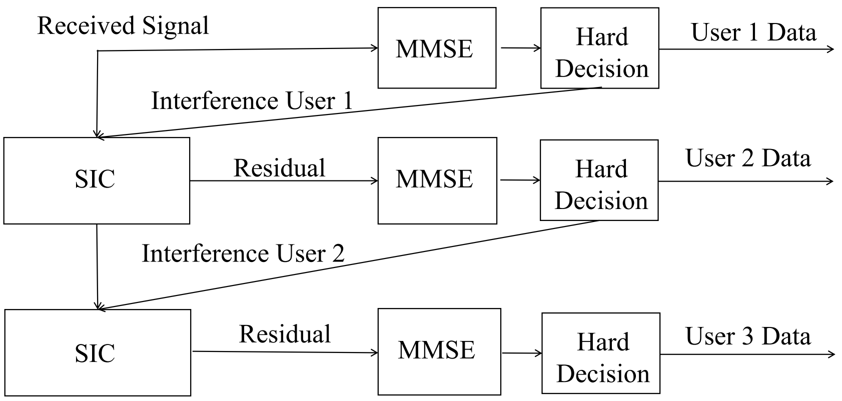


Figure 3.2: SIC receiver: The received base band signal $\mathbf{r}(t)$ consists of the desired signal and noise plus interference from other users. Here we show the MMSE-SIC procedure for the first 3 users. For the second user, the residual signal is given by $\mathbf{r}(t) - \sqrt{p_1} b_1(t)\mathbf{h}_1$, and for the third user the residual is given by $\mathbf{r}(t) - \sqrt{p_1} b_1(t)\mathbf{h}_1 - \sqrt{p_2} b_2(t)\mathbf{h}_2$.

Note that we would expect the nonlinear MMSE-SIC to outperform the MMSE receiver as long as there exists sufficient disparity among user channels resulting in perfect SIC. However, the MMSE-SIC receiver is clearly more complex than the MMSE receiver and it also suffers from the lack of robustness [see Remark 3.1] in dynamic environments.

Remark 3.2. (Parameter Estimation) In literature, various methods have been developed to obtain estimations to the signal-plus-noise covariance matrix $\mathbf{P}\mathbf{P}^H + 2\sigma_n^2\mathbf{I}$ and the user channel matrix $\mathbf{P}$ in (3.8). Before, actual data communication can begin, conventional receivers carry out estimation of required parameters with the help of training pilots. In this sense, the actual detection of $b_1(t)$ is "indirect" and it suffers from errors in parameter estimation.

3.2.3 An Online Learning Receiver

In this section we briefly present the general concept of the online learning based multiuser detection [see, e.g., [STY09, ALCYS18, ACS19]]. This receiver is the focus of this chapter and the details are provided in Sections 3.3 and 3.4.

Without any loss of generality, suppose that the receiver is interested in the modulation data $(b_1(t))_{t\in\mathbb{N}}$ of user $k = 1$ in (3.1). As common in literature, we further assume that the channels between the users and the base station undergo Rayleigh block fading [DV17]. Under this assumption, channels remain constant for a block of complex channel symbols known as the coherence block. More precisely, let $t_b \in \mathbb{N}$ denote the start of the coherence block $b \in \mathbb{N}$, where $|t_b - t_{b+1}| := T_{\text{block}}$ is the coherence block size. Then, $(\forall k \in \overline{1, K})$ $(\forall t \in \overline{t_b, t_{b+1} - 1})(\exists \mathbf{h}_k^b \in \mathbb{C}^M)$ $\mathbf{h}_k(t) = \mathbf{h}_k^b$; i.e., $\mathbf{h}_k^b$ is the fixed channel of user k for the coherence block b, which lasts from time $t = t_b$ to time $t = t_{b+1} - 1$. In conventional receivers, $(b_1(t))_{t\in\mathbb{N}}$ is detected "indirectly" in the sense that the receivers first perform parameter estimation (user channels, SIC order, signal-plus-noise covariance matrix etc.) before data communication is carried out. This parameter estimation is carried out by using training pilots and therefore it is prone to errors. In contrast, the goal here is to learn to detect $(b_1(t))_{t\in\mathbb{N}}$ from the received signals $(\mathbf{r}(t))_{t\in\mathbb{N}}$ directly without any intermediate parameter estimation.

For clarity, assume in the following discussion that the modulation scheme is BPSK, i.e., $(\forall t \in \mathbb{N})$ $b_1(t) \in \{+1, -1\}$ [note: the ideas can be extended to higher modulation schemes as shown in Section 3.4]. In mathematical terms, the algorithm should ideally approximate a function $g^\star : \mathbb{C}^M \to \mathbb{R}$ such that [STY09, ALCYS18, ACS19]

$$(\forall t \in \mathbb{N})\ g^\star(\mathbf{r}(t)) = b_1(t). \tag{3.9}$$

Obviously, the objective is too optimistic because there is always noise at the receiver. To include the effect of noise, akin to the classical *bounded error estimation* techniques [see Section 1.7], the optimization goal is relaxed by introducing a noise-tolerance parameter $\epsilon > 0$. The optimization goal now becomes to find $g : \mathbb{C}^M \to \mathbb{R}$ such that

$$(\forall t \in \mathbb{N})\ \ |g(\mathbf{r}(t)) - b_1(t)| \leq \epsilon. \tag{3.10}$$

To this end, at the start of each coherence block b, the desired user sends $T_{\text{train}} < T_{\text{block}}$ training symbols $(b_1(t))_{t\in\overline{t_b, t_b + T_{\text{train}} - 1}}$, which are also known to the base station. With the corresponding received signals $(\mathbf{r}(t))_{t\in\overline{t_b,t_b+T_{\text{train}}-1}}$, a training set

$$\mathcal{S} := \left\{(\mathbf{r}(t), b_1(t))\ ,\ t \in \overline{t_b, t_b + T_{\text{train}} - 1}\right\}$$

is constructed that is used to approximate the function g in (3.10). Denoting the approximation by f we use $(f(\mathbf{r}(t))_{t\in\overline{t_b+T_{\text{train}},t_{b+1}-1}}$ as the estimates of the information symbols $(b_1(t))_{t\in\overline{t_b+T_{\text{train}},t_{b+1}-1}}$ in the coherence block b. In the remainder we consider a single coherence block starting at $t = t_b = 1$.

Remark 3.3 (Online Learning). Unlike traditional batch learning methods, the online method considered in this chapter improves the estimates of the ideal function g as soon as a training sample is obtained; i.e., it does not wait for the acquisition of the whole $\mathcal{S}$ to start the learning process. By doing so, detection of information symbols can start almost immediately after the last training sample of the set $\mathcal{S}$ becomes available, which is an important feature in high data rate systems.

We now briefly outline our online learning goal before moving on to the details in Sections 3.3 and 3.4. First, we assume that $g \in \mathcal{H}$, where $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ is a suitable RKHS associated with the kernel $\kappa : \mathbb{C}^M \times \mathbb{C}^M \to \mathbb{R}$ [see Section (1.5)]. We refer to $\mathcal{H}$ as the *problem element set* because we restrict our search for g satisfying (3.10) to this set. Using the reproducing property of $\mathcal{H}$, we can rewrite (3.10) as

$$(\forall t \in \mathbb{N})\ C_t := |\langle g, \kappa(\cdot, \mathbf{r}(t))\rangle_{\mathcal{H}} - b_1(t)| \leq \epsilon, \tag{3.11}$$

which is a closed convex set of functions in $\mathcal{H}$ [STY09]. An appropriate and computationally convenient objective is to find a function $g^\star \in \bigcap_{t\geq t_o} C_t$, for some $t_o \in \mathbb{N}$, as considered in [TSY11, STY09], which satisfies (3.11) for every $t \geq t_o$ under the assumption that $\bigcap_{t\geq t_o} C_t \neq \emptyset$. In other words, $g^\star$ belongs to all (infinite in number) but a finite number of sets C_t in (3.11). Note that, $\bigcap_{t\geq t_o} C_t \subset \mathcal{H}$ can also be seen as the *feasible solution set* in the context of convex feasibility and set theoretic problems [BB96, Com93].

Before we close out this section, Observation 3.1 shows that our online framework exhibits some robustness properties and that there is a well-defined notion of optimization attached to our framework. Furthermore, this connection enables us to use a numerically robust online learning algorithm.

Observation 3.1 (Connection with Robust Support Vector Regression (SVR)). *To align our work with conventional loss function based learning, we note that if* $(\forall t \in \mathbb{N})\ g^\star \in C_t$ *(i.e.,* $g^\star \in \bigcap_{t\in\mathbb{N}} C_t$*) then [STY09, The15]*

$$(\forall t \in \mathbb{N})\ g^\star \in \operatorname*{arg\,min}_{f\in\mathcal{H}}\ \mathcal{L}_t(f(\mathbf{r}(t)), b_1(t)) := \max\left\{0, |\langle f, \kappa(\cdot, \mathbf{r}(t))\rangle_{\mathcal{H}} - b_1(t)| - \epsilon\right\},$$

where $\mathcal{L}_t : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is the well-known linear ϵ-insensitive loss function used in the canonical form of robust support vector regression (SVR) [Vap95, SS04]. It is well-known that function approximation with this loss function is robust against outliers and uncertainty about the underlying noise distribution. From this perspective, our online framework can be seen as an online version of unregularized SVR [the loss functions $\mathcal{L}_t$ are infinite in number and they arrive in a sequential manner in contrast to SVR]. Moreover, since $\mathcal{L}_t$ are continuous and convex, we can use a numerically robust online learning algorithm to approximate $g^\star \in \bigcap_{t \geq t_o} C_t$ for some $t_o \in \mathbb{N}$.

3.3 Nonlinear Adaptive Multiuser Detection

In this section we design the nonlinear component of the partially nonlinear filter introduced in Section 3.1.2. For simplcity and to avoid technical digressions, we assume once again that the receiver is interested in the modulation data of user 1 and that the modulation scheme is BPSK. Recall from Section 3.2.3 that our objective is a good approximation of the "ideal" function in (3.9). However due to noise in the system, we relax the objective to (3.10) which, roughly speaking, keeps the filter output close to the desired $b_1(t)$. Now, let us fix $\epsilon < 1$ in (3.11) and note that if we find some function $g \in \mathcal{H}$ satisfying (3.11), the hard-decision (3.6) will determine $b_1(t)$ correctly for every $t \in \mathbb{N}$. This means that the BER in this case shall be 0 and g is optimal in the sense of minimizing the BER.

We start by assuming that the unknown noise sequence $(\mathbf{n}(t))_{t\in\mathbb{N}}$ in (3.1) is bounded, which makes our approximation framework a particular case of bounded error estimation presented in Section 1.7. Formally, we make the following assumption:

Assumption 3.1 (Bounded Noise)**.** *The unknown noise sequence* $(\mathbf{n}(t))_{t\in\mathbb{N}}$ *in* (3.1) *satisfies*

$$(\exists W_{noise} \in \mathbb{R})\ (\forall t \in \mathbb{N})\ \|\mathbf{n}(t)\| \leq W_{noise}. \tag{3.12}$$

Note that Assumption 3.1 in fact always holds in practical systems involving measurements with electrical circuits for obvious reasons. Under Assumption 3.1, the sequence $(\mathbf{n}(t))_{t\in\mathbb{N}}$ may be assumed to be sampled from a bounded distribution, e.g., a truncated Gaussian distribution, which is a particularly useful assumption in cases where the measurements are bounded (see [CK94] for a detailed analysis). In particular, note that if one assumes that $(\mathbf{n}(t))_{t\in\mathbb{N}}$ is sampled from an AWGN (which is a typical assumption in wireless communications), and one discards measurements that do not satisfy (3.12) for a fixed W_{noise}, then the remaining measurements can be seen as being sampled from a truncated Gaussian distribution.[2]

[2] Note that, for a sufficiently large value of the truncation W_{noise}, the truncated Gaussian distribution can approximate the AWGN reasonably well.

In the following, define by $\mathcal{X} \subset \mathbb{C}^M$ the *input space* of received vectors in (3.1). Note that we have not yet specified the RKHS $\mathcal{H}$ to which a good filter g satisfying (3.11) belongs. The choice of an appropriate $\mathcal{H}$ is crucial, and it may depend on many things, such as any prior knowledge we may have about g or the "richness" of $\mathcal{H}$ (note: a rich space may ensure the existence of g satisfying (3.11)). For example, if we choose $\mathcal{H}$ to be the space of all linear functions, then the conventional linear MMSE filter f^{MMSE}, which minimizes the mean squared error between the filter response and $b_1(t)$, belongs to $\mathcal{H}$. However, if $M < K$ or there does not exist sufficient disparity among user channels, then there may not exist $f^{\mathrm{MMSE}} \in \mathcal{H}$ satisfying (3.11) for a sufficiently small ϵ. The reason is that in this case user signals may become linearly inseparable in $\mathcal{H}$. It has been observed that, even in the linearly separable scenarios, a better choice for $\mathcal{H}$ is a space of nonlinear functions [CHW04]. Among the candidates for a suitable nonlinear filter, the nonlinear MAP filter defined in (3.4) is a natural choice. The reason is that even though the optimal MAP filter works with the AWGN assumption, it should also work well with truncated Gaussian noise based on the discussion above. In the following, we do not make this distinction explicitly to avoid notational clutter, and we continue to refer to $f^\star$ defined in (3.4) as the optimal MAP filter. As a result, we may assume that, for given a sequence $(\mathbf{r}(t))_{t\in\mathbb{N}} \subset \mathcal{X}$ in (3.1), the goal of the proposed algorithm is to find a $g^\star$ such that

$$(\forall t \in \mathbb{N})\ g^\star(\mathbf{r}(t)) \approx f^\star(\mathbf{r}(t)), \tag{3.13}$$

where $f^\star$ is the nonlinear MAP filter given by the expression in (3.4). We refer to $g^\star$ as the proxy optimal filter in the sequel, and we implicitly assume that such a $g^\star$ satisfies (3.11).

In light of the above, an appropriate choice of $\mathcal{H}$ is the space to which $f^\star$ belongs. Adding this model based prior knowledge has several advantages. In particular, one can show that $\mathcal{H}$ can be an RKHS that exhibits very attractive approximation and computational properties. To this end, in Section 3.3.1 we present our results regarding the RKHS $\mathcal{H}$ to which $f^\star$ belongs. In Section 3.3.2 we show that $f^\star$ belongs to an intersection of certain closed convex sets in $\mathcal{H}$. Utilizing these results, in Section 3.3.3 we study how a proxy optimal filter can be approximated by using the online learning rationale presented in Section 3.2.3. Because purely nonlinear filters are not robust in a dynamic wireless environment, we show in Section 3.4 how to extend the space $\mathcal{H}$ to also include linear functions, which adds robustness to the aggregate filter.

3.3.1 A New Look at the MAP Filter

The MAP filter $f^\star : \mathcal{X} \to \mathbb{R}$ in (3.4), defined over $\mathcal{X}$, is given by

$$(\forall \mathbf{x} \in \mathcal{X})\ f^\star(\mathbf{x}) := \sum_{q=1}^{N_{\text{mod}}} v_q \exp\left(-\frac{\|\mathbf{x} - \bar{\mathbf{r}}_q\|^2}{2\sigma_n^2}\right), \tag{3.14}$$

where the centers $\bar{\mathbf{r}}_q \in \tilde{\mathcal{X}} \subset \mathcal{X}$ and the constants v_q are given in (3.3) and (3.5), respectively.

Observation 3.2 (Required Knowledge for the MAP filter)**.** *The exact representation of $f^\star$ requires:*

1. *the knowledge of the set*

$$\tilde{\mathcal{X}} := \{\bar{\mathbf{r}}_q,\ 1 \le q \le N_{mod}\} \subset \mathbb{C}^M,\ card(\tilde{\mathcal{X}}) = N_{mod}, \tag{3.15}$$

which contains the centers $\bar{\mathbf{r}}_q$ in (3.14)*;*

2. *the knowledge of the noise parameter σ_n^2.*

Observation 3.2 shows that approximation of $f^\star$ requires an estimation of the set $\tilde{\mathcal{X}}$ which is complex for a large number of users. In contrast, our results in the following show that the approximation of $f^\star$ can be carried out using low-complexity algorithms in the RKHS to which $f^\star$ belongs.

The MAP Filter and RKHSs

We now present our results regarding the function spaces to which the MAP filter belongs. In particular, we show that it belongs to certain RKHSs. Note that by defining $\kappa_{\text{G}}^{\sigma_n}(\cdot,\cdot) := \exp\left(-\frac{\|\cdot-\cdot\|^2}{2\sigma_n^2}\right)$), we can write $f^\star$ in (3.15) as $f^\star = \sum_{q=1}^{N_{\text{mod}}} v_q\ \kappa_{\text{G}}^{\sigma_n}(\cdot, \bar{\mathbf{r}}_q)$.

Lemma 3.1. *Suppose the input space $Int(\mathcal{X}) \neq \emptyset$, i.e., $\mathcal{X}$ has nonempty interior. Then, the optimal MAP filter $f^\star = \sum_{q=1}^{N_{mod}} v_q\ \kappa_{\text{G}}^{\sigma_n}(\cdot, \bar{\mathbf{r}}_q)$ belongs to $\mathcal{H}_{\text{G}}^{\sigma}$ (i.e., the RKHS associated with the Gaussian kernel* $(\forall \mathbf{u} \in \mathcal{X})$ $(\forall \mathbf{v} \in \mathcal{X})$ $\kappa_{\text{G}}^{\sigma}(\mathbf{u}, \mathbf{v}) := \exp\left(-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2}\right)$*), if* $0 < \sigma < \sqrt{2}\sigma_n$*.*

Proof. Since $f^\star = \sum_{q=1}^{N_{\text{mod}}} v_q\ \kappa_{\text{G}}^{\sigma_n}(\cdot, \bar{\mathbf{r}}_q)$, it suffices to show that $(\forall q \in \overline{1, N_{\text{mod}}})$ $(\exists \bar{\mathbf{r}}_q \in \tilde{\mathcal{X}} \subset \mathcal{X})$ $\kappa_{\text{G}}^{\sigma_n}(\cdot, \bar{\mathbf{r}}_q) \in \mathcal{H}_{\text{G}}^{\sigma}$ because $\mathcal{H}_{\text{G}}^{\sigma}$ is a vector space. We have $(\forall \mathbf{r} \in \mathcal{X})$ $\exp\left(-u\frac{\|\cdot-\mathbf{r}\|^2}{2\sigma^2}\right) \in \mathcal{H}_{\text{G}}^{\sigma}$ if and only if $0 < u < 2$ [Min10, Theorem 3]. So by letting $\sigma_n := \frac{\sigma}{\sqrt{u}}$ we get the desired result that σ must satisfy $0 < \sigma < \sqrt{2}\sigma_n$. □

Note that according to Lemma 3.1, $f^\star \in \mathcal{H}_{\mathrm{G}}^{\sigma_n}$ has a representation as a finite sum of Gaussian kernels in $\mathcal{H}_{\mathrm{G}}^{\sigma_n}$, i.e.,

$$(\forall \mathbf{x} \in \mathcal{X})\ f^\star(\mathbf{x}) = \sum_{q=1}^{N_{\mathrm{mod}}} v_q\ \kappa_{\mathrm{G}}^{\sigma_n}(\mathbf{x}, \bar{\mathbf{r}}_q),\ (\forall q \in \overline{1, N_{\mathrm{mod}}})\ \bar{\mathbf{r}}_q \in \tilde{\mathcal{X}} \subset \mathcal{X}, \tag{3.16}$$

where v_q is defined in (3.4) and $\tilde{\mathcal{X}}$ is the set of unknown noiseless centers (3.3). We make the following two important observations with regards to approximation of $f^\star$:

1. Lemma 3.1 shows that $f^\star \in \mathcal{H}_{\mathrm{G}}^{\sigma_n}$ can also be approximated in $\mathcal{H}_{\mathrm{G}}^{\sigma}$ for a sufficiently small σ if $0 < \sigma < \sqrt{2}\sigma_n$. Therefore, we can (in principle) obtain a function $g^\star = \sum_{i=1}^{\infty} \alpha_i\ \kappa_{\mathrm{G}}^{\sigma}(\cdot, \mathbf{u}_i) \in \mathcal{H}_{\mathrm{G}}^{\sigma}$ satisfying (3.13), for given $(\mathbf{u}_i)_{i\in\mathbb{N}}$ and $(\alpha_i)_{i\in\mathbb{N}}$, without the centers $\mathbf{u}_i$ being equal to centers $\bar{\mathbf{r}}_q$ in (3.16). Therefore, the exact knowledge of $\tilde{\mathcal{X}}$ (the size of which increases exponentially with number of users K) is not required. Furthermore, working in RKHSs is computationally attractive because, as we shall see later, these spaces allow for low-complexity approximation algorithms based primarily on easy-to-compute inner-products.

2. Since the approximation requires the width of the Gaussian kernel to satisfy $0 < \sigma < \sqrt{2}\sigma_n$, a good approximation of the noise power $2\sigma_n^2$ should suffice. An important fact, which will be utilized in Section 3.4, is that the RKHS $\mathcal{H}_{\mathrm{G}}^{\sigma}$ does not contain any polynomials (including the linear and the nonzero constant functions) [Min10].

3.3.2 The MAP Filter Output with Bounded Noise

In the previous section we have shown that the optimal MAP filter $f^\star$ belongs to the space $\mathcal{H}_{\mathrm{G}}^{\sigma}$ if $0 < \sigma < \sqrt{2}\sigma_n$. Therefore, the RKHS $\mathcal{H}$ can be chosen to be the RKHS $(\mathcal{H}_{\mathrm{G}}^{\sigma_n}, \langle\cdot,\cdot\rangle_{\mathcal{H}_{\mathrm{G}}^{\sigma_n}})$, i.e., in (3.13) we can take $g^\star \in \mathcal{H}_{\mathrm{G}}^{\sigma_n}$. Next, we restrict $f^\star$ to a *feasible solution set* contained in $\mathcal{H}_{\mathrm{G}}^{\sigma_n}$ under Assumption 3.1. To this end, Lemma 3.2 provides a bound on the distance between the output of $f^\star$ and the modulation symbol of the desired user (i.e., the desired output) reminiscent of (3.10). The proof is provided in Appendix 3.6.1. We will show later that this bound enables us to restrict $f^\star$ to an intersection of sequentially arriving closed convex sets. Consequently, since this intersection is nonempty, we can apply the online learning rationale of Section 3.2.3.

Lemma 3.2. *Suppose Assumption 3.1 holds and that*

$$(\forall p \in \tilde{\mathcal{X}})\ (\forall q \in \tilde{\mathcal{X}}\backslash\{p\})\ \frac{\|\bar{\mathbf{r}}_p - \bar{\mathbf{r}}_q\|}{2} := \alpha_{disparity}^{p,q} \geq W_{noise}, \tag{3.17}$$

where $\tilde{\mathcal{X}}$ is the set of noiseless centers (3.3). *Define*

$$\epsilon_1 := \left|1 - \ exp\left(-\frac{W^2_{noise}}{2\sigma_n^2}\right)\right|, \ \epsilon_2 := \max_{q\in\overline{1,N_{mod}}} \left|\sum_{p=1,p\neq q}^{N_{mod}} exp\left(-\frac{(\alpha^{p,q}_{disparity})^2}{2\sigma_n^2}\right)\right| \tag{3.18}$$

Then, the optimal MAP filter $f^\star$ satisfies

$$(\forall t\in\mathbb{N}) \ \ |f^\star(\mathbf{r}(t)) - b_1(t)| \leq \epsilon_1+\epsilon_2 =: \epsilon, \tag{3.19}$$

where $(b_1(t))_{t\in\mathbb{N}} \subset \{+1,-1\}$ *are the modulation symbols of the desired user and* $(\mathbf{r}(t))_{t\in\mathbb{N}} \subset \mathcal{X}$ *are the received vectors.*

Remark 3.4 (Example: A BSPK Scenario). In Lemma 3.2 ϵ_1 bounds the desired part of the filter output that only depends on the bound on the noise sequence, while ϵ_2 provides a worst-case bound on the unwanted residual interference part. The constant ϵ_2 depends on the disparity between user channels so it is determined by the scenario. In multiuser systems, such as NOMA systems, users are multiplexed in the SNR domain by keeping some disparity between users [WRS+16, DLK+17]. Furthermore, users are generally also separated in the spatial domain. This means that ϵ_2 is generally small. For example, in a standard BPSK system (depicted in Figure 3.5 with the desired user 1 in the middle) with $K = 5$ users, $M = 2$ antennas, SNRs (in dBs) equal to $\{10, 13, 16, 19, 22\}$, angles of arrival given as $\{50°, 70°, 60°, 40°, 30°\}$, and the noise parameter $\sigma_n = \sqrt{0.1}$, users are linearly inseparable. The channel for the kth user is given by $(\forall k \in \overline{1,K})$ $\mathbf{h}_k := [1, e^{\pi j\cos\theta_k}, \ldots, e^{\pi j(M-1)\cos\theta_k}]^\intercal \in \mathbb{C}^M$, where θ_k is the angle of arrival (in radians). In this case $\epsilon_2 = 0.0726$. If we assume a reasonably large bound $W_{\text{noise}} = 2\sigma_n$ then from (3.18) we get $\epsilon_1 = 0.8647$. For reliable BPSK detection this satisfies the condition $\epsilon_1+\epsilon_2 = \epsilon < 1$ in (3.10).

In the following, we define $\sigma := \sigma_n$ for notational simplicity. Using the reproducing property of $(\mathcal{H}^\sigma_{\text{G}}, \langle\cdot,\cdot\rangle_{\mathcal{H}^\sigma_{\text{G}}})$, we can rewrite (3.19) as

$$(\forall t\in\mathbb{N}) \ \ \left|\langle f^\star, \kappa^\sigma_{\text{G}}(\cdot,\mathbf{r}(t))\rangle_{\mathcal{H}^\sigma_{\text{G}}} - b_1(t)\right| \leq \epsilon. \tag{3.20}$$

Furthermore, the set

$$(\forall t\in\mathbb{N}) \ C_t := \left\{f\in\mathcal{H}^\sigma_{\text{G}} : \left|\langle f, \kappa^\sigma_{\text{G}}(\cdot,\mathbf{r}(t))\rangle_{\mathcal{H}^\sigma_{\text{G}}} - b_1(t)\right| \leq \epsilon\right\} \subset \mathcal{H}^\sigma_{\text{G}} \tag{3.21}$$

is a closed convex set of functions in $\mathcal{H}^\sigma_{\text{G}}$ also known as a hyperslab and in particular $(\forall t\in\mathbb{N})$ $f^\star \in C_t$. We can now obtain the following result which shows that fixing an

$\epsilon > 0$ in (3.19) restricts $f^\star$ to an intersection of certain closed convex sets under certain technical assumptions.

Theorem 3.1. *Let the RKHS* $(\mathcal{H}_{\mathrm{G}}^{\sigma}, \langle\cdot,\cdot\rangle_{\mathcal{H}_{\mathrm{G}}^{\sigma}})$ *associated with the Gaussian kernel* $\kappa_{\mathrm{G}}^{\sigma}$ *satisfy the condition in Lemma 3.1. Fix* $\epsilon > 0$ *and suppose that* $\epsilon > \epsilon_2$ *and that* (3.17) *is satisfied. Then,*

a). $(\forall t_o \in \mathbb{N})$ $(\forall t \geq t_o)$ $f^\star \in \bigcap_{t\geq t_o \in \mathbb{N}} C_t \subset \mathcal{H}_{\mathrm{G}}^{\sigma}$.

b). $(\forall t_o \in \mathbb{N})$ $(\forall t \geq t_o)$ $C_t := \left\{ f \in \mathcal{H}_{\mathrm{G}}^{\sigma} : \left| \langle f, \kappa_{\mathrm{G}}^{\sigma}(\cdot, \mathbf{r}(t)) \rangle_{\mathcal{H}_{\mathrm{G}}^{\sigma}} - b_1(t) \right| \leq \epsilon \right\} \neq \emptyset$.

Proof. Note that ϵ_2 is determined by the system (see Remark 3.4) because it depends on the set $\tilde{\mathcal{X}}$ containing the noiseless centers. Then by fixing $\epsilon > 0$, the value $W_{\text{noise}} = W_{\text{noise}}^{\epsilon}$ can be calculated by (3.18) if $\epsilon > \epsilon_2$. If (3.17) is satisfied for this W_{noise}, then Lemma 3.2, (3.20), and (3.21) yield $(\forall t \in \mathbb{N})$ $f^\star \in C_t$. Therefore, $(\forall t \in \mathbb{N})$ $C_t \neq \emptyset$, which implies both (a) and (b).

□

3.3.3 Online Approximation of the Proxy Optimal Filter

In light for Theorem 3.1, we consider approximating a proxy optimal filter $(\mathcal{H}_{\mathrm{G}}^{\sigma} \ni)$ $g^\star : \mathbb{C}^M \to \mathbb{R}$ satisfying $(\forall t \in \mathbb{N})$ $g^\star(\mathbf{r}(t)) \approx f^\star(\mathbf{r}(t))$, which has the general form given by $g^\star = \sum_{i=1}^{\infty} \alpha_i \kappa_{\mathrm{G}}^{\sigma}(\cdot, \mathbf{r}(t))$, where the coefficients $(\alpha_i)_{i\in\mathbb{N}} \subset \mathbb{R}$ and centers $(\mathbf{x}_i)_{i\in\mathbb{N}} \subset \mathbb{C}^M$ are determined by the proposed learning algorithm. If for the fixed ϵ above, the conditions in Theorem 3.1 are satisfied we have that $(\forall t \in \mathbb{N})$ $f^\star \in C_t$ and therefore $f^\star \in \bigcap_{t\in\mathbb{N}} C_t$, where the sets C_t are defined in (3.21). The online nature of the problem means that the sets C_t arrive sequentially. Ideally, we would like to obtain some function $g^\star \in \bigcap_{t\in\mathbb{N}} C_t$ because $f^\star \in \bigcap_{t\in\mathbb{N}} C_t$, and therefore $g^\star$ is a good approximation of $f^\star$ because it agrees with all the information we have about $f^\star$ [BB96, Com93]. In the following, we relax our approximation problem to finding a function in $\bigcap_{t\geq t_o} C_t$, for some $t_o \in \mathbb{N}$, i.e., we find a point in the intersection of all (infinite in number) but some finite number of information sets (also see Section 3.2.3). The advantage of this technical relaxation (which should not affect the quality of approximation significantly) is that we can use a computationally convenient and robust algorithm. To this end, in the following we discuss the proposed online learning algorithm and some of its important features.

Canonical HyperSlab APSM-based Algorithm

Since hyperslabs are closed convex sets, we can use a particular version of APSM [YO05] that we refer to as the HyperSlab APSM (see, e.g., [STY09, CSTY13, CYM09, CYS04, CYY05, CYM09]). In an online setting, at any time t, we have access to training samples $\mathcal{S}(i) := \{(\mathbf{r}(i), b_1(i))\}_{i\in\overline{1,t}}$ or equivalently the training hyperslabs $(C_i)_{i\in\overline{1,t}}$ by, for example, storing the past samples in the memory. HyperSlab APSM allows us to use a finite number of hyperslabs to be used concurrently during a single iteration of the algorithm. This has been shown to accelerate the convergence of the iterate towards $\bigcap_{t\geq t_o} C_t$ [The15, Chapter 8]. In more detail, denote by $\mathcal{J}_t \subset \mathbb{N}$ the indices of the sets in $(C_i)_{i\in\overline{1,t}}$ that we intend to process at iteration/time t of the training procedure. Starting from an arbitrary $f_1 \in \mathcal{H}_{\mathrm{G}}^{\sigma}$, the simplified HyperSlab APSM iteration is given by [**?**]

$$(\forall t \in \mathbb{N})\ f_{t+1} = \sum_{j\in\mathcal{J}_t} q_j^t\ P_{C_j}(f_t), \tag{3.22}$$

where $P_{C_j}(f_t)$ is the projection of f_t onto the set C_t, and where $(q_j^t)_{j\in\mathcal{J}_t}$ are nonnegative weights satisfying $\sum_{j\in\mathcal{J}_t} q_j^t = 1$. It is important to mention that $P_{C_j}(f_t)$ has a closed-form (see, e.g., [The15]) because C_t is a hyperslab. So the iterations (3.22) have a low complexity.

Convergence, Monotonicity, and Robustness of the HyperSlab APSM

Informally, the proposed HyperSlab APSM iterative online algorithm improves the current estimate, that we denote by f_t, as soon as C_t arrives, i.e., it does not wait for the acquisition of the whole training sample set to start the learning process. This property means that the detection of $b_1(t)$ can start immediately starting at some $t = T_{\text{train}}+1$ which is vital in high data rate systems [see also Section 3.2.3]. Note that the asymptotic convergence of (3.22) to a point in $\bigcap_{t\in\mathbb{N}} C_t$ can be established under certain assumptions [YO05, Theorem 2], but we are interested in the behavior of the algorithm for a finite training time.

In Theorem 3.2, we use a result from [YO05, Theorem 2(a) and 2(b)] which provides the sufficient conditions under which the monotonicity of the general APSM iteration is established. For completeness, we provide a proof for the fulfillment of these conditions in our setting in Section 3.6.2. The proof, which follows from examples in [YO05], serves a dual purpose in that it also shows how the iteration (3.22) can be obtained from the general APSM algorithm [YO05]. In Section 3.6 we demonstrate Theorem 3.2 by simulation in a standard multiuser scenario.

Theorem 3.2. *[YO05] If $f^\star \in \bigcap_{t\in\mathbb{N}} C_t$, then for the APSM iteration* (3.22) *it holds that, if* $(\forall t \in \mathbb{N})$ $f_t \notin \cap_{j\in\mathcal{J}_t} C_j$,

$$(\forall t \in \mathbb{N})\ \|f^\star - f_{t+1}\|_{\mathcal{H}_{\mathrm{G}}^{\sigma}} < \|f^\star - f_t\|_{\mathcal{H}_{\mathrm{G}}^{\sigma}}.$$

It is known that the APSM iteration in (3.22) is robust against a finite number of excessive noise events $\|\mathbf{n}(t)\| > W^{\epsilon}_{\text{noise}}$ which may result in the violation of the condition (3.17) and therefore also of Theorem 3.1(a) for our choice of ϵ. To be more precise, suppose at some iteration $t = t_o$, $\|\mathbf{n}(t)\| > W^{\epsilon}_{\text{noise}}$. Then, if $f^\star \in \bigcap_{t>t_o} C_t$ we have [YO05, TSY11]

$$(\forall t > t_o)\ \|f^\star - f_{t+1}\|_{\mathcal{H}^{\sigma}_{\text{G}}} < \|f^\star - f_t\|_{\mathcal{H}^{\sigma}_{\text{G}}};$$

i.e., the algorithm "resets" and starts moving toward $f^\star$ again.

3.4 Robust and Practical Implementation

In Section 3.3 we have presented our results regarding adaptive approximation of a good nonlinear filter in an RKHS that only contains nonlinear functions. In this section, in order to add robustness to our design, we will extend our filter design by adding a linear component. This is achieved by extending the RKHS by adding linear functions to it. Furthermore, we present a practical implementation of an online learning based multiuser receiver, taking into consideration complexity and memory aspects.

Before we proceed, we summarize the practical considerations in implementing the online algorithm presented in Section 3.3.3:

1. Lemma 3.1 shows that the width of the Gaussian kernel should be chosen such that it satisfies $0 < \sigma < \sqrt{2}\sigma_n$, where σ_n is the noise standard deviation. This, in particular, implies that an arbitrarily small nonnegative σ can be chosen. In Section 3.4.1, we argue that σ should be chosen such that it is close to σ_n which requires a rough estimation of the receiver noise variance.

2. The extension to higher modulation schemes (e.g., the quadrature phase-shift keying (QPSK)) requires filtering in complex-valued Hilbert spaces because the output of any receive filter is generally complex-valued, while the canonical HyperSlab APSM in (3.22) works with real-valued Hilbert spaces.

3. The MAP receiver is nonlinear and, like *all* nonlinear receivers, it is sensitive to changes in the environment. For example, in 5G machine-type dynamic environments devices transmit sporadically which means that users enter and leave the environment intermittently [also see Remark 3.1]. This may degrade the performance of a purely nonlinear receiver in dynamic environments.

4. Since the algorithm in Section 3.3.3 is of an online nature, the complexity and memory aspects need to be taken into account.

3.4.1 Selection of the Gaussian RKHS

In general the selection of a suitable approximation space, which we refer to as the *problem element set* $\mathcal{H}$, is a nontrivial problem. In Example 3.1 we discuss this general problem briefly in classical learning theory by omitting unnecessary mathematical details. See [NG96, CS02, Pon03] for further details.

Example 3.1 (A Balancing Act). Suppose we are given a training sample set $\mathcal{D}$ consisting of noisy observations of some unknown function f_ρ. We seek to approximate f_ρ from a class of functions $\mathcal{H}$ by, e.g., minimizing the *empirical risk/error*. The empirical risk/error, given $\mathcal{D}$, can be decomposed into the *approximation bias error* and the *sample error* (also known as *estimation error*). The approximation bias error depends on the learning capacity of $\mathcal{H}$, while the sample error depends on both $\mathcal{H}$ and the size of $\mathcal{D}$. The richer/larger the class $\mathcal{H}$ is, the smaller the approximation bias error but larger the sample error tends to be. Generally, one can then improve the sample error only by increasing the number of samples. Therefore, there exists a balancing act in the sense that one should select an $\mathcal{H}$ that is not unnecessarily large or, in other words, "it is just rich enough" to obtain a good approximation of f_ρ without requiring a very large $\mathcal{D}$.

We have shown in Lemma 3.1 that the optimal MAP filter $f^\star$ belongs to certain special Gaussian RKHSs associated with certain Gaussian kernels. Lemma 3.1 shows the range in which the width σ of the Gaussian kernel, which is associated to a unique RKHS $(\mathcal{H}_{\mathrm{G}}^{\sigma}, \langle\cdot,\cdot\rangle_{\mathcal{H}_{\mathrm{G}}^{\sigma}})$, may be chosen in relation to σ_n to ensure $f^\star \in \mathcal{H}_{\mathrm{G}}^{\sigma}$. In particular, Lemma 3.1 shows that any $0 < \sigma < \sqrt{2}\sigma_n$ (where σ_n is the noise standard deviation) will do the job which means that any arbitrarily small $0 < \sigma < \sigma_n$ may be chosen. It is known that for two widths σ_1 and σ_2, if $\sigma_2 < \sigma_1 < 1$, then $\mathcal{H}_{\mathrm{G}}^{\sigma_1} \subset \mathcal{H}_{\mathrm{G}}^{\sigma_2}$ [Bel18], i.e., decreasing the width of the Gaussian kernel expands the functions space by adding new functions. But note that we know that $f^\star \in \mathcal{H}_{\mathrm{G}}^{\sigma_n}$, which is unlike the general learning problem mentioned in Example 3.1 where no such information is available for function f_ρ. Therefore, we do not need a larger space than $\mathcal{H}_{\mathrm{G}}^{\sigma_n}$, and operating in a larger space of functions is likely to require more samples to approximate $f^\star$. In other words, though "narrower" Gaussian kernels are more expressive, they may also require more samples to approximate functions. To conclude, a good approximation of receiver noise power is required for a good detection performance with a relatively small sample size. In this sense, the information about receiver noise power adds to our overall model based knowledge. We shall demonstrate this observation by simulation in Section 3.5.

In the next section we show how our method described in previous sections can be extended to higher modulation schemes (e.g., QPSK) requiring complex modulation symbols.

3.4.2 Detection in Real-Valued RKHS

We use the approach described in [STY09], which exploits the bijection between $\mathbb{C}^m$ and $\mathbb{R}^{2m}$ to estimate complex symbols with real RKHSs. For each $t \in \mathbb{N}$, we split $\mathbf{r}(t) \in \mathbb{C}^M$ in (3.1) into two $2M$-dimensional vectors given by

$$\mathbf{r}_1(t) := \begin{bmatrix} \Re(\mathbf{r}(t))^\intercal \\ \Im(\mathbf{r}(t))^\intercal \end{bmatrix} \in \mathbb{R}^{2M}, \ \mathbf{r}_2(t) := \begin{bmatrix} \Im(\mathbf{r}(t))^\intercal \\ -\Re(\mathbf{r}(t))^\intercal \end{bmatrix} \in \mathbb{R}^{2M}.$$

Similarly, the complex symbols $(b_1(t))_{t\in\mathbb{N}}$ of the desired user are mapped to vectors

$$\begin{bmatrix} b_{1,1}(t) \\ b_{1,2}(t) \end{bmatrix} := \begin{bmatrix} \Re(b_1(t)) \\ \Im(b_1(t)) \end{bmatrix} \in \mathbb{R}^2.$$

Our task is to learn a function $f : \mathbb{R}^{2M} \to \mathbb{R}$ that operates on $\mathbf{r}_1(t)$ and $\mathbf{r}_2(t)$ separately as illustrated in Figure 3.3. The relation between f and its complex-valued counterpart $f^{\mathrm{c}} : \mathbb{C}^M \to \mathbb{C}$ is given by

$$(\forall t \in \mathbb{N}) \ f^{\mathrm{c}}(\mathbf{r}(t)) = f(\mathbf{r}_1(t)) + i f(\mathbf{r}_2(t)), \tag{3.23}$$

where i is determined by $i^2 = -1$. To handle the BPSK case, assumed throughout in Section 3.3, we simply set $f(\mathbf{r}_2(t)) = 0$ during detection.

To simplify notation in the discussion that follows, we define:

$$(\forall t \in \mathbb{N}) \ (\forall l \in \overline{1,2}) \ n := 2t + l - 2, \mathbf{y}_n = \mathbf{y}_{2t+l-2} := \mathbf{r}_l(t), \text{ and } s_n = s_{2t+l-2} := b_{1,l}(t).$$

The advantage of using this simplified notation is that we have natural mappings from natural numbers to the real and imaginary parts of the complex symbols $(b_1(t))_{t\in\mathbb{N}}$ and the complex received signals $(\mathbf{r}(t))_{t\in\mathbb{N}}$.

Note that, since we have transformed the received vectors from $\mathbb{C}^M$ to $\mathbb{R}^{2M}$, we also transform the set of centers $\tilde{\mathcal{X}} \subset \mathbb{C}^M$ and the set of received signals $\mathcal{X} \subset \mathbb{C}^M$ in Section 3.3.3, to sets $\tilde{\mathcal{R}} \subset \mathbb{R}^{2M}$ and $\mathcal{R} \subset \mathbb{R}^{2M}$, respectively. With this transformation, and under Assumption 3.1 $(\forall n \in \mathbb{N})$ $\mathbf{y}_n \in \mathcal{R}$, and we refer to $\mathcal{R}$ as the *input space* of received signals.

3.4.3 Robust Partially Linear Filtering

Two important qualities of an ideal receive filter are a “high resolution” and “robustness” against changes in the environment. Having a high resolution means that the receiver is able to detect users well even if they are not separated well in space. Generally, nonlinear

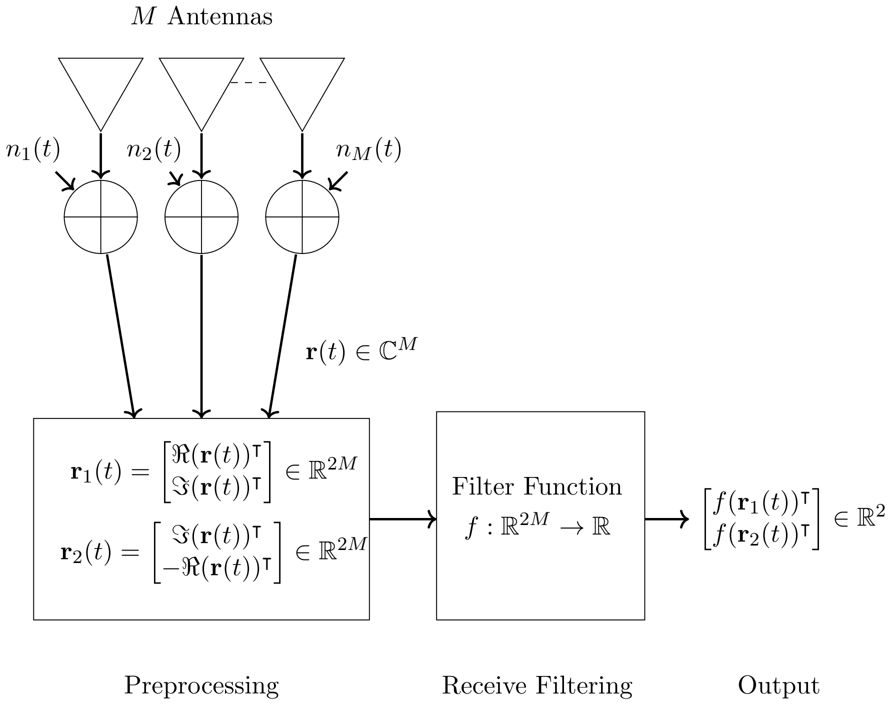

M Antennas
$n_1(t)$
$n_2(t)$
$n_M(t)$
$\mathbf{r}(t) \in \mathbb{C}^M$
$\mathbf{r}_1(t) = \begin{bmatrix} \Re(\mathbf{r}(t))^\intercal \\ \Im(\mathbf{r}(t))^\intercal \end{bmatrix} \in \mathbb{R}^{2M}$
$\mathbf{r}_2(t) = \begin{bmatrix} \Im(\mathbf{r}(t))^\intercal \\ -\Re(\mathbf{r}(t))^\intercal \end{bmatrix} \in \mathbb{R}^{2M}$
Filter Function
$f : \mathbb{R}^{2M} \to \mathbb{R}$
$\begin{bmatrix} f(\mathbf{r}_1(t))^\intercal \\ f(\mathbf{r}_2(t))^\intercal \end{bmatrix} \in \mathbb{R}^2$
Preprocessing
Receive Filtering
Output

Figure 3.3: Uplink Detection: The received base band signal $\mathbf{r}(t)$ is split into two real parts $\mathbf{r}_1(t)$ and $\mathbf{r}_2(t)$ to enable real processing. Note that the illustration shows the processing for a single desired user, but the same processing is applied to every desired user in parallel.

filters have a higher resolution than their linear counterparts for the same number of receive antennas. In [STY09] the authors design a nonlinear receive filter (which they refer to as a nonlinear beamformer) in an infinite-dimensional RKHS associated with the Gaussian kernel. The employed method is the HyperSlab APSM in Section 3.3.3 (with closed-convex sets similar to those in (3.21) for a fixed value of ϵ) and the modulation scheme is BPSK. In contrast to our work, the authors, however, do not provide a specific reason for using a Gaussian kernel besides the fact that the resulting filter design is nonlinear. The Gaussian kernel is one of the most widely used kernels in nonlinear regression due to its powerful approximation properties. Interestingly, we have shown [see Lemma **??**] that, in fact, the choice of a (certain) Gaussian kernel is indeed well-motivated if the objective is to minimize the BER of the desired user.

Unfortunately, one of the main drawbacks of *all* nonlinear filters (including the optimal MAP filter) is that they tend to be less robust than linear filters against changes in the environment. For instance, in the case of linear filtering, if a user leaves the system during detection, the SINRs of the remaining users improve. However, this property cannot be ensured in general with nonlinear filters. The performance may in fact deteriorate. In systems where users transmit sporadically, such as massive machine-type communications, robustness and reliability are important requirements. In [STY09], robustness is achieved by incorporating the knowledge about the angles of arrival of user signals. Since we do not assume any information about user channels or their angles of arrival, we consider an alternative approach to achieving robustness.

To achieve the robustness of conventional linear filters and the high resolution of the nonlinear filters simultaneously, we can naturally assume that an ideal filter should have a linear component f_{L} and a nonlinear component f_{G}. Recall that the theory in Section 1.5.1 has been developed for these kind of scenarios. Furthermore, the nonlinear part should be an approximation of the proxy optimal filter $g^{\star}$, with the complex-to-real transformation defined in (3.23). In more detail, we propose to work with a function space that contains functions of the type $f := f_{\mathrm{L}} + f_{\mathrm{G}}$, where ideally we want f_{G} to be the proxy optimal filter $g^{\star}$. To this end, we propose to work in the sum space of a linear and a Gaussian kernel. The kernels are given by

$$(\forall \mathbf{u} \in \mathcal{R})\ (\forall \mathbf{v} \in \mathcal{R})\ \kappa_{\mathrm{L}}(\mathbf{u}, \mathbf{v}) := \mathbf{u}^T \mathbf{v} \text{ and } \kappa_{\mathrm{G}}(\mathbf{u}, \mathbf{v}) := \exp\left(-\frac{\|\mathbf{u} - \mathbf{v}\|^2_{\mathbb{R}^l}}{2\sigma^2}\right),$$

respectively, where $0 < \sigma < \sqrt{2}\sigma_n$. The kernels κ_{L} and κ_{G} are known to be reproducing kernels associated with RKHSs $(\mathcal{H}_{\mathrm{L}}, \langle\cdot,\cdot\rangle_{\mathcal{H}_{\mathrm{L}}})$ and $(\mathcal{H}_{\mathrm{G}}, \langle\cdot,\cdot\rangle_{\mathcal{H}_{\mathrm{G}}})$, respectively. Fact 3.1 shows that this is equivalent to extending $\mathcal{H}_{\mathrm{G}}$ to the RKHS $\mathcal{H}$ that now also contains certain linear functions.

Fact 3.1 (Sum RKHS). *[Min10, Yuk15a] Denote by $\mathcal{H} := \mathcal{H}_L + \mathcal{H}_G$ the sum space RKHS associated with the kernel $\kappa := w_L\ \kappa_L + w_G\ \kappa_G$, where $w_L, w_G > 0$. With this particular sum space $\mathcal{H}$, if Int$(\mathcal{R}) \neq \emptyset$, then*

$$\mathcal{H}_L \cap \mathcal{H}_G = \{0\},$$

which implies that $\mathcal{H}_G$ does not contain any linear functions including nonzero constant functions.

As a result of Fact 3.1, norms and inner-products in the Hilbert space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H},\boldsymbol{w}})$, where $\boldsymbol{w} := [w_{\mathrm{L}},\ w_{\mathrm{G}}]$, can be easily computed as shown in (1.4) and (1.5). If we use convex weighting, i.e., $w_{\mathrm{L}} = 1 - w_{\mathrm{G}}$, with $w_{\mathrm{G}} \in\ ]0, 1[$, then w_{G} controls the nonlinearity of the aggregate filter.

Based on the results in Section 3.3, we assume that the sum filter belongs to closed convex sets given by

$$(\forall n \in \mathbb{N})\ C_n := \left\{ g \in \mathcal{H} : |g(\mathbf{y}_n) - s_n| = |\left\langle g, \kappa(\mathbf{y}_n, \cdot)\right\rangle_{\mathcal{H},\boldsymbol{w}} - s_n| \leq \epsilon \right\},$$

where $\epsilon \in]0\ \infty[$ is the noise-bound parameter, and $\mathbf{y}_n \in \mathcal{R}$ is the real received signal. With these sets, similar to Section 3.3.3, we pose the learning problem as follows:

$$\text{find } f \in \mathcal{H} \text{ such that } f \in \bigcap_{n \geq N_o} C_n, \tag{3.24}$$

under the assumption that $\bigcap_{n \geq N_o} C_n \neq \emptyset$, for some $n \geq N_o$.

We can solve (3.24) by using the HyperSlab APSM introduced in Section 3.3.3) under certain assumptions [YO05]. Denote by $\mathcal{J}_n \subset \mathbb{N}$ the indices of the sets in $(C_n)_{n \in \mathbb{N}}$ that we intend to process at iteration $n \in \mathbb{N}$ of the training procedure. Starting from $f_1 = \mathbf{0} \in \mathcal{H}$, the HyperSlab APSM produces a sequence of filters $(f_n)_{n \in \mathbb{N}}$ in $\mathcal{H}$ with the iterations given by

$$(\forall n \in \mathbb{N})\ f_{n+1} = \sum_{j \in \mathcal{J}_n} q_j^n P_{C_j}(f_n), \tag{3.25}$$

where $P_{C_j}(f_n) = f_n + \beta_j^n \kappa(\mathbf{y}_j, \cdot) = f_n + \beta_j^n (w_{\mathrm{L}}\ \kappa_{\mathrm{L}}(\mathbf{y}_j, \cdot) + w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\mathbf{y}_j, \cdot))$ is the projection of f_n onto the set C_j, with β_j^n given by [see Section 1.6.2]

$$\beta_j^n := \begin{cases} \frac{s_j - \langle f_n, \kappa(\mathbf{y}_j, \cdot)\rangle_{\mathcal{H},\boldsymbol{w}} - \epsilon}{\kappa(\mathbf{y}_j, \mathbf{y}_j)}, & \text{if } \langle f_n, \kappa(\mathbf{y}_j, \cdot)\rangle_{\mathcal{H},\boldsymbol{w}} - s_j < -\epsilon, \\ 0, & \text{if } |\langle f_n, \kappa(\mathbf{y}_j, \cdot)\rangle_{\mathcal{H},\boldsymbol{w}} - s_j| \leq \epsilon, \\ \frac{s_j - \langle f_n, \kappa(\mathbf{y}_j, \cdot)\rangle_{\mathcal{H},\boldsymbol{w}} + \epsilon}{\kappa(\mathbf{y}_j, \mathbf{y}_j)}, & \text{if } \langle f_n, \kappa(\mathbf{y}_j, \cdot)\rangle_{\mathcal{H},\boldsymbol{w}} - s_j > \epsilon, \end{cases}$$

and where $(q_j^n)_{j \in \mathcal{J}_n}$ are nonnegative weights satisfying $\sum_{j \in \mathcal{J}_n} q_j^n = 1$. The monotonicity of iteration (3.25) follows directly from Theorem 3.2:

$$(\forall f^\star \in \bigcap_{n \in \mathbb{N}} C_n)\ \|f_{n+1} - f^\star\| < \|f_n - f^\star\|,$$

if $f_n \notin \bigcap_{j \in \mathcal{J}_n} C_j$.

3.4.4 Practical Issues

Next, we look at issues related to the implementation of (3.25). The first issue is the selection of an appropriate set $\mathcal{J}_n$. A reasonable and simple way of selecting the sample set $\mathcal{J}_n$ is to include the $W \in \mathbb{N}$ most recent samples. More precisely, at time $n \in \mathbb{N}$, we define $\mathcal{J}_n$ as the set given by $\mathcal{J}_n := \overline{n - W + 1, n}$ if $n \geq W$, or $\mathcal{J}_n := \overline{1, n}$ otherwise. The window size W is a design parameter chosen based on the available computational power. Larger sizes typically improve the performance at the cost of increased computational complexity.

The second issue pertains to the memory requirement and complexity of the learning framework. To understand the main challenges, let us look at how the algorithm in (3.25) proceeds. At time $n \in \mathbb{N}$, we can show that the filter estimate generated by 3.25 is given by

$$f_n = \sum_{i=1}^{n-1} \gamma_i^{(n)} \kappa(\mathbf{y}_i, \cdot),$$

where $(\gamma_i^{(n)})_{i \in \overline{1,n-1}}$ are real coefficients that depend on the sets $(C_n)_{n \in \mathbb{N}}$ [TSY11].

Since f_n is expressed as a linear combination of the elements in the set

$$\mathcal{D}_{n-1} := \{\kappa(\mathbf{y}_1, \cdot), \kappa(\mathbf{y}_2, \cdot), \ldots, \kappa(\mathbf{y}_{n-1}, \cdot)\}, \tag{3.26}$$

f_n belongs to a subspace $\mathcal{H}_{n-1} \subset \mathcal{H}$ spanned by $\mathcal{D}_{n-1}$. Note that $\mathcal{H}_{n-1}$ is also a Hilbert space if equipped with the same inner-product of the sum RKHS $\mathcal{H}$. In the following we will refer to the set (3.26) as the *learning dictionary.*

At each iteration $n \in \mathbb{N}$ of the algorithm a new element $\kappa(\mathbf{y}_n, \cdot) = w_\mathrm{L} \ \kappa_\mathrm{L}(\mathbf{y}_n, \cdot) + w_\mathrm{G} \ \kappa_\mathrm{G}(\mathbf{y}_n, \cdot)$ is admitted, and the dictionary is extended to $\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{\kappa(\mathbf{y}_n, \cdot)\}$. It follows that $\mathcal{H}_{n-1} \subset \mathcal{H}_n \subset \mathcal{H}$, and the extended space $\mathcal{H}_n$ is spanned by $\mathcal{D}_n$. Therefore, to evaluate $f_{n+1}(\mathbf{y})$ (by using the reproducing property discussed in Section 1.5), we need to store $\mathcal{D}_n^{\mathrm{mem}} := \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$ along with the coefficients $\gamma_1^{(n+1)}, \gamma_2^{(n+1)}, \ldots, \gamma_n^{(n+1)}$ in the memory of the receiver [note: $\mathcal{D}_n \subset \mathcal{H}$ can be trivially recovered from $\mathcal{D}_n^{\mathrm{mem}} \subset \mathbb{R}^{2M}$]. Moreover, the coefficients $\gamma_i^{(n)}$, the number of which increases with n, are required at each iteration n by the projections $(j \in \mathcal{J}_n)$ $P_{C_j}(f_{n-1})$ in (3.25). This fact shows that the memory requirements and the computational complexity may become prohibitive when n becomes sufficiently large. To keep the complexity and the memory requirements of the learning algorithm at manageable levels, we need to use online dictionary learning techniques, as explained below.

3.4.5 Online Dictionary Learning

It follows from Section 1.5.1 that the filter estimate f_n at time $n \in \mathbb{N}$ can be uniquely decomposed as the sum of its linear and Gaussian components as follows:

$$f_n := \sum_{i=1}^{n-1} \gamma_i^{(n)} \kappa(\mathbf{y}_i, \cdot) := f_{\mathrm{L},n} + f_{\mathrm{G},n} = w_\mathrm{L} \ \sum_{i=1}^{n-1} \gamma_i^{(n)} \kappa_\mathrm{L}(\mathbf{y}_i, \cdot) + w_\mathrm{G} \ \sum_{i=1}^{n-1} \gamma_i^{(n)} \kappa_\mathrm{G}(\mathbf{y}_i, \cdot), \tag{3.27}$$

where $f_{\mathrm{L},n} \in \operatorname{span}(\mathcal{D}_{\mathrm{L},n-1})$ and $f_{\mathrm{G},n} \in \operatorname{span}(\mathcal{D}_{\mathrm{G},n-1})$, and where $(\forall k \in \mathbb{N})$ $\mathcal{D}_{\mathrm{L},k} = \{w_{\mathrm{L}}\ \kappa_{\mathrm{L}}(\cdot, \mathbf{y}_1), \ldots, w_{\mathrm{L}}\ \kappa_{\mathrm{L}}(\cdot, \mathbf{y}_k)\}$ and $\mathcal{D}_{\mathrm{G},k} = \{w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\cdot, \mathbf{y}_1), \ldots, w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\cdot, \mathbf{y}_k)\}$. To curb the growth of the dictionaries $\mathcal{D}_{\mathrm{L},n}$ and $\mathcal{D}_{\mathrm{G},n}$ as a function of n in a way to have a minor impact on the performance of the filter f_n, we use dictionary sparsification, as explained below.

Dictionary sparsification has its origins in the seminal work in [EMM04], but here we use an approach similar to that proposed in [Yuk15b], which handles the linear and Gaussian components of the sequence $(f_n)_{n\in\mathbb{N}}$ of filters separately. In our approach we use admission control to verify whether the most recent inputs $w_{\mathrm{L}}\ \kappa_{\mathrm{L}}(\mathbf{y}_n, \cdot)$ and $w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\mathbf{y}_n, \cdot)$ should be added to the dictionaries $\mathcal{D}_{\mathrm{L},n-1}$ and $\mathcal{D}_{\mathrm{G},n-1}$, respectively. Briefly, the idea is to check if $w_{\mathrm{L}}\ \kappa_{\mathrm{L}}(\mathbf{y}_n, \cdot)$ and $w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\mathbf{y}_n, \cdot)$ can be approximated (in some sense) by a linear combination of elements previously admitted in dictionaries $\mathcal{D}_{\mathrm{L},n-1}$ and $\mathcal{D}_{\mathrm{G},n-1}$, respectively. Newly arriving elements are only added to the dictionary if such an approximation is not possible. The particular techniques for dictionary sparsification of the linear and Gaussian components of the proposed filter are described in the next two subsections.

Dictionary for the Linear Component

Admission control for the linear part can be easily done as follows. Since $\mathcal{H}_{\mathrm{L}}$ is nothing but the Euclidean space $\mathbb{R}^{2M}$, it is spanned by the Euclidean basis

$$\mathcal{D}_{\mathrm{L}} := \{w_{\mathrm{L}}\ \kappa_{\mathrm{L}}(\mathbf{e}_1, \cdot), w_{\mathrm{L}}\ \kappa_{\mathrm{L}}(\mathbf{e}_2, \cdot), \ldots, w_{\mathrm{L}}\ \kappa_{\mathrm{L}}(\mathbf{e}_{2M}, \cdot)\},$$

where $\mathbf{e}_m \in \mathbb{R}^{2M}$ is a vector having a one at the mth index and zeros elsewhere. So, every $\kappa_{\mathrm{L}}(\mathbf{y}_n, \cdot)$ can be expressed as $w_{\mathrm{L}}\ \sum_{m=1}^{2M} [\mathbf{y}_n]_m \kappa_{\mathrm{L}}(\mathbf{e}_m, \cdot)$, with $[\mathbf{y}_n]_m$ the mth entry of $\mathbf{y}_n$. As a result, the linear component

$$(\forall n \in \mathbb{N})\ f_{\mathrm{L},n} = w_{\mathrm{L}}\ \sum_{i=1}^{n-1} \gamma_i^{(n)} \kappa_{\mathrm{L}}(\mathbf{y}_i, \cdot) = w_{\mathrm{L}}\ \sum_{m=1}^{2M} \gamma_m^{(\mathrm{L},n)} \kappa_{\mathrm{L}}(\mathbf{e}_m, \cdot)$$

consists of only $2M$ basis functions with their coefficients $\gamma_m^{(\mathrm{L},n)}$ updated by each projection $(j \in \mathcal{J}_n)$ $P_{C_j}(f_n)$ in (3.25), and we also have $(\forall n \in \mathbb{N})$ $\mathcal{D}_{\mathrm{L},n} = \mathcal{D}_{\mathrm{L}}$ and $\mathcal{H}_{\mathrm{L},n} = \mathcal{H}_{\mathrm{L}}$. With the proposed sparsification technique for the linear component, note that the memory and computational requirements of $f_{\mathrm{L},n}$ do not increase with n.

Dictionary for the Gaussian Component

The proposed sparsification technique for the dictionary $\mathcal{D}_{\mathrm{G},n}$ is based on the studies in [EMM04, ST08], and it can be summarized as follows. Suppose that we start with the dictionary $\mathcal{D}_{\mathrm{G},1} = \{w_{\mathrm{G}}\ \kappa(\mathbf{y}_1, \cdot)\}$. At time $n \geq 2$, we have the dictionary $\mathcal{D}_{\mathrm{G},n-1}$, and the objective is to determine whether the newly arriving element $w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\mathbf{y}_n, \cdot)$ should be added to $\mathcal{D}_{\mathrm{G},n-1}$ to construct $\mathcal{D}_{\mathrm{G},n}$.

Informally, if $w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\mathbf{y}_n, \cdot)$ can be well approximated by any vector in the subspace $\operatorname{span}(\mathcal{D}_{\mathrm{G},n-1})$, then functions in $\operatorname{span}(\mathcal{D}_{\mathrm{G},n-1} \cup \{w_{\mathrm{G}}\ \kappa_G(\mathbf{y}_n, \cdot)\})$ can also be well approximated by functions in $\operatorname{span}(\mathcal{D}_{\mathrm{G},n-1})$, so $w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\mathbf{y}_n, \cdot)$ does not need to be added to $\mathcal{D}_{\mathrm{G},n-1}$. As commonly done in approximation theory in Hilbert spaces, we can define as the best approximation of $w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\mathbf{y}_n, \cdot)$ in the subspace $\mathcal{H}_{\mathrm{G},n-1} := \operatorname{span}(\mathcal{D}_{\mathrm{G},n-1})$ the projection $P_{\mathcal{H}_{\mathrm{G},n-1}}(w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\mathbf{y}_n, \cdot))$. With this definition the squared norm

$$d_n := \left\| w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\mathbf{y}_n, \cdot) - P_{\mathcal{H}_{\mathrm{G},n-1}}(w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\mathbf{y}_n, \cdot)) \right\|^2_{\mathcal{H}_{\mathrm{G}}}$$

of the residual $w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\mathbf{y}_n, \cdot) - P_{\mathcal{H}_{\mathrm{G},n-1}}(w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\mathbf{y}_n, \cdot))$ serves as a measure to indicate how well the best vector $P_{\mathcal{H}_{\mathrm{G},n-1}}(w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\mathbf{y}_n, \cdot))$ in the subspace $\operatorname{span}(\mathcal{D}_{\mathrm{G},n-1})$ is able to approximate $w_{\mathrm{G}}\ \kappa_G(\mathbf{y}_n, \cdot)$. Therefore, we can update the dictionary as follows:

$$\mathcal{D}_{\mathrm{G},n} = \begin{cases} \mathcal{D}_{\mathrm{G},n-1}, & \text{if } d_n \leq \alpha, \\ \mathcal{D}_{\mathrm{G},n-1} \cup \{w_{\mathrm{G}}\ \kappa_G(\mathbf{y}_n, \cdot)\}, & \text{otherwise,} \end{cases}$$

where $\alpha > 0$ is a design parameter. For completeness we show the steps required for the computation of d_n in Section 3.6.3.

The Proposed Learning Algorithm

Applying the sparsification techniques above to the algorithm in 3.25, we obtain the following iterations:

$$\begin{aligned} f_{n+1} &= P_{\mathcal{H}_n}\Big(\sum_{j \in \mathcal{J}_n} q_j^n P_{C_j}(f_n) \Big), \\ &= P_{\mathcal{H}_n}\Big(f_n + \sum_{j \in \mathcal{J}_n} q_j^n \beta_j \kappa(\mathbf{y}_j, \cdot) \Big), \\ &= f_n + \sum_{j \in \mathcal{J}_n} q_j^n \beta_j P_{\mathcal{H}_n}(\kappa(\mathbf{y}_j, \cdot)), \text{(linearity of orthogonal projections)} \end{aligned} \tag{3.28}$$

where $f_1 = \mathbf{0} \in \mathcal{H}$,

$$P_{\mathcal{H}_n}(\kappa(\mathbf{y}_j, \cdot)) = P_{\mathcal{H}_{\mathrm{L},n}}(w_{\mathrm{L}}\ \kappa_{\mathrm{L}}(\mathbf{y}_j, \cdot)) + P_{\mathcal{H}_{\mathrm{G},n}}(w_{\mathrm{G}}\ \kappa_{\mathrm{G}}(\mathbf{y}_j, \cdot)).$$

```
Algorithm 2 Online Adaptive Filtering Algorithm
  Initialization: Fix $\epsilon > 0$, training block length $T_{\text{train}} \in \mathbb{N}$, $W \in \mathbb{N}$, $\alpha > 0$, $\mathcal{D}_0 := \emptyset$, and $f_1 = 0$.
  At $n \geq 1$ Repeat:
    1. Sample Update: The training samples $\{(\mathbf{y}_j, s_j) : j \in \mathcal{J}_n\}$ are available. Set $(\forall j \in \mathcal{J}_n)$ $q_j^n = 1/|\mathcal{J}_n|$, where $|\mathcal{J}_n|$ is the cardinality of $\mathcal{J}_n$.
    2. Dictionary Update: Follow the procedure in Section 3.4.5 to update $\mathcal{D}_{n-1}$.
    3. Adaptive Learning: Follow the procedure in Section 3.4.5 to calculate $f_{n+1}$.
```

The projection $P_{\mathcal{H}_{\text{L},n}}(w_{\text{L}}\ \kappa_{\text{L}}(\mathbf{y}_j, \cdot))$ is given by

$$P_{\mathcal{H}_{\text{L},n}}(\kappa_{\text{L}}(\mathbf{y}_j, \cdot)) = w_{\text{L}} \sum_{m=1}^{2M} [\mathbf{y}_j]_m \kappa_{\text{L}}(\mathbf{e}_m, \cdot),$$

and details of the projection $P_{\mathcal{H}_{\text{G},n}}(w_{\text{G}}\ \kappa_{\text{G}}(\mathbf{y}_j, \cdot))$ are given in Section 3.6.3. Note that the projections $P_{\mathcal{H}_n}(\kappa(\mathbf{y}_j, \cdot))$ in (3.28) ensure that $f_{n+1} \in \mathcal{H}_n$ meaning that we can track f_{n+1} using $\mathcal{D}_n$. We summarize the algorithm in Algorithm 2.

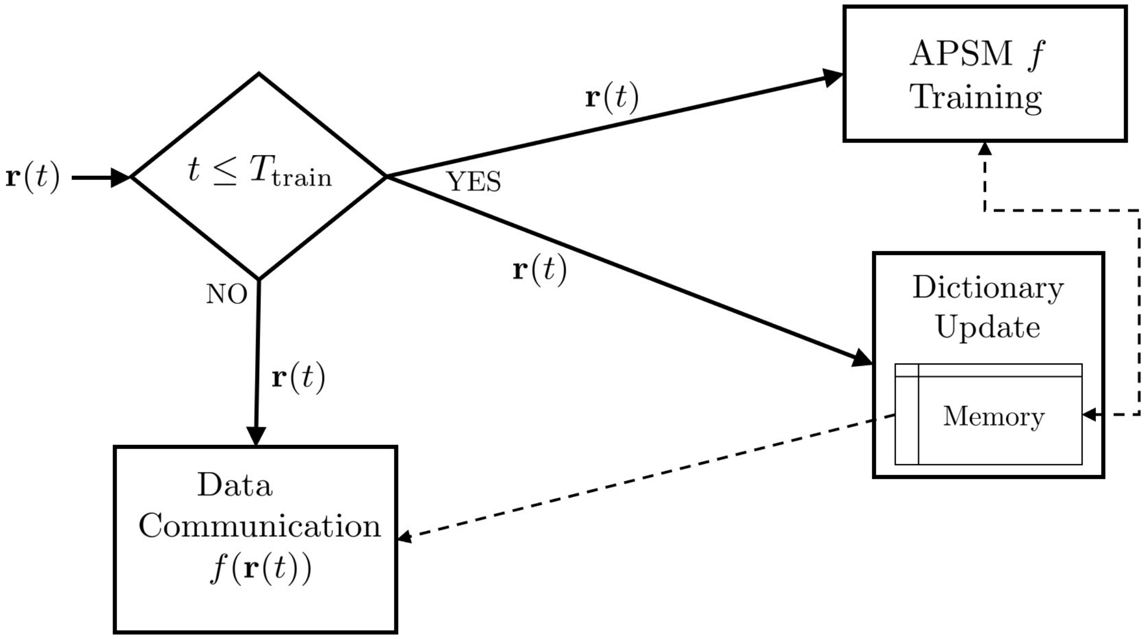


Figure 3.4: Dictionary Learning: The illustration of Algorithm 2.

Weight Tuning, Complexity, and Parallel Computation

In this section we discuss the effect of the Gaussian weight $w_{\mathrm{G}} = 1 - w_{\mathrm{L}}$ on the nonlinearity and the complexity of the filter estimate in (3.28). In more detail, increasing the weight w_{G} makes the sum filter $f = f_{\mathrm{L}} + f_{\mathrm{G}}$ more complex and nonlinear. In situations were the desired user faces strong interference from other users, we require the sum filter to be highly nonlinear which means that f_{G} should dominate the linear part f_{L}. On the other hand, if the desired user has a high SNR then a linear filter should suffice. However, it is unclear how these intuitions in static scenarios carry over to dynamic environments. Therefore, the best performance based criteria for the weights should be the BER. If for the desired user the BER performance is not satisfactory we can alter w_{G} to improve performance. In the case when the estimation of BER is not available at the receiver, the weight tuning can be done in the following simple way which adds to the complexity of the framework. Let w_{G}^{max} be the maximum Gaussian weight value. We can start by some $w_{\mathrm{G}}^{1} > 0$ and BER threshold v^{BER} and perform the learning until time $t = T_{\mathrm{train}}$. We can then estimate the BER by calculating the error rate $v^{w_{\mathrm{G}}^{1}}$ over the set $\mathcal{S} := \left\{(\mathbf{r}(t), b_1(t))\ ,\ t \in \overline{1, T_{\mathrm{train}}}\right\}$ which is available at T_{train}. If $v^{w_{\mathrm{G}}^{1}} > v^{\mathrm{BER}}$, then we can increase the weight w_{G}^{1} by some prefixed amount to $w_{\mathrm{G}}^{2} > w_{\mathrm{G}}^{1}$ and perform the learning again with $\mathcal{S}$. Since the set $\mathcal{S}$ is relatively small, a few iterations over a discrete weight set $\{w_{\mathrm{G}}^{1}, w_{\mathrm{G}}^{2}, \dots, w_{\mathrm{G}}^{max}\}$ values should suffice or we can be stop when the weight w_{G}^{max} is reached. Note that this heuristic method can also be performed in parallel over the set $\{w_{\mathrm{G}}^{1}, w_{\mathrm{G}}^{2}, \dots, w_{\mathrm{G}}^{max}\}$ using parallel computation and running Algorithm 2 in parallel for each value in $\{w_{\mathrm{G}}^{1}, w_{\mathrm{G}}^{2}, \dots, w_{\mathrm{G}}^{max}\}$. Furthermore, many operations of Algorithm 2 can be performed in parallel on modern graphical processing units (GPUs) that increases the speed of Algorithm 2 substantially. Complexity of Algorithm 2 is dominated by the dictionary sparsification step that has a quadratic complexity in the current dictionary size. However, since the dictionary size is upper bounded by the available memory size S, the complexity is upper bounded by $\mathcal{O}(S^2)$. Once the dictionary size exceeds S, we have to drop older samples to accommodate new ones.

3.5 Numerical Evaluation

Recall that our robust partially linear filter design in (3.27) consists of a linear component and a nonlinear component. The nonlinear component (which is crucial for a good performance) should ideally be a good approximation of the optimal MAP filter (3.14). We referred to this approximation as the proxy optimal filter in Section 3.3. In the first part of our numerical evaluation, which is of a rather theoretical nature, we simulate the performance of the proxy optimal filter based on the theory presented in Section 3.3 in a simple nondynamic scenario. In Section 3.4, we have extended the filter design from Section 3.3 by adding a linear component in order to make the overall design robust in dynamic environments. Moreover, we have made the canonical Hyperslab APSM algorithm [see Section 3.3.3] amenable to a practical implementation. In the second part of the numerical evaluation we simulate the performance of this robust multiuser detection framework in a more realistic and dynamic setting. Below we provide a summary of the results:

1. In Section 3.5.1 we consider a purely nonlinear design, i.e., $f = f_{\mathrm{G}}$, based on the theory presented in Section 3.3. For these results, we consider BPSK modulation, AWGN channels, and a simple nondynamic/static simulation scenario. The first result shows that by using the canonical HyperSlab APSM algorithm (3.22) we can approximate the nonlinear optimal MAP filter (3.14) well. We then compare the performance of this approximation with that of the optimal MAP filter in terms of the BER under AWGN. We simulate the observation in Section 3.4.1, to demonstrate the efficacy of using model based prior knowledge regarding the width of the Gaussian kernel. Moreover, we also compare the performance with a purely linear design $f = f_{\mathrm{L}}$, MMSE, and MMSE-SIC in a scenario where the desired user faces considerable multiuser interference.

2. In Section 3.5.2 we perform simulations to show the performance of the robust partially linear filter developed in Section 3.4 in a dynamic environment with sporadically transmitting interfering users. In this case, we consider QPSK modulation with Rayleigh block fading channels to show the performance in a realistic wireless setting. In particular, we compare the performance of our design with a purely nonlinear filter, and we show that our partially linear design is more robust against changes in the environment than a purely nonlinear design.

3.5.1 Performance of Proxy Optimal MAP Filter

In this section we consider a simple channel model illustrated in Figure 3.5. This simple channel model was considered in similar studies in [STY09, CHW04] to focus on the performance of nonlinear filtering. We consider a BPSK system with $K = 5$ users. The desired user 1 is situated in the middle and it has the lowest receive SNR. The channel for the kth user is given by

$$(\forall k \in \overline{1, K}) \ \mathbf{h}_k := [1, e^{\pi j \cos\theta_k}, \dots, e^{\pi j (M-1)\cos\theta_k}]^\intercal \in \mathbb{C}^M,$$

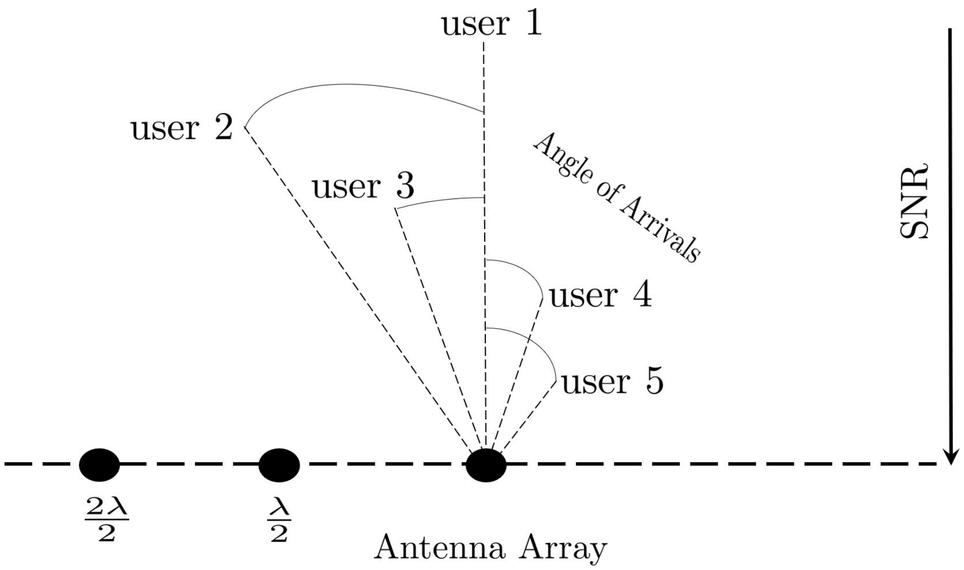


Figure 3.5: Users are separated in space in terms of their angle of arrivals and also in terms of SNRs. User 1 is the worst user with the lowest SNR and it is situated in the middle. User 5 is the best user with the best SNR.

where θ_k is the angle of arrival (in radians) and the distance between the antenna elements is $\lambda/2$. The received signal is given by [STY09, CHW04]

$$\mathbf{r} : \mathbb{N} \to \mathbb{C}^M : t \mapsto \sum_{k=1}^{K} \sqrt{p_k} b_k(t) \mathbf{h}_k + \mathbf{n}(t),$$

where $p_k \in]0\ \infty[$ and $b_k(t) = \{\pm 1\}$ are the power and the modulation symbol, respectively, for user $k \in \overline{1,K}$, and where $\mathbf{n}(t) \in \mathbb{C}^M$ denotes additive noise. Note that if we fix the noise variance σ_n^2 then p_k can be chosen based on the SNR difference between user k and user 1 by first fixing $p_1 = 1$.

Approximation of Optimal MAP Filter

Recall that in Section 3.3.3 we presented the canonical HyperSlab APSM algorithm that can be used to approximate the optimal MAP filter [we refer to the approximation as the proxy optimal filter] under the assumption that receiver noise is bounded. We fix the number of antennas to $M = 2 < K = 5$. To satisfy Assumption 3.1, for the first simulation, we "truncate" AWGN of mean 0 and variance $\sigma_n^2 = 0.1$ to a reasonably large bound $(\forall t \in \mathbb{N})\ \|\mathbf{n}(t)\| \leq W_{\text{noise}} = 2\sigma_n$ in Lemma 3.2. This truncated Gaussian/Normal noise can be generated by using the standard method in [Bot16]. The power of the desired user is set to $p_1 = 1$ in the simulation setup of Figure 3.5 [also see the discussion following

Assumption 3.1]. User SNRs in dBs are equal to $\{10, 13, 16, 19, 22\}$ in that order, while the angles of arrival are $\{50°, 70°, 60°, 40°, 30°\}$. With this setting the system parameter ϵ_2 in Lemma 3.2 which is used to calculate the hyperslab width ϵ is $\epsilon_2 = 0.0726$. The parameter $\epsilon_1 = 1 - \exp(-\frac{(W_{\text{noise}})^2}{2\sigma_n^2})$ equals $\epsilon_1 = 0.1353$. If we fix $\epsilon = 0.25 > \epsilon_1 + \epsilon_2$ then all assumptions and conditions in Lemma 3.2 and Theorem 3.1 are satisfied. Since the width of the Gaussian kernel is equal to the noise AWGN standard deviation σ_n (and we operate in the associated RKHS $\mathcal{H}_{\text{G}}^{\sigma_n}$), we can compute the relative error of approximation given by

$$\frac{\|f^\star - g^\star\|_{\mathcal{H}_{\text{G}}^{\sigma_n}}}{\|f^\star\|_{\mathcal{H}_{\text{G}}^{\sigma_n}}}, \tag{3.29}$$

where $f^\star$ is the optimal MAP filter (3.14) and $g^\star := f_{t=T_{\text{train}}}$ is the proxy optimal filter obtained by the canonical HyperSlab APSM iteration (3.22). Note that both $f^\star$ and $g^\star$ have finite known representations in $\mathcal{H}_{\text{G}}^{\sigma_n}$ and this allows us to calculate

$$\|f^\star - g^\star\|_{\mathcal{H}_{\text{G}}^{\sigma_n}} = \langle f^\star - g^\star, f^\star - g^\star \rangle_{\mathcal{H}_{\text{G}}^{\sigma_n}}^{1/2}, \|f^\star\|_{\mathcal{H}_{\text{G}}^{\sigma_n}} = \langle f^\star, f^\star \rangle_{\mathcal{H}_{\text{G}}^{\sigma_n}}^{1/2}.$$

Figure 3.6 shows the relative error of approximation (3.29) as a function of training time T_{train}. We observe that, under the bounded noise assumption, increasing the training time decreases the error monotonically [see Theorem 3.2]. Note that since $\mathcal{H}_{\text{G}}^{\sigma_n}$ is an RKHS, the relatively small error in Figure 3.6 implies that the filter output $g^\star(\mathbf{r}(t))$ is close to $f^\star(\mathbf{r}(t))$ for every $\mathbf{r}(t) \in \mathbb{C}^M$.

The proxy optimal filter was designed based on the prior knowledge about the optimal MAP filter. In particular, the choice of the Gaussian kernel width σ plays a deciding role in the approximation being carried out in the best RKHS $\mathcal{H}_{\text{G}}^{\sigma}$. According to Lemma 3.1, the Gaussian kernel width must satisfy $0 < \sigma < \sqrt{2}\sigma_n$, where σ_n is the AWGN standard deviation. However, in Section 3.4.1 we have reasoned that σ should be a good approximation of σ_n because setting $\sigma < \sigma_n$ results in an unnecessary enlargement of the RKHS [in the sense that if $\sigma < \sigma_n$ then $\mathcal{H}_{\text{G}}^{\sigma_n} \subset \mathcal{H}_{\text{G}}^{\sigma}$]. In this regard, Figure 3.7 shows the efficacy of including accurate prior knowledge in the framework. In the case when $\sigma \neq \sigma_n$, we cannot compute the error (3.29) because we do not know the representation of $f^\star$ in $\mathcal{H}_{\text{G}}^{\sigma}$. But since we know that $f^\star$ is optimal under AWGN assumption, we can use BER as a substitute for error (3.29) under AWGN. For this simulation, the noise in the system is therefore AWGN with variance $\sigma_n^2 = 0.1$. In this case the weakest desired user $k = 1$ in Figure 3.5 is linearly inseparable from other users. The hyperslab width is set to $\epsilon = 0.95$. Figure 3.7 shows an empirical demonstration of our observation in Section 3.4.1 and it shows that σ should be a good estimation of σ_n because this results in a better perfor-

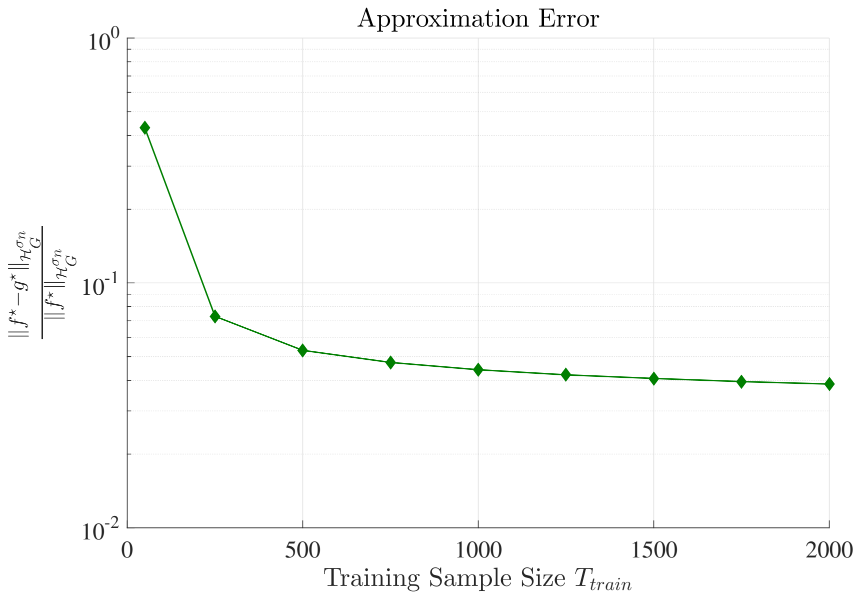


Figure 3.6: The relative error of approximation 3.29 using the APSM iteration (3.22) and operating in RKHS $\mathcal{H}_{\mathrm{G}}^{\sigma_n}$.

mance that is closer to the optimal $f^\star$. In particular, we see that an arbitrarily small value of $\sigma > 0$ should not be chosen.

Comparison with Other Techniques

In this section we compare the performance of the proxy optimal filter with other conventional techniques in terms of BER. We consider AWGN channel between users and the base station receiver in Figure 3.5.

The first result in this section compares the performance of the proxy optimal filter with that of the optimal MAP filter, the adaptive linear filter, the linear MMSE filter, and the nonlinear MMSE-SIC filter. The adaptive linear filter can be designed by using a purely linear kernel κ_{L} in our framework in Section 3.4, i.e., a filter of the form $f_{\mathrm{L}} = \sum_{i=0}^{2M} \gamma_{\mathrm{L},i}\ \kappa_{\mathrm{L}}(\cdot, \mathbf{e}_i)$. We fix the number of antennas to $M = 2$ in the simulations. User SNRs in dBs are equal to $\{10, 16, 13, 19, 22\}$ in that order, while the angles of arrival are $\{50^\circ, 70^\circ, 60^\circ, 40^\circ, 30^\circ\}$. In this case the weakest user $k = 1$ in Figure 3.5 cannot be linearly separated from other users. The results are shown in Figure 3.8. We see clearly that the proxy optimal filter (with width $\sigma = \sigma_n$) outperforms other techniques and it shows a comparable performance to optimal MAP filter with sufficient training. Note that, in contrast to the optimal MAP filter, that has complete user knowledge, the proxy

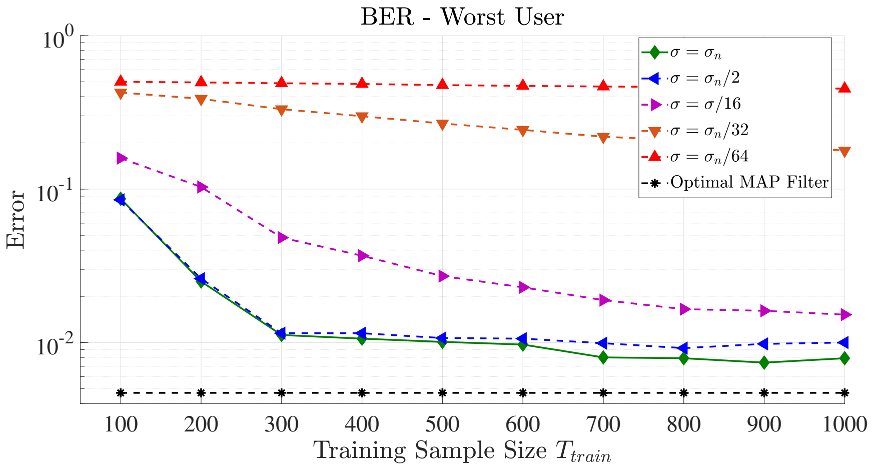


Figure 3.7: Different line plots show effect of the Gaussian kernel width σ which plays a deciding role in the selection of the suitable RKHS.

optimal filter only works with the training samples. The MMSE and MMSE-SIC filters also have complete knowledge of user channels and powers.

The second result compares the performance of the above mentioned techniques when the system gradually becomes linearly separable. We achieve this by increasing the number of receive antennas $M = \{2, 3, 4, 5, 6\}$ at the receiver. User SNRs in dBs are equal to $\{10, 16, 13, 19, 22\}$ in that order, while the angles of arrival are $\{60°, 90°, 75°, 45°, 30°\}$. The size of the training sample set is fixed at $T_{\text{train}} = 250$ samples. The results are shown in Figure 3.9. We see that as the system becomes linearly separable all techniques show a good performance.

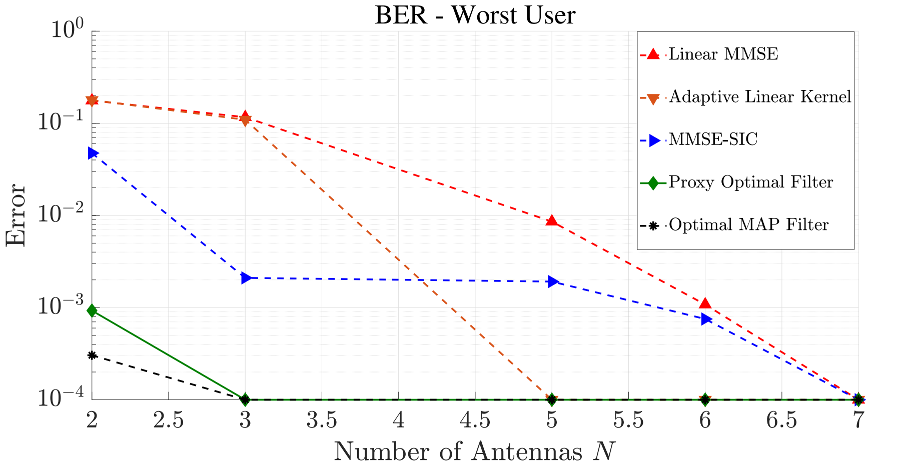


Figure 3.9: Comparison between different filtering techniques with an increasing training time T_{train} in terms of the BER of user 1.

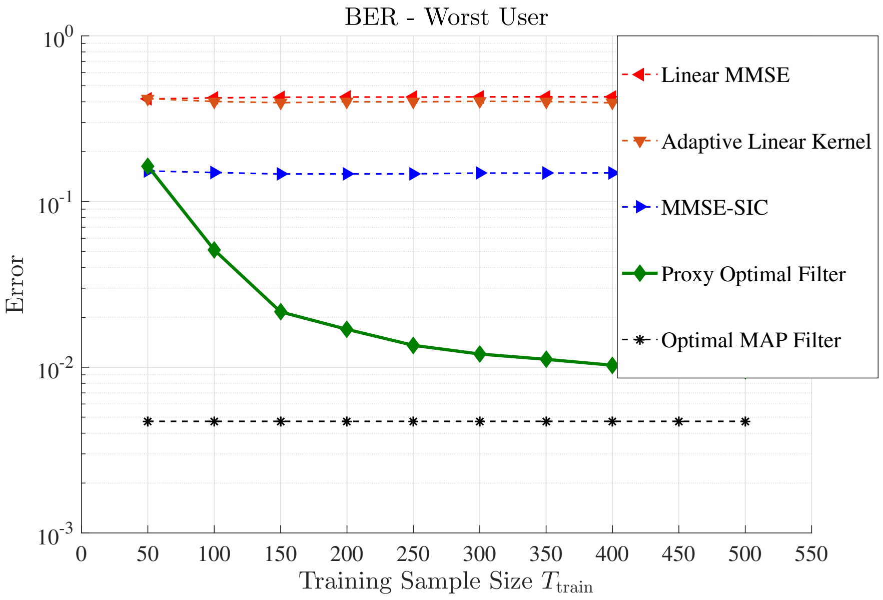


Figure 3.8: Comparison between different filtering techniques with an increasing training time T_{train} in terms of the BER of user 1.

The next two results compare the performance of the proxy optimal filter with that of 3 state-of-art nonlinear networks. For the first state-of-art technique, we consider the recursive adaptive RBF detector proposed in [CHW04]. Similar to the proxy optimal filter, this method is also online and training is performed sequentially. This method [also discussed in Section 3.1.1] is based on robust k-means clustering combined with the recursive least squares technique. Similar to the optimal MAP filter, the RBF detector has the form of an RBF network given by $\sum_{i=1}^{2^K} \beta_i \ \exp\left(-\frac{\|\cdot - \mathbf{c}_i\|^2}{2\sigma_n^2}\right)$. Robust k-means clustering is used to estimate the optimal centers $\mathbf{c}_i \in \mathbb{C}^M$, while the recursive least squares algorithm is used to calculate the coefficients $\beta_i \in \mathbb{R}$. We simulate this filter for step-size values $\{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$ required for the k-means clustering algorithm, and we show the best achieved results. The other 2 techniques we consider are based on fully-connected 2-layer feed-forward neural networks [see also the discussion in Section 3.1.1]. The first neural network uses the ReLu transfer function in the hidden layer, while the second network uses the Gaussian radial basis transfer function. Since BPSK detection is a 2-class classification problem, both networks use the softmax activation in the output layer. An important parameter for neural networks is the width (i.e., the number of neurons) of the hidden layer. We perform simulations with various layer sizes in the set $\{8, 16, 24, 32, 40, 48, 56, 64, 72, 80\}$, and we show the best achieved performance. Moreover, the networks are trained using state-of-art batch methods (in contrast to adaptive online training required by our application), and therefore the performance may not be achieved in practice.

We simulate two cases. In the first case, shown in Figure 3.10, user SNRs (in dBs) are equal to $\{10, 16, 13, 19, 22\}$ (in that order), while the angles of arrival are $\{50°, 70°, 60°, 40°, 30°\}$. In the second case, shown in Figure 3.11, we reduce the multi-access interference by increasing the angular separation to $\{90°, 150°, 120°, 60°, 30°\}$. In both cases the desired user is situated in the middle and it has the worst SNR. All results are an average of 100 experiments. We see that in both cases the proxy optimal filter achieves the best performance requiring the least number of samples. The largest performance gap is seen around the $T_{\text{train}} = 300$ mark. This clearly demonstrates the suitability of proxy optimal filter for real-time applications with a small number of samples.

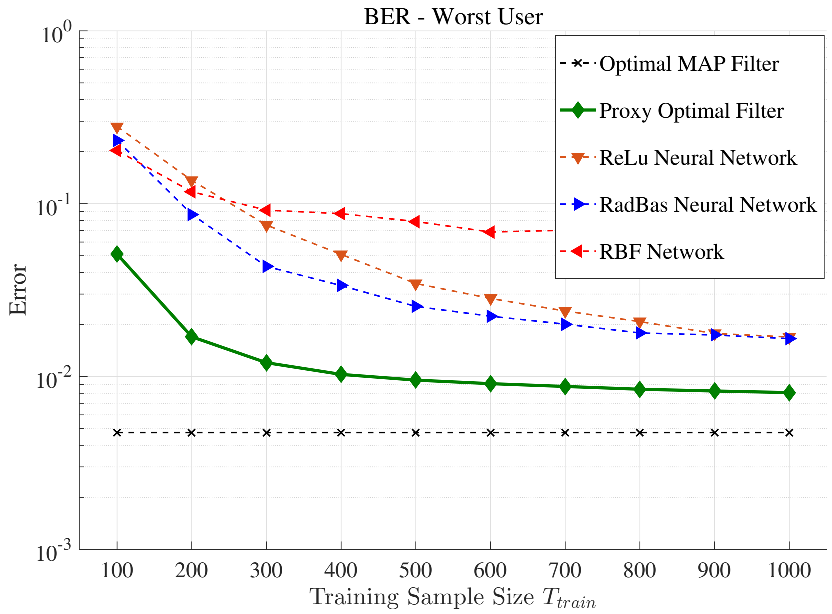


Figure 3.10: Comparison between different filtering techniques with an increasing training time T_{train} in terms of the BER of user 1.

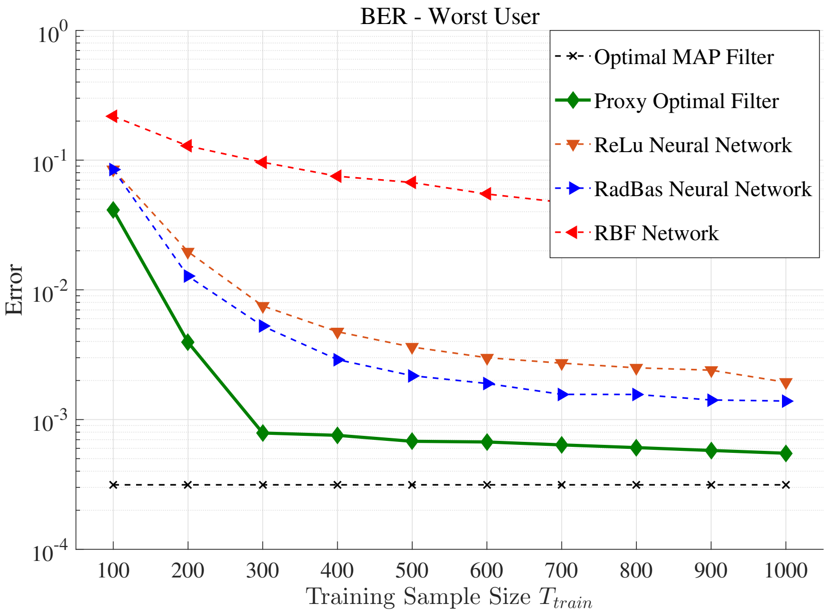


Figure 3.11: Comparison between different filtering techniques with an increasing training time T_{train} in terms of the BER of user 1.

3.5.2 Performance of the Robust Partially Linear Filter

In this section, we evaluate the performance of our proposed partially linear adaptive filter and its practical implementation presented in Section 3.4. In this case, our adaptive filter consists of a linear and a nonlinear component with kernel weights $w_{\text{L}} = 0.1$ and $w_{\text{G}} = 0.9$, respectively, fixed. Note that ideally $w_{\text{G}} \approx 1$, however a highly nonlinear filter does not react to a change in the environment well. To make the simulations more realistic than the setting considered in the last section, we consider QPSK modulation and Rayleigh block fading on the channels between users and the base station receiver. We use the algorithm in Algorithm 2 to perform online learning with dictionary sparsification. The learning is shown for $T_{\text{train}} = 2000$ channel symbols and the BER performance is evaluated in intervals of 100 training symbols. The dictionary novelty factor is set to $\alpha = 0.1$. The window size is fixed at $W = 50$ and we use $\epsilon = 0.1$. Note that in a real system, the training can be stopped at any point and data communication could be started. The training can then restart when the environment (e.g., the channel) has changed significantly. To simulate a real system, we change the channel independently at random after 500 symbols and then continue the training. All results are a uniform average of 100 experiments. For results in this section we show average BER of the 5 users.

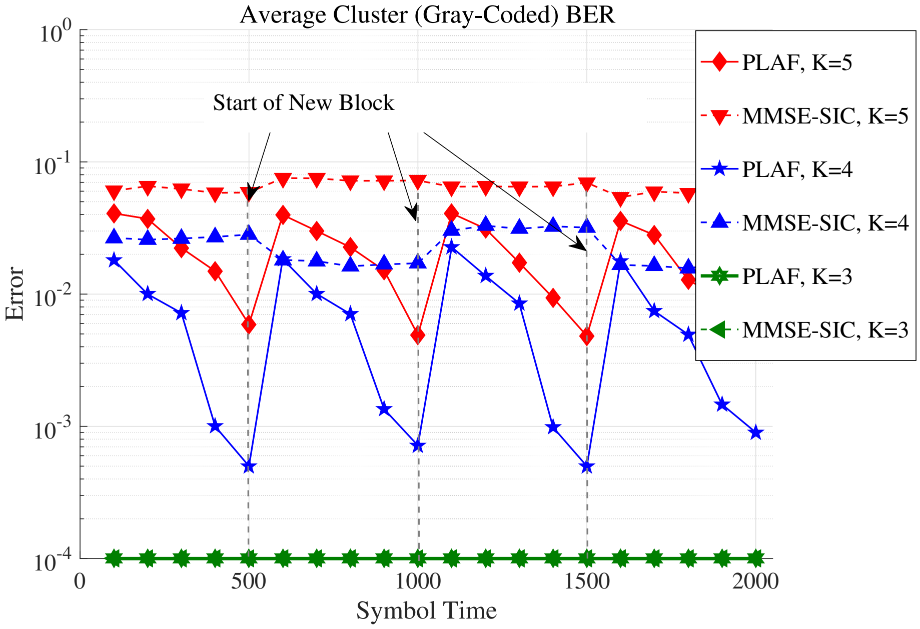


Figure 3.12: Performance Comparison: Comparison between the partially linear adaptive filter (PLAF) and the (symbol-level) MMSE-SIC filter for different number of users. The number of receive antennas is fixed at $M = 3$.

The first result in Figure 3.12 shows that the robust partially linear filter outperforms the nonlinear MMSE-SIC receiver (which we assume has very good estimations of user channels and the covariance matrix). We perform the comparison with nonlinear MMSE-SIC receiver because this receiver performs better than the linear MMSE and other linear receivers. Note that we do not compare the performance with the optimal MAP filter because in this section we only consider practical receivers and the optimal MAP filter is not implemented in practice. But it is clear that the optimal MAP filter should outperform the partially linear filter in a fixed static environment with full and perfect user knowledge. We fix the number of antennas at the receiver to $M = 3$ and simulate an increasing number of users $K \in \{3, 4, 5\}$. Note that as the number of users becomes larger than the number of receive antennas, we expect the system to become linearly inseparable.

The second result in Figure 3.13 compares the performance of the partially linear adaptive filter with that of the purely nonlinear filter (with $w_G = 1$). After a block of 500 symbols, the Rayleigh fading channel is changed to a new random independent value and a new set of active devices is selected randomly with probabilities $\rho \in \{0.7, 0.8, 1\}$. The training is continued after this change. The scenario models the dynamic environment of, for example, 5G massive machine-type communication systems. Our objective here is to

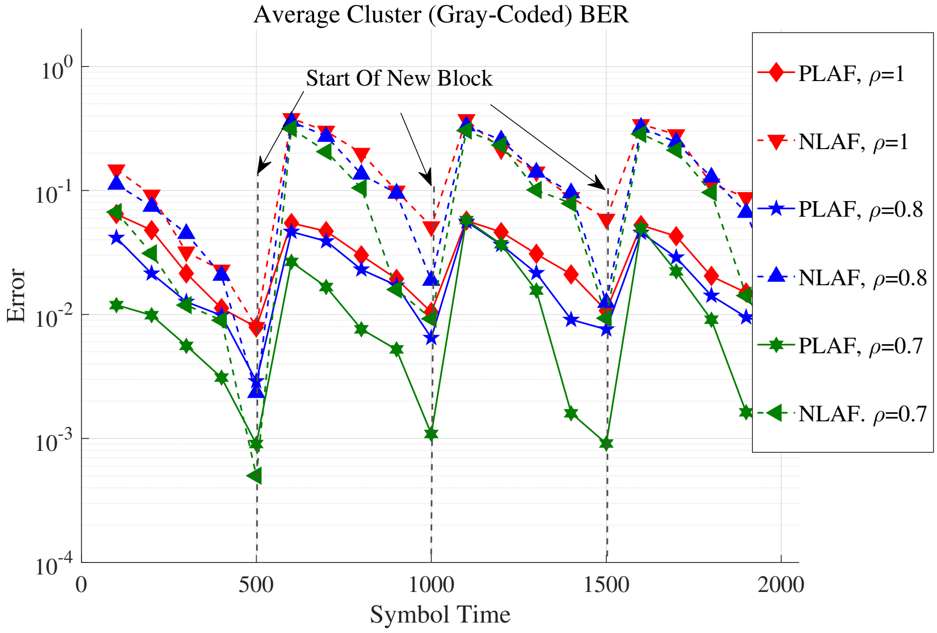


Figure 3.13: Performance Comparison: Comparison between the partially linear adaptive filter (PLAF) and the purely nonlinear adaptive filter (NLAF) for different user activation probabilities ρ. The number of receive antennas is fixed as $M = 3$.

show that the partially linear filter is robust against the changes in the environment. It can be seen that the "jumps" in the BER are smaller as compared to the purely nonlinear filter. Also note that the purely non filter requires larger number of training samples to reach the same BER level as compared to the partially linear filter. In other words, it adapts comparatively slower to the new environment. The reason is that the partially linear filter $f = f_{\mathrm{L}} + f_{\mathrm{G}}$ has a significant linear part while the purely nonlinear filter f_{G} has a nonlinear part only which requires more training to achieve the same BER performance. Of course for a sufficiently long training, the purely nonlinear filter will out perform the partially linear filter if the environment during this time remains fixed.

3.6 Supplementary Material and Proofs

3.6.1 Proof of Lemma 3.2

The output of the optimal MAP filter is given by

$$(\forall t \in \mathbb{N})\ f^{\star}(\mathbf{r}(t)) := \sum_{q=1}^{N_{\text{mod}}} v_q \exp\left(-\frac{\|\mathbf{r}(t)-\bar{\mathbf{r}}_q\|^2}{2\sigma_n^2}\right),$$

where

$$v_q = \frac{\operatorname{sgn}(b_{q_1})}{N_{\text{mod}}(2\pi\sigma_n^2)^M} = \frac{b_{q_1}}{N_{\text{mod}}(2\pi\sigma_n^2)^M}.$$

Proof. We have $(\forall t \in \mathbb{N})$

$$|f^{\star}(\mathbf{r}(t))| \leq \left|\sum_{q=1}^{N_{\text{mod}}} \frac{b_{q_1}}{N_{\text{mod}}(2\pi\sigma_n^2)^M} \exp\left(-\frac{\|\mathbf{r}(t)-\bar{\mathbf{r}}_q\|^2}{2\sigma_n^2}\right)\right|,$$

and

$$\left|\ f^{\star}(\mathbf{r}(t)) - \frac{b_1(t)}{N_{\text{mod}}(2\pi\sigma_n^2)^M}\ \right| \leq \left|\ \sum_{q=1}^{N_{\text{mod}}} \frac{b_{q_1}}{N_{\text{mod}}(2\pi\sigma_n^2)^M} \exp\left(-\frac{\|\mathbf{r}(t)-\bar{\mathbf{r}}_q\|^2}{2\sigma_n^2}\right) - \frac{b_1(t)}{N_{\text{mod}}(2\pi\sigma_n^2)^M}\ \right|.$$

Note that for every $t \in \mathbb{N}$ there exists $q \in \overline{1, N_{\text{mod}}}$ such that $b_{q_1} = b_1(t)$ and $\bar{\mathbf{r}}_q + \mathbf{n}(t) = \mathbf{r}(t)$. Then separating the qth center from the remaining centers we have $(\forall t \in \mathbb{N})$

$$\begin{aligned}\left|\ f^{\star}(\mathbf{r}(t)) - \frac{b_1(t)}{N_{\text{mod}}(2\pi\sigma_n^2)^M}\ \right| &\leq \left|\frac{b_1(t)}{N_{\text{mod}}(2\pi\sigma_n^2)^M} \exp\left(-\frac{\|\mathbf{r}(t)-\bar{\mathbf{r}}_q\|^2}{2\sigma_n^2}\right) - \frac{b_1(t)}{N_{\text{mod}}(2\pi\sigma_n^2)^M}\ \right| \\ &+ \left|\ \sum_{p=1, p\neq q}^{N_{\text{mod}}} \frac{b_{p_1}}{N_{\text{mod}}(2\pi\sigma_n^2)^M} \exp\left(-\frac{\|\mathbf{r}(t)-\bar{\mathbf{r}}_p\|^2}{2\sigma_n^2}\right)\ \right|.\end{aligned}$$

Leaving out the scalar normalization $N_{\text{mod}}(2\pi\sigma_n^2)^M$ and using the triangular inequality we get

$$\begin{aligned}(\forall t \in \mathbb{N})\ \mid f^{\star}(\mathbf{r}(t)) - b_1(t) \mid &\leq \underbrace{\left|\ \exp\left(-\frac{\|\mathbf{r}(t)-\bar{\mathbf{r}}_q\|^2}{2\sigma_n^2}\right) - 1\ \right|}_{\text{noise term}} \\ &+ \underbrace{\left|\ \sum_{p=1, p\neq q}^{N_{\text{mod}}} \exp\left(-\frac{\|\mathbf{r}(t)-\bar{\mathbf{r}}_p\|^2}{2\sigma_n^2}\right)\ \right|}_{\text{interference term}}.\end{aligned} \tag{3.30}$$

Since $(\forall t \in \mathbb{N})\ \|\mathbf{r}(t) - \bar{\mathbf{r}}_q\| = \|\mathbf{n}(t)\| \leq W_{\text{noise}}$ by Assumption 3.12, for the noise term above we have

$$\begin{aligned}(\forall t \in \mathbb{N})\ \epsilon_1 &= \left| \exp\left(-\frac{W_{\text{noise}}^2}{2\sigma_n^2}\right) - 1 \right| \\ &\geq \left| \exp\left(-\frac{\|\mathbf{r}(t) - \bar{\mathbf{r}}_q\|^2}{2\sigma_n^2}\right) - 1 \right|.\end{aligned}$$

Now suppose that $(\forall p, q \in \overline{1, N_{\text{mod}}})\ (p \neq q)\ \frac{\|\bar{\mathbf{r}}_q - \bar{\mathbf{r}}_p\|}{2} := \alpha_{\text{disparity}}^{q,p} \geq W_{\text{noise}}$, then due to Assumption 3.12 $(\forall t \in \mathbb{N})\ \|\bar{\mathbf{r}}_q - \bar{\mathbf{r}}_p\| \geq \alpha_{\text{disparity}}^{q,p} + \|\mathbf{n}(t)\| \implies \|\bar{\mathbf{r}}_q - \bar{\mathbf{r}}_p\| - \|\mathbf{n}(t)\| \geq \alpha_{\text{disparity}}^{q,p}$.

Due to triangular inequality it can be verified that $\|\bar{\mathbf{r}}_q - \bar{\mathbf{r}}_p + \mathbf{n}(t)\| \geq \|\bar{\mathbf{r}}_q - \bar{\mathbf{r}}_p\| - \|\mathbf{n}(t)\|$ which implies that $\|\mathbf{r}(t) - \bar{\mathbf{r}}_p\| \geq \alpha_{\text{disparity}}^{q,p}$ for every term in the interference term in (3.30). We have

$$\left| \sum_{p=1, p\neq q}^{N_{\text{mod}}} \exp\left(-\frac{(\alpha_{\text{disparity}}^{q,p})^2}{2\sigma_n^2}\right) \right| \geq \left| \sum_{p=1, p\neq q}^{N_{\text{mod}}} \exp\left(-\frac{\|\mathbf{r}(t) - \bar{\mathbf{r}}_p\|^2}{2\sigma_n^2}\right) \right|$$

for the interference term in (3.30). If we define the worst case bound as

$$\epsilon_2 := \max_{q \in \overline{1, N_{\text{mod}}}} \left| \sum_{p=1, p\neq q}^{N_{\text{mod}}} \exp\left(-\frac{(\alpha_{\text{disparity}}^{p,q})^2}{2\sigma_n^2}\right) \right|,$$

then for the interference part in (3.30) we get the desired bound

$$(\forall t \in \mathbb{N})\ \epsilon_2 \geq \left| \sum_{p=1, p\neq q}^{N_{\text{mod}}} \exp\left(-\frac{\|\mathbf{r}(t) - \bar{\mathbf{r}}_p\|^2}{2\sigma_n^2}\right) \right|.$$

□

3.6.2 Proof of Theorem 3.2

Let $(\Theta_t)_{t\in\mathbb{N}}$, $\Theta_t : \mathcal{H} \to \mathbb{R}$, be a sequence of continuous convex functions. For an arbitrary $f_1 \in \mathcal{H}$, the sequence $(f_t)_{t\in\mathbb{N}} \subset \mathcal{H}$ of the APSM algorithm is given by [YO05]

$$f_{t+1} := \begin{cases} f_t - \frac{\Theta_t(f_t)}{\|\Theta_t'(f_t)\|^2}\Theta_t'(f_t), & \text{if } \Theta_t'(f_t) \neq 0, \\ f_t, & \text{otherwise.} \end{cases} \tag{3.31}$$

Now, define $(\forall t \in \mathbb{N})\ \Theta_t(\cdot) := \sum_{j \in \mathcal{J}_t} q_j^t\ d(\cdot, C_j)$, where $(\forall f \in \mathcal{H})\ (\forall C_t \subset \mathcal{H})\ d(f, C_t) := \inf_{y \in C_t} \|y - f\|_{\mathcal{H}} = \|f - \mathbf{P}_{C(t)}(f)\|_{\mathcal{H}}$, and where $\mathcal{J}_t \subset \mathbb{N}$ are the indices of the sets in $(C_i)_{i \in \overline{1,t}}$ that we intend to process at iteration/time t. The functions $\Theta_t(\cdot)$ are continuous and convex, and a subgradient is given by

$$\Theta_t'(f) := \sum_{j \in \mathcal{J}_t} q_j^t d'(f, C_j), \tag{3.32}$$

where

$$d'(f, C_j) := \begin{cases} \frac{f - \mathbf{P}_{C_j}(f)}{\left\| f - \mathbf{P}_{C_j}(f) \right\|_{\mathcal{H}}}, & \text{if } f \notin C_j, \\ 0, & \text{otherwise.} \end{cases} \tag{3.33}$$

Replacing $d'(f, C_j)$ in (3.32) by (3.33), we get

$$\Theta_t'(f) := \frac{\sum_{j \in \mathcal{J}_t} q_j^t (f - \mathbf{P}_{C_j}(f))}{\Theta_t(f)}$$

Finally, by replacing $\Theta_t(f_t)$ and $\Theta_t'(f_t)$ in (3.31), we get the iteration (3.22). We are now in the position of proving the claim in Theorem 3.2.

Proof. Recall from Section 1.6 that, because C_t is convex, the projection $P_{C_t}(f)$ is the minimizer of $d(f, C_t)$. Then, $(\forall f \in \bigcap_{j \in \mathcal{J}_t} C_j)\ \Theta_t'(f) = 0$. In other words, the set $\Omega_t := \bigcap_{j \in \mathcal{J}_t} C_j$ is the set of minimizers of Θ_t. Clearly, $\Theta_t(f) = 0$ for any $f \in \bigcap_{j \in \mathcal{J}_t} C_j$. Now since $(\forall t \in \mathbb{N})\ f^\star \in \Omega_t$ (because $(\forall t \in \mathbb{N})\ f^\star \in C_t$) this implies that $f^\star \in \bigcap_{t \in \mathbb{N}} \Omega_t$ so $\bigcap_{t \in \mathbb{N}} \Omega_t \neq \emptyset$. Now since all required conditions in [YO05, Theorem 2((a) and (b))] have been fulfilled, the claim follows. □

3.6.3 Derivation of the Projections onto the Subspace spanned by the Nonlinear Dictionary

Let $\mathcal{D}_{\mathrm{G},n-1}$ denote the Gaussian dictionary at time index $n-1$ and let $S_{n-1} := |\mathcal{D}_{\mathrm{G},n-1}|$ denote its cardinality. Denote by $\mathbf{\Psi}_l^{n-1} \in \mathcal{D}_{G,n-1}$ the lth element of $\mathcal{D}_{\mathrm{G},n-1}$. We denote by $\mathbf{K}_{n-1} \in \mathbb{R}^{S_{n-1} \times S_{n-1}}$ the standard *Gram* matrix at time $n-1$, with its ith row and the jth column given by

$$(\forall i \in \overline{1, S_{n-1}})\ (\forall j \in \overline{1, S_{n-1}})\ \mathbf{K}_{n-1} := \left\langle \mathbf{\Psi}_i^{n-1}, \mathbf{\Psi}_j^{n-1} \right\rangle_{\mathcal{H}_{\mathrm{G}}}.$$

Note that $\mathbf{K}_{n-1}$ is positive definite because the elements $(\boldsymbol{\Psi}_l^{n-1})_{l\in\overline{1,S_{n-1}}}$ of the Gaussian dictionary $\mathcal{D}_{\mathrm{G},n-1}$ are linearly independent by assumption (see Section 3.4.5). As a result, the inverse $\mathbf{K}_{n-1}^{-1}$ exists.

The projection $\kappa_{\mathrm{G}}(\mathbf{y}_n,\cdot)$ on the linear closed subspace $\mathcal{H}_{\mathrm{G},n-1}\subset\mathcal{H}_{\mathrm{G}}$ spanned by $\mathcal{D}_{\mathrm{G},n-1}$ is given by [ST08]

$$\mathbf{P}_{\mathcal{H}_{\mathrm{G},n-1}}(\kappa(\mathbf{y}_n,\cdot))=\sum_{l=1}^{S_{n-1}}\boldsymbol{\zeta}_{\mathbf{y}_n,l}^{n}\boldsymbol{\Psi}_l^{n-1},$$

where $\boldsymbol{\zeta}_{\mathbf{y}_n}^n\in\mathbb{R}^{S_{n-1}}$ is given by $\boldsymbol{\zeta}_{\mathbf{y}_n}^n=\mathbf{K}_{n-1}^{-1}\boldsymbol{\xi}_{\mathbf{y}_n}^n$; the vector $\boldsymbol{\xi}_{\mathbf{y}_n}^n$ is given as

$$\boldsymbol{\xi}_{\mathbf{y}_n}^n=\begin{bmatrix}\left\langle\kappa_{\mathrm{G}}(\mathbf{y}_n,\cdot),\boldsymbol{\Psi}_1^{n-1}\right\rangle_{\mathcal{H}_{\mathrm{G}}}\\ \vdots\\ \left\langle\kappa_{\mathrm{G}}(\mathbf{y}_n,\cdot),\boldsymbol{\Psi}_{S_{n-1}}^{n-1}\right\rangle_{\mathcal{H}_{\mathrm{G}}}\end{bmatrix}.$$

Suppose now that $\mathbf{K}_{n-1}^{-1}$ is given, then the distance of $\kappa_{\mathrm{G}}(\mathbf{y}_n,\cdot)$ from $\mathcal{D}_{\mathrm{G},n-1}$ is the solution to [ST08]

$$d_n^2:=\kappa_{\mathrm{G}}(\mathbf{y}_n,\mathbf{y}_n)-(\boldsymbol{\xi}_{\mathbf{y}_n}^n)^{\intercal}\boldsymbol{\zeta}_{\mathbf{y}_n}^n.$$

Given d_n, $\boldsymbol{\xi}_{\mathbf{y}_n}^n$, and $\boldsymbol{\zeta}_{\mathbf{y}_n}^n$ the inverse $\mathbf{K}_{n-1}^{-1}$ is updated for the next iteration n to $\mathbf{K}_n^{-1}$ which further enables us to calculate $\boldsymbol{\xi}_{\mathbf{y}_{n+1}}^{n+1}$, $\boldsymbol{\zeta}_{\mathbf{y}_{n+1}}^{n+1}$, and d_{n+1} in that order. In more detail, we initialize the inverse by $\mathbf{K}_1^{-1}:=1/\kappa_{\mathrm{G}}(\mathbf{y}_1,\mathbf{y}_1)$. For $n\geq 2$ if $\kappa_{\mathrm{G}}(\mathbf{y}_n,\cdot)$ is admitted to the dictionary, i.e., if $\mathcal{D}_{G,n}=\mathcal{D}_{G,n-1}\cup\{\kappa_{\mathrm{G}}(\mathbf{y}_n,\cdot)\}$, then

$$\mathbf{K}_n^{-1}:=\begin{bmatrix}\mathbf{K}_{n-1}^{-1}+\frac{\boldsymbol{\zeta}_{\mathbf{y}_n}^n(\boldsymbol{\zeta}_{\mathbf{y}_n}^n)^{\intercal}}{d_n^2} & -\frac{\boldsymbol{\zeta}_{\mathbf{y}_n}^n}{d_n^2}\\ -\frac{(\boldsymbol{\zeta}_{\mathbf{y}_n}^n)^{\intercal}}{d_n^2} & \frac{1}{d_n^2}\end{bmatrix},$$

otherwise $\mathbf{K}_n^{-1}:=\mathbf{K}_{n-1}^{-1}$.

Now we look at how to calculate $\mathbf{P}_{\mathcal{H}_{G,n}}(\kappa(\mathbf{y}_j,\cdot))$ for each $j\in\mathcal{J}_n$. We start by considering the latest training sample $j=n$. If $\kappa_{\mathrm{G}}(\mathbf{y}_n,\cdot)\in\mathcal{D}_{G,n}$ then obviously $\mathbf{P}_{\mathcal{H}_{G,n}}(\kappa_{\mathrm{G}}(\mathbf{y}_n,\cdot))=\kappa_{\mathrm{G}}(\mathbf{y}_n,\cdot)$, otherwise $\mathbf{P}_{\mathcal{H}_{G,n}}(\kappa_{\mathrm{G}}(\mathbf{y}_n,\cdot))=\mathbf{P}_{\mathcal{H}_{G,n-1}}(\kappa_{\mathrm{G}}(\mathbf{y}_n,\cdot))$. The projection $\mathbf{P}_{\mathcal{H}_{G,n-1}}(\kappa_{\mathrm{G}}(\mathbf{y}_j,\cdot))$ in (3.6.3) is already available to us because $\boldsymbol{\zeta}_{\mathbf{y}_n}^n$ and $\mathcal{D}_{G,n-1}$ are both known to us from the dictionary update step (see Section 3.4.5 and Algorithm 2). It follows that $(\forall n\in\mathbb{Z}_{\geq 0})$ $(\forall j\in\mathcal{J}_n)$, either $\mathbf{P}_{\mathcal{H}_{G,n}}(\kappa_{\mathrm{G}}(\mathbf{y}_j,\cdot))=\kappa_{\mathrm{G}}(\mathbf{y}_j,\cdot)$ or $\mathbf{P}_{\mathcal{H}_{G,n}}(\kappa_{\mathrm{G}}(\mathbf{y}_j,\cdot))=\mathbf{P}_{\mathcal{H}_{G,n-1}}(\kappa_{\mathrm{G}}(\mathbf{y}_j,\cdot))$.

4 Robust Approximation of Probability Density Functions

4.1 Introduction

An important problem in unsupervised learning, adaptive signal processing, Bayesian approximation, and many other engineering fields is the approximation of probability density functions (pdfs). In the first part of this chapter we present a method that uses prior knowledge and training sample sets to approximate pdfs. In contrast to many existing techniques that use well-known loss functions, our proposed method is based on the set-membership or set-theoretic paradigm [Com93]; it is particularly suited to applications where a large sample set is not available. Briefly, we incorporate all available information about the underlying pdf in the form of closed convex sets in a Hilbert space. The intersection of all of these information sets forms a feasible solution set, which contains valid and reasonable solutions to the pdf approximation problem. In more detail, recall that pdfs are well-behaved functions in the sense that they must be nonnegative and they must integrate to unity. We incorporate these two properties as prior knowledge in our framework, and we approximate the pdf of a random source $\mathbf{X}$ given a training sample set $\mathcal{D}_{\mathbf{X}} := \{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}$ consisting of i.i.d. realizations of $\mathbf{X}$. Our iterative approximation method is based on the POCS technique which means that our method is numerically robust with convergence guarantees. Furthermore, the involved projections on convex sets can be performed in parallel to increase speed. Results show that our proposed method can work well with relatively small number of samples, and therefore it is suited to dynamic wireless scenarios where large sample sets are not available.

In the second part of this chapter we apply our approximation method to approximation of likelihood functions in a cloud-radio access network (CRAN) to perform distributed multiuser detection/demodulation. CRAN is envisaged to be a key enabler of cell-less[1] uplink because of its low cost and spectrum efficiency [TTQL17, CCY+15]. In conventional CRAN joint baseband processing at centralized cloud processors or a central unit is performed on behalf of distributed remote radio heads (RRHs). This migration of pro-

[1] In a cell-less system a user transmits to multiple (generally close-by) base stations simultaneously.

cessing is made possible by deploying fronthaul links between RRHs and the central unit [see Figure 4.2]. Under the assumption of high-capacity fronthaul links, the cost reduction by using low-complexity RRHs is complemented by performance benefits emanating from joint detection/processing at the central unit [NAY$^+$17]. The conventional CRAN with the common public radio interface (CPRI) specification prescribes simple scalar quantization for fronthaul links, but the performance of this approach degrades in the presence of stringent fronthaul capacity constraints [PSSS14]. It has been shown that, in contrast to conventional CRAN, it maybe advantageous to apply pre-processing (which in our case consists of detection using a multiuser receive filter) locally at the RRHs followed by fusion and maximum likelihood-based decision at the CU [USDP17]. Following this idea, we develop a learning-based "detect and forward" scheme, whereby the likelihood ratios associated with local detection are combined at the CU to obtain the final estimate. We note that, whereas in some Bayesian detection techniques the likelihood information may come naturally, in non-Bayesian methods considered here, this is not the case. In this way, existing detection methods (e.g., the one developed in Chapter 3) can be extended to the cell-less CRAN setting.

4.1.1 PDFs and Likelihoods

Let $(\mathcal{W}, \mathcal{F}, \mathbb{P})$ be a probability space, with $\mathcal{W}$ the sample space, $\mathcal{F}$ the space of all events, and $\mathbb{P} : \mathcal{F} \to [0,1]$ the probability measure. We denote by $f_{\mathbf{X}} : \mathbb{R} \to \mathbb{R}$, the Lebesgue-integrable pdf of a continuous real-valued random variable $\mathbf{X} : \mathcal{W} \to \mathbb{R}$. Then, the probability that $\mathbf{X}(\omega) \in \mathcal{A}$, where $\mathcal{A} \subset \mathbb{R}$, is given by

$$\mathbb{P}[\ \mathbf{X} \in \mathcal{A}\ ] := \mathbb{P}\{\mathbf{X} \in \mathcal{A}\} = \int_{\mathcal{A}} f_{\mathbf{X}}(x)dx,$$

where here and in the remainder we use the universal shorthand $\{\mathbf{X} \in \mathcal{A}\} := \{\omega \in \mathcal{W} : \mathbf{X}(\omega) \in \mathcal{A}\}$.

Likelihood functions serve as the basis for the classical maximum likelihood estimation method and they are also an important part of general Bayesian inference. In more detail, we use the following interpretation of Likelihood functions as parametrized pdfs when the data is generated by a real and continuous random source $\mathbf{X}$ that depends on a parameter b. Suppose b has a uniform (or an unknown) distribution and $\mathbf{X} = x \in \mathbb{R}$ is a given realization, then

$$\mathcal{L}(b; x) := \varphi_{\mathbf{X}}(x|b) \propto \varphi(b|x)$$

is the likelihood of the data x being generated by $\mathbf{X}$ with parameter b fixed, where $\varphi_{\mathbf{X}}(\cdot|b) : \mathbb{R} \to \mathbb{R}$ is a function of x when the parameter b is held fixed, and where $\varphi(b|\cdot) : \mathbb{R} \to \mathbb{R}$ is the posterior density function for b. We can view $\varphi_{\mathbf{X}}(\cdot|b)$ as the likelihood distribution of $\mathbf{X}$ when b is held fixed.

4.1.2 Related Work

The approximation of pdfs is a well-studied topic. Here we discuss two classical and widely used algorithms for density approximation. First, we discuss the classical kernel density estimation method which is a (quasi) nonparametric method. Kernel density estimators are by far the most popular and well-studied methods. Their mathematical properties are well-understood, and due to their nonparametric nature, they tend to be more flexible than their parametric counterparts [BGK10].

Suppose we are given a univariate data set $\mathcal{D} := \{x_1, \ldots, x_N\} \subset \mathbb{R}$ of i.i.d. samples of a random source $\mathbf{X}$ with an unknown density $f^*_{\mathbf{X}} : \mathbb{R} \to [0, 1]$. The *Parzen-Rosenblatt* kernel density estimator of $f^*_{\mathbf{X}}$ is given by

$$f^h_{\mathbf{X}} = \frac{1}{Nh} \sum_{i=1}^{N} \kappa^h(x, x_i), \tag{4.1}$$

where $\kappa^h : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is referred to as the kernel (function) which satisfies certain properties [Sil86, BGK10, Sch11], and the parameter $h \in \mathbb{R}$ is the bandwidth parameter. A well-known kernel is the Gaussian kernel

$$(\forall x \in \mathbb{R})(\forall y \in \mathbb{R})\ \kappa^\sigma(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-|x - y|^2}{2\sigma^2}\right).$$

The Gaussian kernel is widely used due to its inherent smoothness, convenient mathematical properties, and its universal approximation property [MXZ06]. Note that there also exist many other kernels with good approximation behavior. The differences among various kernels are not significant in terms of the quality of approximation, and therefore a particular kernel is mainly chosen based on its convenient mathematical properties such as its smoothness [WJ94]. Once a kernel has been chosen, the approximation of $f^*_{\mathbf{X}}$ boils down to the choice of the kernel bandwidth parameter h which has a significant effect on the performance [Sil86]. Unfortunately, there exists no universally accepted procedure to select h [Sch11]. In more detail, there exist two popular criteria to judge how well $f^h_{\mathbf{X}}$ approximates $f^*_{\mathbf{X}}$, namely, the integrated squared error and the mean integrated squared error [note: the plugin estimators [SJ91] are based on the second criteria and they are preferred in practice to cross-validation methods that consider the integrated squared error]. However, both these criteria involve the unknown $f^*_{\mathbf{X}}$ or its derivatives, so only data-driven methods can be used to estimate h. If one assumes that $f^*_{\mathbf{X}}$ is normal and a Gaussian kernel is used in (4.1), then *Silverman's* rule of thumb bandwidth selector is optimal in terms of mean integrated squared error. A survey of bandwidth selection can be found in [Sch11].

Note that once h and the kernel are fixed in (4.1), kernel density estimator is nonparametric,[2] as no further adaptation of parameters is required and the number of kernel terms increases with sample data. The resulting approximation is of a low complexity but it may require a large number of samples to achieve acceptable performance [Fro13]. In more detail, most (if not all) of the bandwidth estimation methods show that the optimal bandwidth h is inversely proportional to N. So for a small number of samples, h should be relatively large which could be problematic if $f^*_{\mathbf{X}}$ has significant "local features" and $f^h_{\mathbf{X}}$ is oversmoothed. There exists other well-known problems with kernel density estimators in the case of relatively small sample sets. In particular, they tend to work poorly when the support of the underlying pdf is compact such as in the case of the uniform distribution. This is due to absence of sufficient sample data around the boundaries so that kernel density estimators penalize this lack of data and the approximation decays fast near the boundary. These problems are significant enough to warrant a large body of research work that has tried to alleviate them. Unfortunately, most of the these methods tend to either require a large amount of sample data, or they do not produce "proper" density functions in the sense that the approximations may not be nonnegative or they may not integrate to unity [BGK10]. To the best of our knowledge, one of the most popular and widely used kernel density estimators is the one studied in [BGK10]. This density estimator has low complexity and it also deals with the boundary problem well if the support of the underlying pdf is known. Significantly, this plug-in method also produces a good estimate of the bandwidth which works well in practice, and in contrast to previous methods, it does not require any assumptions regarding the underlying pdf.

Nonparametric models are generally employed when no information, e.g., the shape or functional form, can be assumed about the underlying pdf and therefore the approximation needs to be sufficiently flexible. However, in some cases, especially when the sample set is small, it is useful to assume that the underlying pdf $f^*_{\mathbf{X}}$ can be written as a sum of individual parametrized component pdfs, i.e., as a finite mixture model. More precisely, one assumes that

$$f^*_{\mathbf{X}} \in \mathcal{G} := \left\{ f : f = \sum_{i=1}^{M} w_i \ f_i(\cdot;\theta_i), w_i \geq 0, \theta_i \in \mathbb{R}^n, \sum_{i=1}^{M} \ w_i = 1 \right\},$$

[2] By nonparametric models one means that the number of parameters to be learned is not fixed, and it generally increases with the number of samples.

where M is a fixed number of mixture components, f_i are component densities, and θ_i are parameters of f_i. The question now becomes, first, how to choose the components f_i and, second, whether there exists a low-complexity estimation of the parameters θ_i. In absence of any information on the underlying $f^*_{\mathbf{X}}$, the choice of f_i should be such that their mixture can approximate an arbitrary $f^*_{\mathbf{X}}$ sufficiently well. A popular choice are the Gaussian mixtures, i.e.,

$$f^*_{\mathbf{X}} \in \mathcal{G} := \left\{ f : f = \sum_{i=1}^{M} w_i \ \kappa^{\sigma_i}(\cdot, x_i), \sigma_i > 0, w_i \geq 0, x_i \in \mathbb{R}, \sum_{i=1}^{M} \ w_i = 1 \right\};$$

$\kappa^{\sigma_i}(x, x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-|x-x_i|^2}{2\sigma_i^2}\right)$; w_i, σ_i, and x_i are parameters to be determined. Gaussian mixtures can approximate any $L^2(\mathbb{R})$ density sufficiently well if the number of samples becomes sufficiently large and all σ_i become equal and vanish [PH00]. Due to their popularity, Gaussian mixtures have been well-studied and over the years many methods have been developed to estimate the model parameters. Among these, maximum-likelihood estimation of the parameters is the most popular technique due to its computational efficiency. The expectation-maximization (EM) algorithm is the dominant method to learn Gaussian mixtures and it has a relatively low complexity. However, the quality of solution provided by this algorithm is unknown as the algorithm obtains a local maxima of the likelihood function with high probability; especially for small sample sizes the likelihood function has many local maxima and the log-likelihood value at these points can be arbitrary worse than the global value [see, e.g., [JZB$^+$16]].

4.1.3 Contribution

Our method, detailed in Section 4.2, can be seen as a hybrid between kernel density estimation and Gaussian mixture model but it alleviates the problems encountered by both methods. In more detail, a limitation of kernel density estimators is that every kernel function centered at the samples x_i is equally weighted by the factor $\frac{1}{Nh}$ in (4.1), which means that the weights are nonadaptive. Kernel density estimators do not make use of samples of $f^*_{\mathbf{X}}$ which, as we show in Section 4.2, can be extracted from the sample set $\mathcal{D} := \{x_1, \ldots, x_N\} \subset \mathbb{R}$ using statistical methods. As a result, kernel density estimators lack local adaptivity. Adaptive kernel estimators use a variable bandwidth [each kernel term in (4.1) has its own bandwidth] but these methods are either too complex or they may not produce proper pdf approximations [BGK10]. We propose to use the low-complexity bandwidth estimator developed in [BGK10] that works well in practice, but any reliable method can be used to fix the bandwidth in our framework. Following the bounded error estimation/set-theoretic estimation theory [see Section 1.7], i.e., we construct a feasible

Table 4.1: List of Variables

Description	**Symbol**
Probability of an event $\mathcal{A}$	$\mathbb{P}[\ \mathcal{A}\ ]$
Projection of f onto a set C	$\mathbf{P}_C(f)$
Number of RRHs	$R \in \mathbb{N}$
Number of users	$K \in \mathbb{N}$
Number of sample sets	$Q \in \mathbb{N}$
Received signal at lth RRH	$\mathbf{r}^l(t) \in \mathbb{R}^2M$
Filter at lth RRH	$f_l : \mathbb{R}^2M \to \mathbb{R}$
Likelihood funcion at lth RRH	$\varphi^l(\cdot\|b) : \mathbb{R} \to \mathbb{R}$
Likelihood at lth RRH	$\mathcal{L}_l(\cdot;\cdot)$
Sample set for random variable $\mathbf{X}$	$\mathcal{D}_{\mathbf{X}} \subset \mathbb{R}$
Size of sample set	$N \in \mathbb{N}$
Space of square-integrable function	$L^2 := L^2(\mathbb{R})$
Gaussian mixture set	$\mathcal{G} \subset L^2$
Gram matrix	$\mathbf{G} \in \mathbb{R}^N_{\geq 0} \times \mathbb{R}^N_{\geq 0}$
Channel coherence time	$T_{\text{block}} \in \mathbb{Z}_{>0}$
Training time	$T_{\text{train}} \in \mathbb{Z}_{>0}$

solution set consisting of certain Gaussian mixtures that agree with the prior information about $f^*_{\mathbf{X}}$ and the information extracted from a given sample set. We consider any point in the feasible solution set as a valid and reasonable approximation of $f^*_{\mathbf{X}}$. The algorithmic framework is robust, it solves a convex problem, and it provides convergence guarantees in contrast to, e.g., the EM method. Furthermore, it works well with relatively small sample sets and deals with the aforementioned boundary problem in an efficient way without requiring an explicit knowledge of the support of the underlying pdf. The resulting approximation is a proper pdf, and its evaluation in real-time applications has low complexity.

In Section 4.3 we apply our pdf approximation method to approximate likelihood functions in a cloud-radio access network (CRAN) to perform distributed multiuser detection/demodulation.

4.2 Set-Theoretic Density Approximation

In this section we present a low-complexity technique for obtaining a reliable approximation of the pdf of a random source $\mathbf{X}$ given an i.i.d. sample set. We denote the pdf of a random source $\mathbf{X}$ by $\varphi_{\mathbf{X}}$, and perform a set-theoretic approximation of $\varphi_{\mathbf{X}}$ based on closed convex information sets.

The information sets are constructed from:

1. Prior knowledge about general properties of pdfs.
2. A sample set

$$\mathcal{D}_{\mathbf{X}} := \{x_1, x_2, \dots, x_N\} \subset \mathbb{R},$$

   which we assume to consist of i.i.d observations of $\mathbf{X}$.

We start by assuming that $\varphi_{\mathbf{X}} \in L^2(\mathbb{R})$, where $L^2(\mathbb{R})$ (henceforth denoted by simply L^2) is the Hilbert space of square Lebesgue-integrable functions equipped with the inner-product

$$(\forall f \in L^2)\ (\forall g \in L^2)\ \langle g, f\rangle_{L^2} := \int_{\mathbb{R}} g(x)f(x)dx, \tag{4.2}$$

and the norm

$$\|f\|_{L^2}^2 = \langle f, f\rangle_{L^2} < \infty.$$

Given the samples $(x_i)_{i\in\overline{1,N}} \subset \mathbb{R}$, and the bandwidth $\sigma > 0$, we assume that $\varphi_{\mathbf{X}}$ belongs to a closed subspace $\mathcal{G} \subset L^2$ defined as follows. Consider the Gaussian function

$$(\forall x \in \mathbb{R})\ (\forall y \in \mathbb{R})\ \kappa(x,y) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-|x-y|^2}{2\sigma^2}\right),$$

with $\sigma > 0$, and now define

$$\begin{aligned} \mathcal{G} &:= \operatorname{span}\{\kappa(\cdot, x_1), \dots, \kappa(\cdot, x_N)\} \\ &= \left\{\varphi \in L^2\ :\ \varphi = \sum_{i=1}^{N} w_i\ \kappa(\cdot, x_i), N \in \mathbb{Z}_{\geq 0}, (\forall i \in \overline{1,N})\ w_i \in \mathbb{R}\}\right\}. \end{aligned} \tag{4.3}$$

To give $\mathcal{G}$ a Hilbert space structure, we equip it with the inner-product $\langle h, p\rangle_{\mathcal{G}} = \langle h, p\rangle_{L^2}$ and the norm $\|f\|_{\mathcal{G}}^2 = \langle f, f\rangle_{\mathcal{G}}$. With this construction, it is well-known that $\mathcal{G}$ is a finite-dimensional Hilbert subspace of L^2 [SSM98, OY17].

There are several reasons for working in the subspace $\mathcal{G} \subset L^2$:

1. Note that $\mathcal{G}$ can be considered as a space of certain N-component Gaussian mixtures. Gaussian mixtures are well-known for their ability to approximate well any L^2 density function [PH00].
2. In contrast to [SYY98, Ch. 6.5], rather than estimating the value $f(x) \in \mathbb{R}$, for each real-time input $x \in \mathbb{R}$, we approximate the pdf f uniformly in $\mathcal{G}$. In this way, we require to run the approximation algorithm only once after the sample set $\mathcal{D}$ is available. This makes our approach well suited to real-time applications. Note that L^2 is a space of equivalence classes of functions rather than a space of functions. As

a consequence, $(\forall f \in L^2)$ $f(x)$ is not well-defined unless we regard $f \in \mathcal{G}$, i.e., as an element of the class $\mathcal{G}$ in (4.3) in which case the values $f(x)$ are defined as

$$((w_i)_{i\in\mathbb{N}} \subset \mathbb{R})\ ((x_i)_{i\in\mathbb{N}} \subset \mathbb{R})\ (\forall x \in \mathbb{R})\ f(x) := \sum_{i=1}^{N} w_i\ \kappa(x, x_i).$$

An approximation of $f \in \mathcal{G}$ therefore entails determination of the weights w_i.

3. The inner-product (4.2) has a closed-form solution, which is very convenient for set-theoretic projection-based algorithms that are well-known for their simplicity and nice convergence properties [CCC$^+$12].

In light of the above, the objective now becomes to find a $\varphi^* \in \mathcal{G}$ that is in agreement with all the available information we have about $\varphi_{\mathbf{X}}$. More precisely, suppose that the available information amounts to the fact that $\varphi_{\mathbf{X}}$ is a member of Q closed-convex sets, i.e.,

$$(\forall q \in \overline{1, Q})\ \varphi_{\mathbf{X}} \in C_q \subset \mathcal{G}.$$

Then an approximation of $\varphi_{\mathbf{X}}$ is a solution to the classical problem [see, e.g., [CCC$^+$12]]:

Problem 4.1 (Set Feasibility Problem). *Find $\varphi^* \in \mathcal{G}$ such that $\varphi^* \in \bigcap_{q=1}^{Q} C_q$.*

Obviously, the "quality" of the solution to Problem 4.1 depends on the accuracy of the information on which the sets C_q are based on, while the complexity depends on the adopted algorithmic approach. Set feasibility problems such as Problem 4.1 can be solved by a plethora of projection-based algorithms and their complexity depends on the complexity of calculating the projections P_{C_q} [Com93]. In the following, we present construction of the sets C_q and the details of the adopted projection algorithm.

4.2.1 Construction of Closed Convex Sets

Consider the event $\{a_q \leq \mathbf{X} \leq b_q\}$ [$\mathbf{X}$ is the random variable defined above] where the probability of this event

$$\mathbb{P}[\ a_q \leq \mathbf{X} \leq b_q\ ] = \overline{p}_q$$

is unknown. Given a sample set $\mathcal{D}_{\mathbf{X}} := \{x_1, x_2, \ldots, x_N\}$, we can divide the range of values in $\mathcal{D}_{\mathbf{X}}$ in intervals $(\forall q \in \overline{1, Q})$ $[a_q, b_q]$, where Q is a design parameter. Because the intervals $[a_q, b_q]$ are calculated from $\mathcal{D}_{\mathbf{X}}$, which is random, $\overline{p}_q$ is a random variable. We adopt the approach in [SYY98, Ch. 6.5] to calculate the 95% confidence intervals $\mathcal{P}_q := [P_q^{\mathrm{L}}, P_q^{\mathrm{H}}]$ for each $\overline{p}_q$ such that

$$(\forall q \in \overline{1, Q})\ \mathbb{P}[\ P_q^{\mathrm{L}} \leq \overline{p}_q \leq P_q^{\mathrm{H}}\ ] \approx 0.95.$$

These calculations are computationally inexpensive and their details are provided in Section 4.5.1.

Since $\varphi_{\mathbf{X}}$ is the pdf of $\mathbf{X}$, it must be a member of each set C_q given as

$$C_q := \left\{ \varphi \in \mathcal{G} : \mathbb{P}[\ a_q \leq \mathbf{X} \leq b_q\ ] = \int_{a_q}^{b_q} \varphi(x)dx \in \mathcal{P}_q \right\},$$

which also implies that $\varphi_{\mathbf{X}} \in \bigcap_{q \in \overline{1,Q}} C_q \subset \mathcal{G}$. Furthermore, $\varphi_{\mathbf{X}}$ must also satisfy the "necessary" conditions for pdfs:

1. **Normalization**: $(\forall x \in \mathbb{S}) \int_{\mathbb{S}} \varphi_{\mathbf{X}}(x)dx = 1$; $\mathbb{S} \subset \mathbb{R}$ is the support of $\varphi_{\mathbf{X}}$.

2. **Non-negativity**: $(\forall x \in \mathbb{S})\ \varphi_{\mathbf{X}}(x) \geq 0$.

We assume that $\mathbb{S}$ is a bounded interval in $\mathbb{R}$, i.e., $\mathbb{S}$ is finite. We denote by C_{Q+1} and C_{Q+2}, the sets of functions that satisfy the two necessary conditions, respectively, and note that we now must have that

$$\varphi_{\mathbf{X}} \in \bigcap_{q \in \overline{1,Q+2}} C_q \subset \mathcal{G}.$$

In the next section we present an iterative algorithm to obtain an approximation $\varphi_{\mathbf{X}} \in \bigcap_{q \in \overline{1,Q+2}} C_q \subset \mathcal{G}$.

4.2.2 The Parallel Projection Algorithm

It can be verified that the sets C_{Q+1} and C_{Q+2} are non-empty closed convex subsets of $\mathcal{G}$. The sets $C_1, C_2, \ldots, C_Q$ are closed-convex but they may be empty if the intervals $[a_q, b_q]$ are set too small. To ensure that these sets are nonempty and they contain a sufficient number of samples, we can make the intervals $[a_q, b_q]$ sufficiently large [an appropriate choice is to select the length equal to $\sqrt{2}\sigma$, where $\sigma > 0$ is the width of the Gaussian kernel]. Then we can simply leave out the empty sets from the sequence $C_1, C_2, \ldots, C_Q$.

In light of the above, we may use the standard (sequential) projection onto convex sets (POCS) algorithm [SYY98, Theorem 2.5-1] to find a

$$\varphi^* \in \bigcap_{q \in \overline{1,Q+2}} C_q,$$

assuming that $\bigcap_{q \in \overline{1,Q+2}} C_q \neq \emptyset$. However, in some cases we may have that $\bigcap_{q \in \overline{1,Q+2}} C_q = \emptyset$, in which case an approximation φ yielded by sequential POCS may not be sufficiently close to, in particular, the "necessary sets" C_{Q+1} and C_{Q+2}. To deal with this problem, we use the parallel projection algorithm [see Definition 4.1] which guarantees convergence to a point that minimizes the weighted sum of minimum distances defined in (4.4) from each C_q. Note that if $\bigcap_{q\in\overline{1,Q+2}} C_q \neq \emptyset$, both the sequential and parallel methods converge to a point $\varphi^* \in \bigcap_{q\in\overline{1,Q+2}} C_q$.

Definition 4.1 (Parallel Projection Algorithm)**.** [SYY98, Corollary 2.10-1]. Consider the distance

$$\phi(\varphi) := \sum_{q=1}^{Q+2} \beta_q \left\|\varphi - \mathbf{P}_{C_q}(\varphi)\right\|_{\mathcal{G}}^2 . \tag{4.4}$$

For every choice of $\varphi_{(0)} \in \mathcal{G}$ and every choice of $(\beta_q)_{q\in\overline{1,Q+2}} \subset \mathbb{R}_{>0}$ such that $\sum_{q=1}^{Q+2} \beta_q = 1$, the sequence $\varphi_{(n)}$ generated by

$$\varphi_{(n+1)} = \sum_{q=1}^{Q+2} \beta_q \mathbf{P}_{C_q}(\varphi_{(n)})$$

converges to a $\varphi^* \in \arg\min \phi(\varphi) \in \mathcal{G} \subset L^2$.

The design parameters β_q assign priorities to sets C_q. Therefore, it is intuitive to set $(\forall q \in \overline{1,Q})$ $\beta_{Q+2} = \beta_{Q+1} > \beta_q$. In the following section we show how to calculate each $\mathbf{P}_{C_q}(\varphi_{(n)})$ above.

4.2.3 Details of the Projections

As mentioned previously, all the required inner-products in the following have well-known closed-form solutions. Before we proceed further, we provide some basic results pertaining to the subspace $\mathcal{G}$ in (4.3) to be utilized in this section. Denote by $\mathbf{G} \in \mathbb{R}_{>0}^{N\times N}$ the Gram matrix with entries given by

$$(\forall i, j \in \overline{1,N})\ [\mathbf{G}]_{i,j} := \langle \kappa(\cdot, x_i), \kappa(\cdot, x_j)\rangle_{\mathcal{G}},$$

which is symmetric and positive-definite.

- **Projection onto $\mathcal{G}$**: For a function $h \in L^2$, define a vector-valued mapping

  $$\boldsymbol{\xi} : h \mapsto [\langle h, \kappa(\cdot, x_1)\rangle_{\mathcal{G}}, \ldots, \langle h, \kappa(\cdot, x_N)\rangle_{\mathcal{G}}]^\intercal \in \mathbb{R}^N.$$

  The projection of h onto the subspace $\mathcal{G} \subset L^2$ in (4.3) denoted by $\mathbf{P}_{\mathcal{G}}(h)$ (also referred to as the closest-point to h in $\mathcal{G}$) is given as

  $$\mathbf{P}_{\mathcal{G}}(h) = \sum_{i=1}^{N} \zeta_i(h)\kappa(\cdot, x_i),$$

where $\zeta_i(h) \in \mathbb{R}$ is the ith component of the vector $\boldsymbol{\zeta}(h)$ that is determined by $\mathbf{G}\boldsymbol{\zeta}(h) = \boldsymbol{\xi}(h)$ [Lue97, Ch. 6.9 , Ch. 3.6]. Note that $\mathbf{G}$ is always invertible since it is positive-definite but it still may be ill-conditioned. Therefore, one should use the pseudo-inverse of $\mathbf{G}$ instead of the inverse to determine $\boldsymbol{\zeta}(h)$.

- **Inner-products in $\mathcal{G}$**: For two functions $h = \sum_{i=1}^{N} v_i \kappa(\cdot, x_i)$ and $f = \sum_{i=1}^{N} w_i \kappa(\cdot, x_i)$, their inner-product is given by

$$\langle h, f \rangle_{\mathcal{G}} = \mathbf{v}^{\mathsf{T}} \mathbf{G} \mathbf{w},$$

with $\mathbf{v} := [v_1, \cdots, v_N] \in \mathbb{R}^N$ and $\mathbf{w} := [w_1, \cdots, w_N] \in \mathbb{R}^N$.

Let $\varphi_{(0)} = \sum_{i=1}^{N} w_{(i,(0))} \kappa(\cdot, x_i)$, with $(\forall i \in \overline{1, N})$ $w_{(i,(0))} \in \mathbb{R}$ arbitrary, in Definition 4.1. We now show how to calculate each $\mathbf{P}_{C_q}(\varphi_{(n)})$ in the parallel projection algorithm in Definition 4.1.

1. **Sample Sets**: As discussed above, we must have that

$$(\forall q \in \overline{1, Q}) \int_{a_q}^{b_q} \varphi(x) dx = \int_{-\infty}^{\infty} \mathbf{1}^q(x) \varphi(x) dx \in \mathcal{P}_q, \mathcal{P}_q = [P_q^{\mathrm{L}}, P_q^{\mathrm{H}}],$$

where

$$\mathbf{1}^q(x) := \begin{cases} 1, & x \in [a_q, b_q], \\ 0, & \text{otherwise.} \end{cases}$$

The projection $\mathbf{P}_{C_q}(\varphi_{(n)})$ onto the closed-convex set (a hyperslab)

$$C_q := \left\{ \varphi \in \mathcal{G} : \langle \mathbf{P}_{\mathcal{G}}(\mathbf{1}^q), \varphi \rangle_{\mathcal{G}} = \int_{-\infty}^{\infty} \mathbf{1}^q(x) \varphi(x) dx \in \mathcal{P}_q \right\}$$

is given as [see Section 1.6.2]

$$\mathbf{P}_{C_q}(\varphi_{(n)}) = \begin{cases} \varphi_{(n)} - \frac{q^q - P_q^{\mathrm{H}}}{\|\mathbf{P}_{\mathcal{G}}(\mathbf{1}^q)\|_{\mathcal{G}}^2} \mathbf{P}_{\mathcal{G}}(\mathbf{1}^q), & \text{if } q^q - P_q^{\mathrm{H}} > 0, \\ \varphi_{(n)} - \frac{q^q - P_q^{\mathrm{L}}}{\|\mathbf{P}_{\mathcal{G}}(\mathbf{1}^q)\|_{\mathcal{G}}^2} \mathbf{P}_{\mathcal{G}}(\mathbf{1}^q), & \text{if } q^q - P_q^{\mathrm{L}} < 0, \\ \varphi_{(n)}, & \text{otherwise,} \end{cases}$$

where $q^q := \langle \mathbf{P}_{\mathcal{G}}(\mathbf{1}^q), \varphi_{(n)} \rangle_{\mathcal{G}}$ and its computation is shown in Section 4.5.2.

2. **Normalization**: As discussed above, we must have that

$$\int_{-\infty}^{\infty} 1^{\mathbb{S}}(x)\varphi(x)dx = 1,$$

where

$$\mathbf{1}^{\mathbb{S}}(x) := \begin{cases} +1, & x \in \mathbb{S}, \\ 0, & \text{otherwise.} \end{cases}$$

The projection $\mathbf{P}_{C_{Q+1}}(\varphi_{(n)})$ onto the closed-convex set [a hyperplane]

$$C_{Q+1} := \left\{ \varphi \in \mathcal{G} : \langle \mathbf{P}_{\mathcal{G}}(1^{\mathbb{S}}), \varphi \rangle_{\mathcal{G}} = \int_{-\infty}^{\infty} 1^{\mathbb{S}} \varphi(x)dx = 1 \right\},$$

is given by [see Section 1.6.1 and Section 4.5.1]

$$\mathbf{P}_{C_{Q+1}}(\varphi_{(n)}) = \left\{ \quad \varphi_{(n)} - \frac{\langle \mathbf{P}_{\mathcal{G}}(1^{\mathbb{S}}), \varphi_{(n)} \rangle_{\mathcal{G}} - 1}{\|\mathbf{P}_{\mathcal{G}}(1^{\mathbb{S}})\|_{\mathcal{G}}^2} \mathbf{P}_{\mathcal{G}}(1^{\mathbb{S}}). \right.$$

3. **Non-Negativity**: Recall that $\varphi_{(n)}$ has the general form $((v_i)_{i \in \overline{1,N}} \subset \mathbb{R})$ $\varphi_{(n)} = \sum_{i=1}^{N} v_i \kappa(\cdot, x_i)$. A sufficient condition for nonnegativity of φ is that $(\forall i \in \overline{1,N})$ $v_i \geq 0$. Ensuring this condition entails projection onto the set

$$C_{Q+2} = \left\{ \varphi \in \mathcal{G} : \varphi = \sum_{i=1}^{N} w_i \kappa(\cdot, x_i), (\forall i \in \overline{1,N}) w_i \geq 0 \right\},$$

which is a closed convex cone [BBW18].

Now we discuss two methods of calculating the projection $\mathbf{P}_{C_{Q+2}}(\varphi_{(n)})$ which does note have a closed-form solution because $\mathbf{P}_{C_{Q+2}}(\varphi_{(n)})$ is the solution of the following quadratic program (QP) as shown in Proposition 4.1. The proof is shown in Section 4.5.3.

Proposition 4.1. *Let* $\mathbf{v} = [v_1, v_2, \cdots, v_N]^\intercal \in \mathbb{R}^N$ *and* $\varphi_{(n)} = \sum_{i=1}^{N} v_i \kappa(\cdot, x_i)$. *The projection* $\mathbf{P}_{C_{Q+2}}(\varphi_{(n)})$ *is given as* $\mathbf{P}_{C_{Q+2}}(\varphi_{(n)}) = \sum_{i=1}^{N} w_i \kappa(\cdot, x_i)$, *where* $(i \in \overline{1,N})$ $w_i \geq 0$ *is the* i*th component of*

$$\mathbf{w}^* \in \arg\min_{\mathbf{w} \geq 0} \frac{1}{2} \mathbf{w}^\intercal \mathbf{G} \mathbf{w} - \mathbf{w}^\intercal \mathbf{G} \mathbf{v}.$$

Note that the above QP is always feasible since C_{Q+2} is nonempty and the projection onto $\mathbf{P}_{C_{Q+2}}(\varphi_{(n)})$ always exists. QPs can be solved using standard convex solvers and many efficient open-source solvers exist. The above QP is also equivalent to the classical nonnegative least squares problem so a large body of research work has been carried out to solve it efficiently and fast. These include interior point methods, active set methods, and various accelerated versions of the gradient descent method. The complexity of the above QP is not an issue because N is assumed to be small.

Nevertheless, in Proposition 4.2 we provide another method to compute $\mathbf{P}_{C_{Q+2}}(\varphi_{(n)})$ which consists of a simple iterative fixed point algorithm.

Proposition 4.2. *[BFN15] Let* $\mathbf{v} = [v_1, v_2, \cdots, v_N]^\intercal \in \mathbb{R}^N$ *and* $\varphi_{(n)} = \sum_{i=1}^{N} v_i \kappa(\cdot, x_i)$. *Let* $\mathbf{E}_+ := \mathbf{G} + \mathbf{I}$, $\mathbf{E}_- := \mathbf{G} - \mathbf{I}$, *with* $\mathbf{I}$ *the identity matrix and* $\mathbf{b} = 2\mathbf{G}\mathbf{v}$. *The projection* $\mathbf{P}_{C_{Q+2}}(\varphi_{(n)})$ *is given as* $\mathbf{P}_{C_{Q+2}}(\varphi_{(n)}) = \sum_{i=1}^{N} w_i \kappa(\cdot, x_i)$, *where* $(\forall i \in \overline{1, N})$ $w_i \geq 0$ *is the ith component of the limit of the sequence generated by the iteration*

$$\mathbf{E}_+ \mathbf{w}^{k+1} := -\mathbf{E}_- \left| \mathbf{w}^k \right| - \mathbf{b},$$

where $(\forall k \in \mathbb{N})$ $\left|\mathbf{w}^k\right| := [|w_1^k|, |w_2^k|, \ldots, |w_N^k|]^\intercal \in \mathbb{R}^N$ *and* $\mathbf{w}^1 \in \mathbb{R}^N$ *is arbitrary. The following error bound holds*

$$(\forall k \in \mathbb{N}) \quad \left\| \mathbf{w} - \mathbf{w}^k \right\| \leq \frac{\|\mathbf{E}\|}{1 - \|\mathbf{E}\|} \left\| \mathbf{w}^{k+1} - \mathbf{w}^k \right\|. \tag{4.5}$$

where $\|\mathbf{E}\| < 1$ *is the maximum (in moduli) singular value of* $\mathbf{E}$.

Proof. The proof in [BFN15] (that deals with Euclidean spaces) can be easily adjusted to fit our needs in $\mathcal{G}$ by noting the fact that the set $\{\kappa(\cdot, x_i), \ldots, \kappa(\cdot, x_N)\}$ plays the role of basis in $\mathcal{G}$ with the Gram matrix $\mathbf{G}$. □

The bound in (4.5) means that we can use $\left\|\mathbf{w}^{k+1} - \mathbf{w}^k\right\| < \epsilon$ as a stopping criteria, where ϵ is a given accuracy of the solution.

4.2.4 Adding Boundary Correction

Consider, without the loss of any generality, the uniform density with support $\mathbb{S} := [-2, 2]$. Suppose we have $N = 50$ samples from this uniform density. Figure **??** illustrates the boundary problem with kernel density estimators. In this example we concentrate on the Gaussian kernel and show a general kernel density estimator [for this we used Matlab's *ksdenisty* estimator with default settings] and the kernel density estimator of [BGK10]. We assume that we do not have the explicit knowledge of $\mathbb{S}$. We see that the Gaussian kernel density estimate decays too fast near the boundaries of $\mathbb{S}$ because of the lack of the data around the boundaries.

Before we present our boundary correction technique that is based on the classical boundary reflection method [Jon93], we discuss how this simple method works as proposed in [Jon93]. A simple way to deal with the lack of the data around the boundaries is to reflect the data on either side of $\mathbb{S}$ and add this artificial data to the sample set. Then a new kernel density estimate is obtained for this new data set. In more detail, denote by $\mathcal{D}_{\mathbf{X}} := \{x_1, x_2, \ldots, x_{N-1}, x_N\}$ the original sample set. The new sample set of size $3N$ is given as $\mathcal{D} := \{x_1^-, x_2^-, \ldots, x_{N-1}^-, x_N^-, x_1, x_2, \ldots, x_{N-1}, x_N, x_1^+, x_2^+, \ldots, x_{N-1}^+, x_N^+\}$, where $(\forall i \in \overline{1,N})$ $x_i^- = 2(\min \mathcal{D}_{\mathbf{X}}) - x_i$ and $x_i^+ = 2(\max \mathcal{D}_{\mathbf{X}}) - x_i$. The kernel density estimation based on the new sample set is shown in Figure 4.1. We see that though the estimate now is “flatter” near the boundary, there is loss of mass on $\mathbb{S}$ because the total mass is now spread over a larger region. Note that even the original estimate shown in Figure **??** does not integrate to unity on $\mathbb{S}$ so in this sense it is not a proper pdf. Nevertheless, we can see that most of the mass lies within $\mathbb{S}$ which shows that Gaussian kernel density estimators can naturally “sense” the support of the underlying density.

A particular inelegant solution to compensate for the loss of mass on $\mathbb{S}$ in Figure 4.1 is to multiply the estimate by some number, e.g., 3. Doing this increases the mass somewhat on $\mathbb{S}$ but it generally gives a bad approximation of the underlying density. Furthermore, increasing the size of the sample set to $3N$ by reflecting all of the data clearly increases the complexity in our framework. In our case, we have an explicit normalization constraint [see Section 4.2.1] which means that we can always normalize the density on the original support. We propose to reflect a small percentage of data (e.g., 10-20%) on both sides of the compact support and then feed the new data set to the algorithm. Since we do not know $\mathbb{S}$ we can obtain a rough estimation from $\mathcal{D}_{\mathbf{X}}$. In this sense, our method is an improved boundary reflection method that furnishes proper pdfs while not drastically increasing the complexity. In Section 4.4.1 we show the efficacy of our simple yet effective method for some typical pdfs.

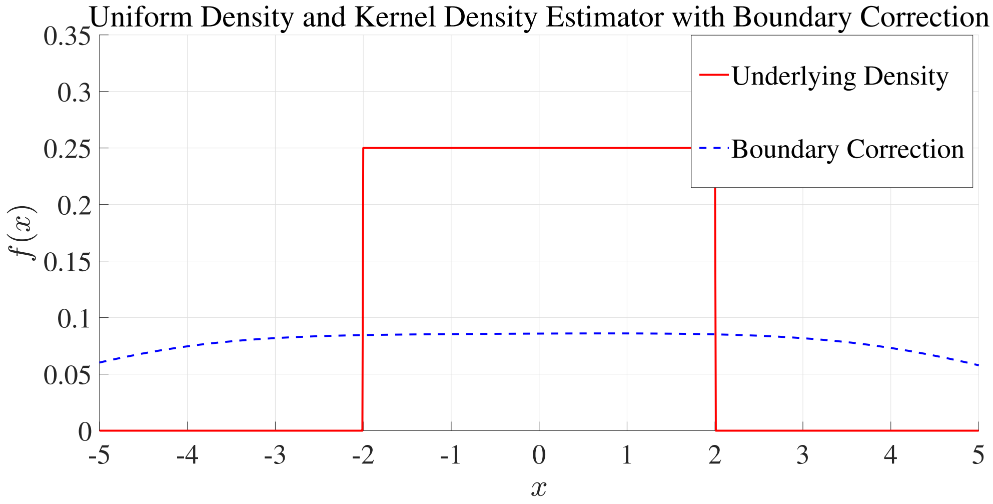


Figure 4.1

4.3 Detection in CRAN Systems with a Limited-Capacity Fronthaul

We consider a CRAN system [CCY+15, USDP17] comprising R RRHs, where each RRH has M antennas. A system with 3 RRHs is shown in Figure 4.2. An RRH is connected to a central unit by a capacity-limited and interference-free fronthaul link whose capacity is bounded above by B_p bits per packet. We consider the uplink where K single-antenna devices broadcast their data to the RRHs [HGW+17, Hua16, NAY+17].

The techniques presented below work in real Hilbert spaces but they can also be used in the complex case by using a bijection [see Section 3.4.2] between an M-dimensional complex vector and $2M$-dimensional real vectors. For clarity of presentation, we consider BPSK modulation in the following. The techniques can be extended to higher modulation schemes by following the discussion in Section 3.4.

At time $t \in \mathbb{Z}_{\geq 0}$, the received signal (sampled at a fixed symbol rate and assuming non-dispersive channels) at RRH $l \in \overline{1, R}$ is given by,

$$\mathbf{r}^l : \mathbb{Z}_{\geq 0} \to \mathbb{R}^{2M} : t \mapsto \sum_{k=1}^{K} \sqrt{p_k} b_k(t) \mathbf{h}_k^l(t) + \mathbf{n}^l(t), \tag{4.6}$$

where $b_k(t) \in \{+1, -1\}$ and $p_k \in \mathbb{R}$ are, respectively, the BPSK symbol and the (fixed) transmit power of device $k \in \overline{1, K}$. The vectors $\mathbf{h}_k^l(t) \in \mathbb{R}^{2M}$ and $\mathbf{n}^l(t) \in \mathbb{R}^{2M}$ denote the channel signature of device k and additive noise at RRH l, respectively. Note that all real vectors are obtained from the original complex vectors by using the bijection in

Section 3.4.2. Akin to the previous chapters, and because our work is a special case of bounded error estimation, we assume that the noise is bounded, i.e., $(\forall l \in \overline{1,R})$ $(\exists W^l \geq 0)$ $(\forall t \in \mathbb{Z}_{\geq 0})$ $\|\mathbf{n}^l(t)\| \leq W^l$.

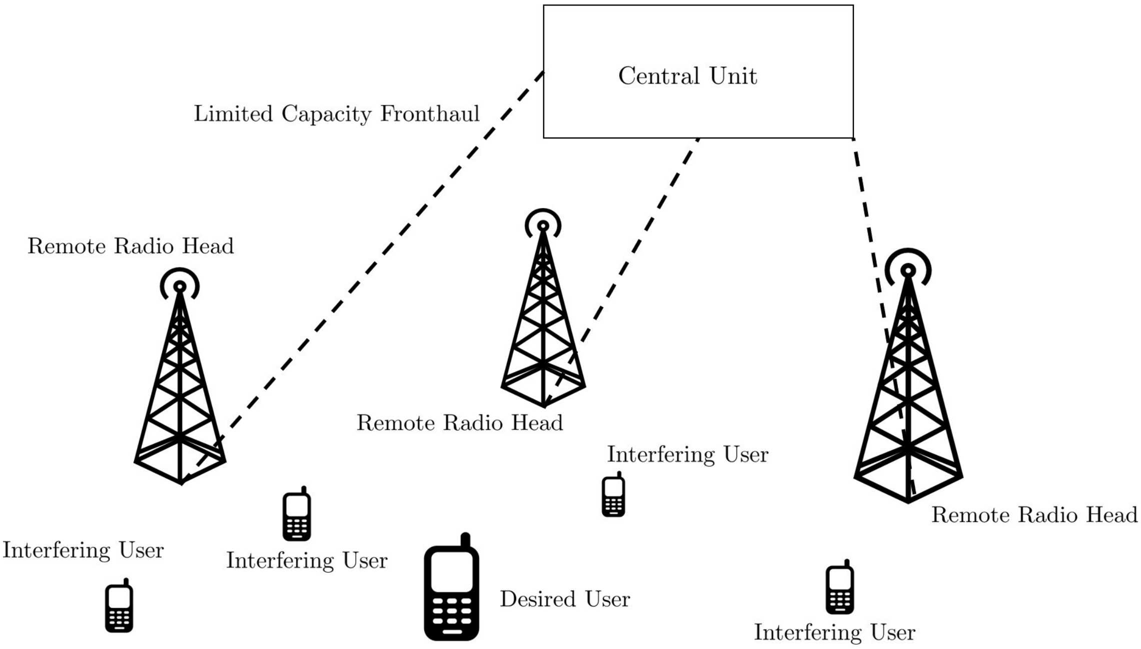


Figure 4.2: CRAN: A CRAN system with 3 RHHs having different distances and channels to the users. The fronthaul can be wired or wireless.

Learning under Small-Scale Fading

Note that the channel signature $\mathbf{h}_k^l(t)$ contains the path-loss, the receive antenna array signature, and small-scale fading. As common in the literature, in which small-scale fading is taken into account, we further assume that the channels between the users and the RRHs undergo Rayleigh block fading [MH99]. Under this assumption, channels remain constant for a block of complex channel symbols known as the coherence block. More precisely, let $(B \in \mathbb{N})$ $t_B \in \mathbb{N}$ denote the start of the coherence block B, where $|t_B - t_{B+1}| := T_{\text{block}}$ is the coherence block size. Then,

$$(\forall k \in \overline{1,K})\ (\forall t \in \overline{t_B, t_{B+1}-1})\ (\exists \mathbf{h}_k^B \in \mathbb{C}^M)\ \mathbf{h}_k(t) = \mathbf{h}_k^B;$$

i.e., $\mathbf{h}_k^B$ is the fixed channel of user k for the coherence block B, which lasts from time $t = t_B$ to time $t = t_{B+1} - 1$.

Many mobile communication systems perform channel estimation or learning of other parameters before the actual data communication [ALCYS18,DV17,WWJ+16]. Under the assumption of Rayleigh block fading, learning (through training) and data communication is performed within each coherence block which is defined as a block of channel symbols over which the channel is assumed to be constant. We assume that the first $T_{\text{train}} < T_{\text{block}}$ channel symbols are used for training. In the remaining time period of $T_{\text{block}} - T_{\text{train}}$, data communication can be performed provided that there exists a detection filter $f_l^k : \mathbb{R}^{2M} \to \mathbb{R}$ to detect the modulation symbol of device $k \in \overline{1, K}$ reliably. This means that any learning algorithm has to work with relatively short training, i.e., a small sample set, before the channel changes to a new independent value rendering the training process useless for the next coherence block.

In the following, we omit the index k since the same processing is applied to each device in parallel. Moreover, we also omit the coherence block index B because the same learning method is performed in each coherence block.

4.3.1 Learning-Based Detect-and-Forward Strategy

In the conventional CRAN, the received signal (4.6) is simply quantized and forwarded to the central unit for centralized processing. We refer to this strategy as *quantize-and-forward* (Q&F) in the remainder.

In contrast, we study a learning-based *detect-and-forward* (D&F) approach which consists of the following steps illustrated in Figure 4.3:

1. **Training**: During time period T_{train}, each RRH $l \in \overline{1, R}$ performs the training to learn a detection filter f_l such that

   $$(\forall t \in \mathbb{Z}_{\geq 0}) \ f_l(\mathbf{r}^l(t)) = b(t) + \widetilde{n}(t), \tag{4.7}$$

   where $\widetilde{n}(t)$ is the residual multiuser interference and noise. The training is performed using a training sequence $(\mathbf{r}^l(t), b(t))_{t \in \overline{1, T_{\text{train}}}}$. It is important to mention here that f_l can be any appropriate continuous detection function/method, and in the simulations later we use the method presented in Section 3.4.3 which serves as an example of a learning-based method. Note that since $\mathbf{r}^l(t)$ in (4.7) is random and bounded, $f_l(\mathbf{r}^l(t))$ as defined in (4.7) is also random and bounded.

   Additionally, each RRH learns likelihood functions [see also Section 4.1.1]

   $$\varphi^l(f_l(\mathbf{r}^l(t))| + 1) = p(+1|f_l(\mathbf{r}^l(t))) \text{ and } \varphi^l(f_l(\mathbf{r}^l(t))| - 1) = p(-1|f_l(\mathbf{r}^l(t))),$$

where $p(+1|\cdot) : \mathbb{R} \to \mathbb{R}$ and $p(-1|\cdot) : \mathbb{R} \to \mathbb{R}$ are the posterior distributions for $b(t)$.[3] The approximation of these likelihood functions (which can be seen as pdfs because their values are equal to the posterior pdfs) can be performed by using the set-theoretic approximation developed in Section 4.2).

The sample set for likelihood function approximation [see Section 4.2] can be generated by observing the response of the (trained) filter f_l to the training sample set, after the training has been completed. For example, let $\varphi_{\mathbf{X}} := \varphi^l(\cdot|+1)$ denote the likelihood function of the filter response given $b(t) = +1$, and recall that a training sequence $(\mathbf{r}(t), b(t))_{t\in\overline{1,T_{\text{train}}}}$ is known at the RRH at $t = T_{\text{train}}$. Then, we can extract a sample set

$$\mathcal{D}_{\mathbf{X}} := \{f_l(\mathbf{r}(t)) : b(t) = +1, t = \overline{1, T_{\text{train}}}\} \subset \mathbb{R}$$

for $\varphi_{\mathbf{X}}$. The same applies to the case when $\varphi_{\mathbf{X}} := \varphi^l(\cdot|-1)$.

2. **Data Communication**: During data communication, the RRH calculates two likelihood values

$$\mathcal{L}_l(+1; \mathbf{r}^l(t)) := \varphi^l(f_l(\mathbf{r}^l(t))|+1) \text{ and } \mathcal{L}_l(-1; \mathbf{r}^l(t)) := \varphi^l(f_l(\mathbf{r}^l(t))|-1).$$

3. **ML Estimation at the CU**: The central unit performs a maximum likelihood estimation of $b(t)$ given by[4]

$$\hat{b}(t) = \operatorname{sgn}\left(\sum_{l=1}^{R} \log \frac{\mathcal{L}_l(+1; \mathbf{r}^l(t))}{\mathcal{L}_l(-1; \mathbf{r}^l(t))}\right), \tag{4.8}$$

where

$$\operatorname{sgn}(x) := \begin{cases} +1, & x \geq 0, \\ -1, & x < 0. \end{cases}$$

[3] We assume modulation symbols are equiprobable. Furthermore, the channel of each device to each RRH is assumed to be uncorrelated.

[4] The log-likelihood ratios in (4.8) can be combined at the central unit by using various methods including consensus and optimal log-likelihood quantization approaches [CSY14, Rav09].

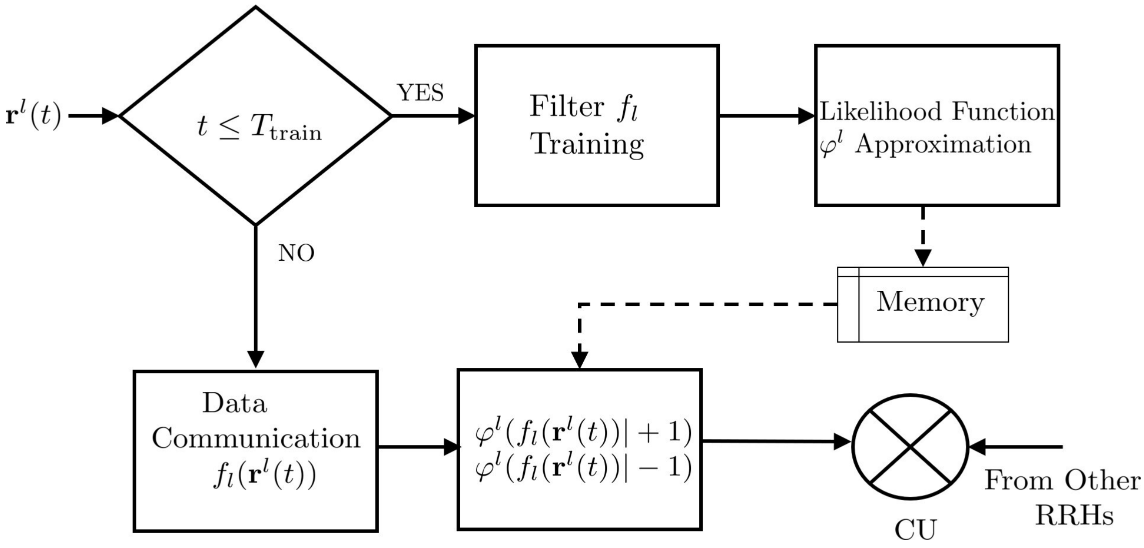

$\mathbf{r}^l(t)$
$t \leq T_{\text{train}}$
YES
NO
Filter f_l Training
Likelihood Function φ^l Approximation
Memory
Data Communication $f_l(\mathbf{r}^l(t))$
$\varphi^l(f_l(\mathbf{r}^l(t))| + 1)$
$\varphi^l(f_l(\mathbf{r}^l(t))| - 1)$
CU
From Other RRHs

Figure 4.3: Detect & Forward (D&F): The steps of the process.

4.4 Numerical Evaluation

4.4.1 Performance of the PDF Estimation Method

In this section, we illustrate the performance of the set-theoretic pdf approximation method developed in Section 4.2. We compare the performance of our method with that of the kernel density estimator studied in [BGK10] that is a state-of-the-art fast kernel density estimator, and also with that of the kernel density estimator with simple boundary correction technique. We did not consider the EM technique for Gaussian mixture because we observed a poor/unreliable performance by using a popular open-source solver [Che19]. We conjecture that this is due to insufficient number of available samples and numerical problems inherent in this method [also see Section 4.1.2]. We compare the performances of the techniques on 3 types of pdfs that are typically used for visual performance evaluation:

1. Uniform distribution defined on $[-2, 2]$.
2. Beta distribution $f(x) = x^{\alpha-1}(1-x)^{\beta-1}$, with $\alpha = 1$ and $\beta = 5$ which is only defined on the interval $[0, 1]$.
3. Normal multimodal distribution with means $\mu_1 = 0$, $\mu_2 = 35$, and $\mu_3 = 55$, and standard deviations $\sigma_1 = 1$, $\sigma_1 = 1$, and $\sigma_1 = 2$.

Note that these 3 pdfs are chosen based on the observation that these generally require a large number of samples to obtain a good approximation, especially on the boundaries.

The simulation parameters are shown in Table 4.2. The bandwidth σ of the Gaussian kernel is calculated as in [BGK10] because this purely data-driven method does not make

Table 4.2: Simulation Parameters for Pdf Approximation

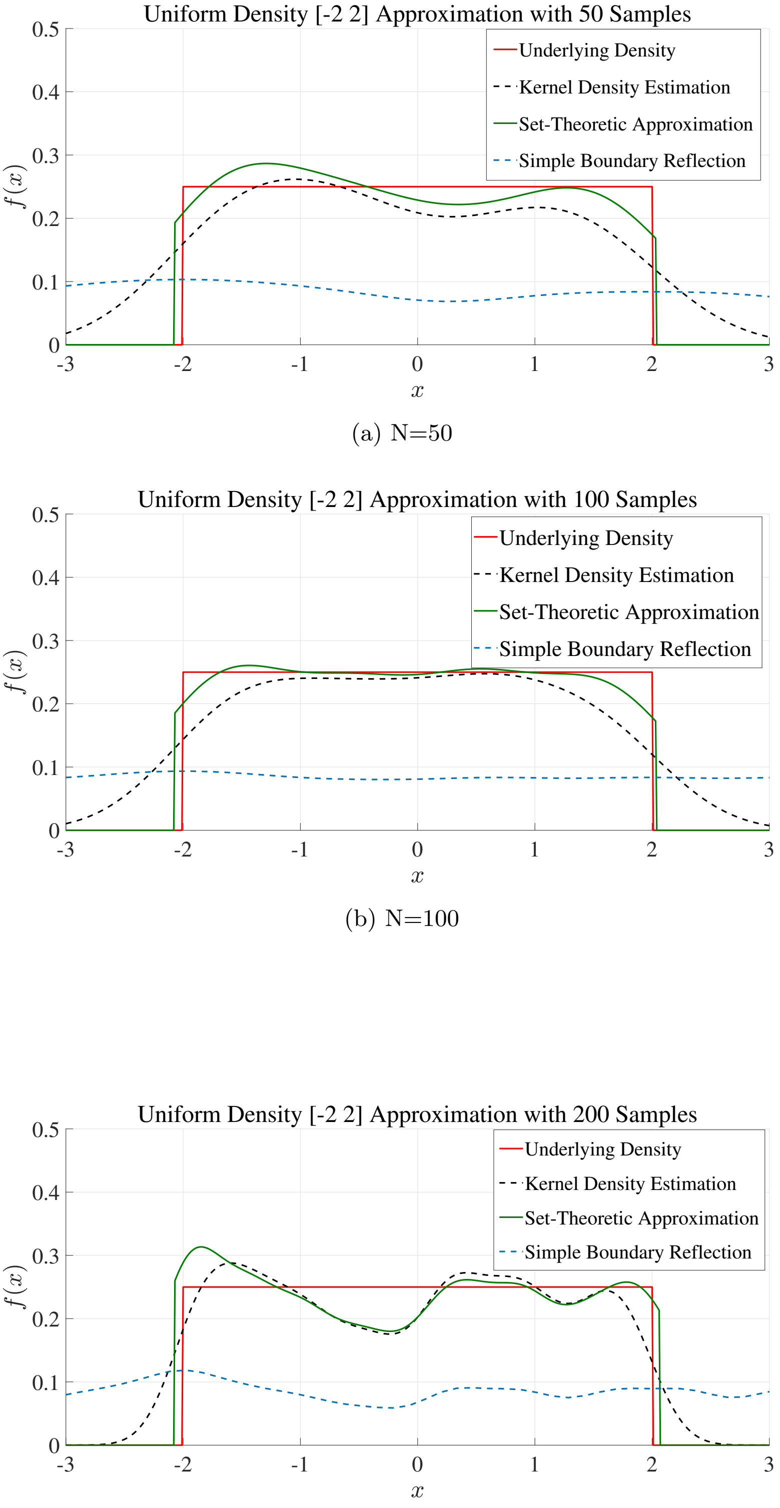
Parameter	**Calculation/Value**
Number of Samples	$N \in \{50, 100, 150, 200\}$
Bandwidth	σ_{opt} [See [BGK10]]
Length of Sample Sets	$5\sigma_{\text{opt}}$
Sample Range	$\text{Range} = \max(\mathcal{D}_{\mathbf{X}}) - \min(\mathcal{D}_{\mathbf{X}})$
Support Estimate	$\mathbb{S} = \left[\min(\mathcal{D}_{\mathbf{X}}) - \frac{\text{Range}}{20}, \max(\mathcal{D}_{\mathbf{X}}) + \frac{\text{Range}}{20}\right]$
Number of Convex Sets	$Q = \frac{\max(\mathcal{D})-\min(\mathcal{D})}{5\sigma_{\text{opt}}} + 2$
Boundary Reflection	10%
Number of POCS Iterations	20
QP Solver	Matlab's Interior Point, Tolerance 10^{-12}

any assumption on the underlying pdf. We assume that the support $\mathbb{S}$ of the underlying pdf is unknown. We obtain a simple estimate of $\mathbb{S}$ as

$$\mathbb{S} = \left[\min(\mathcal{D}_{\mathbf{X}}) - \frac{\text{Range}}{20}, \max(\mathcal{D}_{\mathbf{X}}) + \frac{\text{Range}}{20}\right],$$

where $\text{Range} = \max(\mathcal{D}_{\mathbf{X}}) - \min(\mathcal{D}_{\mathbf{X}})$ is the range of the samples in the sample set $\mathcal{D}_{\mathbf{X}}$. We apply the boundary correction as discussed in Section 4.2.4 where the reflection percentage is fixed at 10%. For our method we set $(\forall x \notin \mathbb{S})\ f(x) = 10^{-6}$ [values on points outside the support] and note that this has no effect on the approximation on $\mathbb{S}$.

Note that the purpose of this evaluation is to show that our method addresses the problems faced by present kernel density estimators using Gaussian kernels. Note that the state-of-the-art method of [BGK10] also applies boundary correction in the case when $\mathbb{S}$ is explicitly known. However, we assume that $\mathbb{S}$ is unknown because we do not assume knowledge of the underlying pdf including its support. Nevertheless, we noticed that the method in [BGK10] does not produce satisfactory performance for small sample sizes, e.g., $N = 50$ even if $\mathbb{S}$ is known explicitly. The results our shown in Figure 4.4, Figure 4.5, and Figure 4.6. We observe that our method shows an overall comparable performance to the method in [BGK10] but the performance on the boundaries of the uniform and the beta distributions is significantly improved.

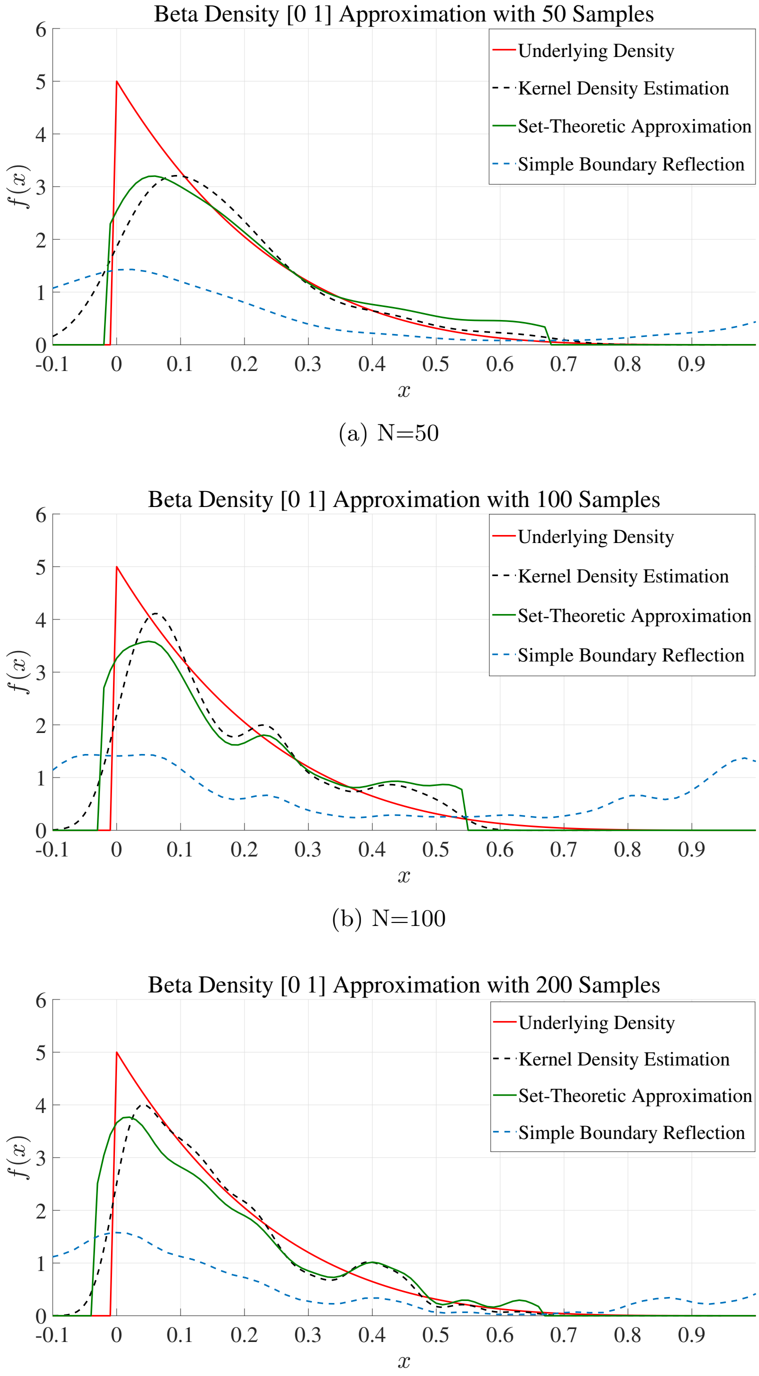


(a) N=50

(b) N=100

(c) N=200

Figure 4.4: Uniform Distribution

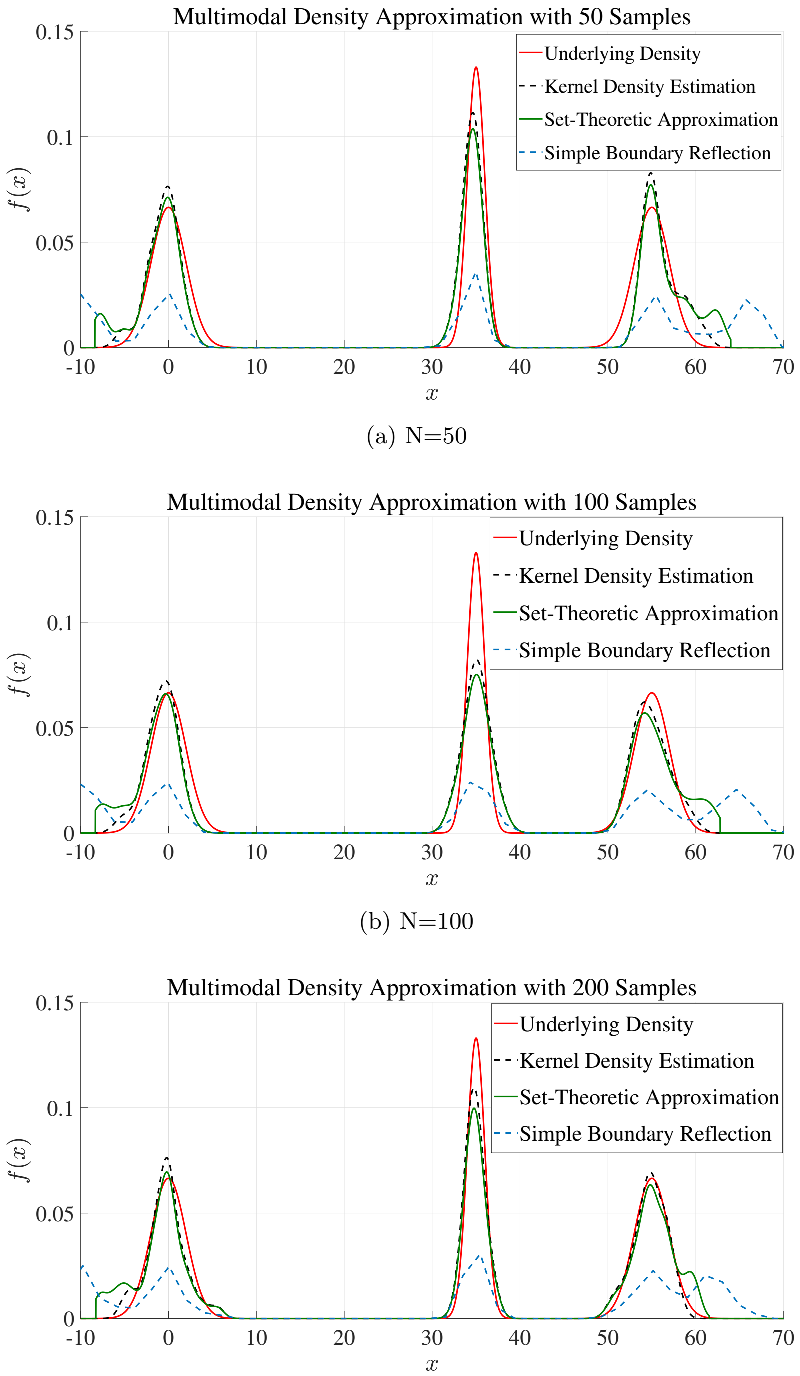


(a) N=50

(b) N=100

(c) N=200

Figure 4.5: Beta Distribution



(a) N=50

(b) N=100

(c) N=200

Figure 4.6: Multimodal Normal Distribution

Table 4.3: Simulation Parameters

Parameter	**Value**
Number of RRHs	$R \in \{2, 3, 4\}$
Number of antennas at each RRH	$M \in \{1, 2, 3, 4, 5, 6\}$
Number of users	$K \in \{4, 6, 8, 10\}$
Fronthaul quantization bits	$B_q \in \{2, 4, 6, 8, 10\}$
Quantization	Max-Llyod
User SNR	$\{-3, -2, \ldots, 9, 10\}$ dB
Channels	Rayleigh block fading
Noise power	0.1 Watts
Gaussian kernel with width for filtering σ_n	0.05
Gaussian kernel width for pdf estimation	*Silverman's* rule of thumb
Test Sample Set	$10^4 - 10^5$
Epsilon ϵ	0.95
Modulation	QPSK
Number of experiments	100
Linear kernel weight	$w_{\mathrm{L}} = 0.20$
Gaussian kernel weight	$w_{\mathrm{G}} = 0.80$

4.4.2 Performance of the CRAN System

In this section we first compare the effect of CRAN fronthaul capacity on the two CRAN strategies for QPSK modulation. Moreover, we demonstrate the diversity gains offered by cell-less CRAN framework that can acheived using our likelihood estimation method. The device SNRs ($k \in \overline{1, K}$) γ_k at each RRH are chosen independently at random from the set $\{-3\,\mathrm{dB}, -2\,\mathrm{dB}, \cdots, 9\,\mathrm{dB}, 10\,\mathrm{dB}\}$. We observed that for SNR values in this range, the device has a strong enough signal at the receiver to be detected.

Simulation of the D&F strategy

We first discuss the D&F strategy. To perform training for D&F, we use Algorithm 2 to perform multiuser demodulation [see Section 3.4.5 Chapter 3]. As mentioned before, we can use any filtering/detection algorithm, but we chose the multiuser filtering method in Chapter 3 because it is suitable for the case when the number of antennas satisfies $M < K$; this is indeed the case here. After the training phase, we use the algorithm in Definition 4.1 to approximate likelihood functions. The values in the sample set $\mathcal{D}_{\mathbf{X}} = \{x_1, x_2, \cdots, x_N\}$ [see Section 4.3.1] are used for the parameters x_i for functions $\kappa(x, x_i) := (1/\sqrt{2\pi\sigma^2}) \exp\left(\frac{-|x-x_i|^2}{2\sigma^2}\right)$. Furthermore, we used $Q = 10$ intervals in Section 4.2.1 and ran 20 iterations of the algorithm in Definition 4.1. We observed a good performance for these heuristics. The obtained functions are then quantized by using the *Max-Llyod* algorithm for quantization bits satisfying $B_{\mathrm{q}} \leq 4$. For quantization bits satisfying $B_{\mathrm{q}} > 4$, we use uniform quantization. The ML decision is performed at the CU by forwarding of likelihood ratios associated with the local detection by each RRH, as explained in Section 4.3.1.

Simulation of the Q&F strategy

For Q&F, we first collect the training data at each RRH ($l \in \overline{1, R}$) $(\mathbf{r}^l(t), b(t))_{t \in \overline{1, T_{\mathrm{train}}}}$. We then estimate and quantize the pdfs of received vectors by the same process as in the D&F case above. The quantized received vectors from all RRHs are stacked to construct (quantized) received vectors $\mathbf{r}^{\mathrm{CU}}(t) := [\mathbf{r}^1(t), \cdots, \mathbf{r}^R(t)]^\intercal \in \mathbb{R}^{2RM}$. In this way, we have a distributed multiple antennas system at the CU where the quality of the training set $(\mathbf{r}^{\mathrm{CU}}(t), b(t))_{t \in \overline{1, T_{\mathrm{train}}}}$ fed to the filtering algorithm Algorithm 2 depends on the level of quantization. The learning and detection is then performed at the CU by using Algorithm 2 to perform multiuser demodulation [see Section 3.4.5 Chapter 3]. During the detection, received vectors are quantized and forwarded to the CU by following the same procedure as during the learning.

Results

In Figure 4.7 we simulate a single cluster of size $K = 6$ with $M = 3 < K$ antennas at each RRH. Figure 4.7 shows the average (Gray-coded) BER for QPSK modulation for increasing values of fronthaul packet lengths which result in quantization bits $B_{\mathrm{q}} = B_{\mathrm{p}}/2K$ per user in the D&F case, and $B_{\mathrm{q}} = B_{\mathrm{p}}/2M$ per receive vector component in the Q&F case. We compare D&F (in solid-lines) and Q&F (in dashed-lines) forwarding strategies for different values of number of RRHs R. The D&F strategy developed in this chapter clearly outperforms the Q&F one for a low fronthaul capacity. On the other hand, Q&F is more suited to situations with a large fronthaul capacity.

In Figure 4.8 we compare the performance of a single centralized BS/RRH as in (CS) (dashed-lines) with a D&F CRAN system (solid-lines) with $R = 3$ and $B_{\mathrm{q}} = 4$ for increasing values of training time T_{t} and cluster size K. The results show that, in the case of a single RRH, the method in Section 3.4.5 Chapter 3 performs poorly even for large values of T_{train}. The spatial diversity inherent in the cell-less CRAN framework results in a much better performance over the range of T_{train}. This shows that the training time in modern communication systems can be significantly reduced by adopting the cell-less CRAN architecture and using a robust and reliable likelihood estimation method.

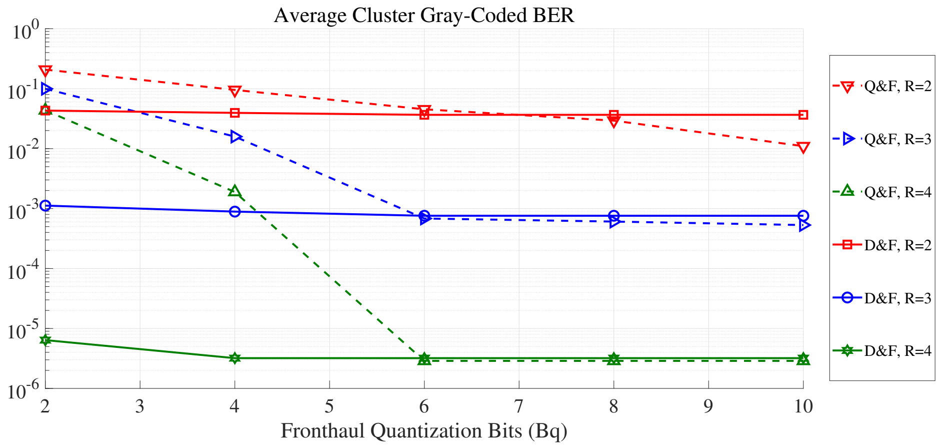


Figure 4.7: D&F vs. Q&F:

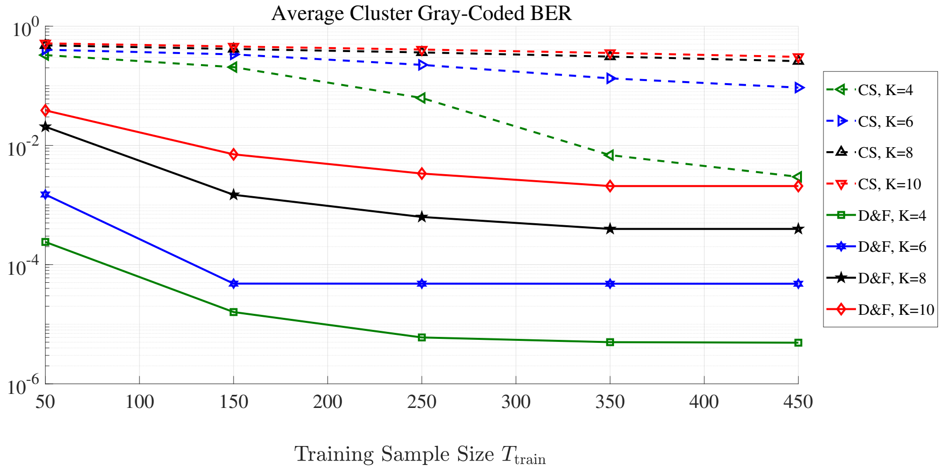


Figure 4.8: Comparison between centralized detection and D&F

4.5 Supplementary Material and Proofs

4.5.1 Calculation of Confidence Intervals

See [SYY98, Ch. 6.5] for more details of the following. Given the i.i.d sample set $\mathcal{D}_{\mathbf{X}} = \{x_1, \ldots, x_N\}$ divide this set in Q intervals $[a_q, b_q]$. Fix $q \in \overline{1, Q}$ and define Bernoulli random variables $\mathbf{Y}_i, \ldots, \mathbf{Y}_N$

$$(\forall i \in \overline{1,N})\ \mathbf{Y}_i := \begin{cases} 1, & x_i \in [a_q, b_q] \\ 0, & \text{otherwise;} \end{cases}$$

each $\mathbf{Y}_i$ has mean $\overline{p}_q$ and variance $\overline{p}_q(1-\overline{p}_q)$. Note that even though $\mathbf{Y}_i$ is in fact a mapping, the above “abuse of notation” is standard in engineering literature.

Now, define another random variable W as

$$W := \frac{\sqrt{N}(\tilde{\mathbf{Y}} - \overline{p}_q)}{\sqrt{\tilde{\mathbf{Y}}(1-\tilde{\mathbf{Y}})}},$$

where $\tilde{\mathbf{Y}} = 1/N \sum_i^N Y_i$. If N is sufficiently large, $W \to \mathcal{N}(0,1)$, i.e., W is normally distributed, which implies that there exist real numbers z_α and $-z_\alpha$ (dependent on $\alpha \in \mathbb{R}$) such that

$$\mathbb{P}[\ -z_\alpha \leq W \leq +z_\alpha\ ] = 1 - \alpha.$$

If we let

$$p_q^{\mathrm{L}} := \tilde{\mathbf{Y}} - \frac{z_\alpha\sqrt{\tilde{\mathbf{Y}}(1-\tilde{\mathbf{Y}})}}{\sqrt{N}}$$

$$p_q^{\mathrm{H}} := \tilde{\mathbf{Y}} + \frac{z_\alpha\sqrt{\tilde{\mathbf{Y}}(1-\tilde{\mathbf{Y}})}}{\sqrt{N}}$$

then it can be verified that

$$\mathbb{P}[\ p_q^{\mathrm{L}} \leq p_q \leq p_q^{\mathrm{H}}\ ] \approx 1 - \alpha.$$

Therefore, $\mathcal{P}_q = [p_q^{\mathrm{L}}, p_q^{\mathrm{H}}]$ is an approximate $100(1-\alpha)\%$ confidence interval for p_q.

4.5.2 Required Integrals and Inner-Products

1. Let $\varphi = \sum_{i=1}^{N} w_i \ \kappa(\cdot, x_i)$ and $[a_q, b_q] \in \mathbb{R}$. The inner-product $\langle \mathbf{P}_{\mathcal{G}}(\mathbf{1}^q), \varphi_{(n)} \rangle_{\mathcal{G}} = \int_{a_q}^{b_q} \varphi(x) dx$ is given as

$$\sum_{i=1}^{N} w_i \ \left\{ \frac{1}{2} \mathrm{erf}\left(\frac{a_q - x_i}{\sqrt{2}\sigma} \right) - \frac{1}{2} \mathrm{erf}\left(\frac{b_q - x_i}{\sqrt{2}\sigma} \right) \right\}.$$

2. By letting $[a_q, b_q] := \mathbb{S}$ above we obtain $\langle \mathbf{P}_{\mathcal{G}}(\mathbf{1}^{\mathbb{S}}), \varphi_{(n)} \rangle_{\mathcal{G}}$.

4.5.3 Proof of Proposition 4.1

Proof. Let $\mathbf{v} = [v_1, v_2, \cdots, v_N]^\intercal \in \mathbb{R}^N$ and $\varphi_{(n)} = \sum_{i=1}^{N} v_i \kappa(\cdot, x_i)$. Now let

$$\mathbf{k} = [\kappa(\cdot, x_i), \cdots, \kappa(\cdot, x_N)]^\intercal$$

such that the Gram matrix for $\mathcal{G}$ is given by $(\forall i, j \in \overline{1, N})$ $[\mathbf{G}]_{i,j} := \langle [\mathbf{k}]_i, [\mathbf{k}]_j \rangle_{\mathcal{G}}$. The projection of $\varphi_{(n)} = \mathbf{v}^\intercal \mathbf{k}$ onto the closed-convex cone $C_{q+2} \subset \mathcal{G}$ is the solution to

$$\mathbf{P}_{C_{q+2}}(\varphi_{(n)}) = \underset{\varphi \in C_{q+2}}{\arg\min} \ \frac{1}{2} \left\| \varphi_{(n)} - \varphi \right\|_{\mathcal{G}}^2 .$$

Now let $(\mathbf{w} \in \mathbb{R}^N_{\geq 0})$ $\varphi = \mathbf{w}^\intercal \mathbf{k}$ and note that

$$\begin{aligned} \left\| \varphi_{(n)} - \varphi \right\|_{\mathcal{G}}^2 &= \langle \varphi_{(n)} - \varphi, \varphi_{(n)} - \varphi \rangle_{\mathcal{G}} \\ &= \langle \mathbf{v}^\intercal \mathbf{k} - \mathbf{w}^\intercal \mathbf{k}, \mathbf{v}^\intercal \mathbf{k} - \mathbf{w}^\intercal \mathbf{k} \rangle_{\mathcal{G}} \\ &= \langle \mathbf{v}^\intercal \mathbf{k}, \mathbf{k}^\intercal \mathbf{v} \rangle_{\mathcal{G}} - \langle \mathbf{v}^\intercal \mathbf{k}, \mathbf{k}^\intercal \mathbf{w} \rangle_{\mathcal{G}} - \langle \mathbf{w}^\intercal \mathbf{k}, \mathbf{k}^\intercal \mathbf{v} \rangle_{\mathcal{G}} + \langle \mathbf{w}^\intercal \mathbf{k}, \mathbf{k}^\intercal \mathbf{w} \rangle_{\mathcal{G}} \\ &= \mathbf{v}^\intercal \mathbf{G} \mathbf{v} - 2\mathbf{v}^\intercal \mathbf{G} \mathbf{w} + \mathbf{w}^\intercal \mathbf{G} \mathbf{w}. \end{aligned} \tag{4.9}$$

Now since the factor $\mathbf{v}^\intercal \mathbf{G} \mathbf{v}$ in (4.9) is independent of $\mathbf{w}$, the claim follows. □

Publication List

[ACS21] [Journal Article - To be submitted] D. A. Awan, R. L.G. Cavalcante, M. Yukawa, and S. Stanczak. Robust Online Multiuser Detection: A Hybrid Model-Data Driven Approach. *to be submitted to: IEEE Transactions on Wireless Communications.*

[ACS19] [Journal Article] D. A. Awan, R. L.G. Cavalcante, and S. Stanczak. Robust Cell-Load Learning with a Small Sample Set. In *IEEE Transactions on Signal Processing*, 68:270–283, 2019.

[DCYS19] [Book Chapter] D. A. Awan, R. L. G. Cavalcante, M. Yukawa, and S. Stanczak. Adaptive Learning for Symbol Detection In *Machine Learning for Future Wireless Communications* John Wiley & Sons, Ltd, 2019.

[ACUS18] [Conference Paper] D. A. Awan, R. L. G. Cavalcante, Z. Utkovski, and S. Stanczak. Set-Theoretic Learning for Detection in Cell-Less C-RAN Systems. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 589–593, Nov 2018.

[ACS18] [Conference Paper] D. A. Awan, R. Cavalcante, and S. Stanczak. A Robust Machine Learning Method for Cell-Load Approximation in Wireless Networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018.

[ALCYS18] [Conference Paper] D. A. Awan, R. L.G. Cavalcante, M. Yukawa, and S. Stanczak. Detection for 5G-NOMA: An Online Adaptive Machine Learning Approach. In *IEEE International Conference on Communications (ICC), Kansas City, USA*, May 2018.

[ACS16] [Conference Paper] D. A. Awan, R. L. G. Cavalcante, and S. Stanczak. Distributed RAN and Backhaul Optimization for Energy Efficient Wireless Networks. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 575–579, Dec 2016.

[MACKS20] [Conference Paper] M. Mehlhose, D. A. Awan, R. L. G. Cavalcante, M. Kurras, and S. Stanczak. Machine Learning-Based Adaptive Receive Filtering: Proof-of-

Concept on an SDR Platform. In *IEEE International Conference on Communications (ICC), Dublin City, Ireland*, *accepted*, January 2020.

[LAS16] [Arxiv Article] Q. Liao, D. A. Awan, and S. Stanczak. Joint Optimization of Coverage, Capacity and Load Balancing in Self-Organizing Networks. *CoRR*, abs/1607.04754, 2016.

Bibliography

[APO92] B. Aazhang, B. . Paris, and G. C. Orsak. Neural networks for multiuser detection in code-division multiple-access communications. *IEEE Transactions on Communications*, 40(7):1212–1222, July 1992.

[Aro50] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[BB96] H. H. Bauschke and J. M. Borwein. On projection algorithms for solving convex feasibility problems. *SIAM Review*, 38(3):367–426, 1996.

[BBW18] H. Bauschke, M. Bui, and X. Wang. Projecting onto the intersection of a cone and a sphere. *SIAM Journal on Optimization*, 28(3):2158–2188, 2018.

[Bel72] G. G. Belford. Uniform approximation of vector-valued functions with a constraint. *Mathematics of Computation*, 26(118):487–492, 1972.

[Bel05] G. Beliakov. Monotonicity preserving approximation of multivariate scattered data. *BIT Numerical Mathematics*, 45(4):653–677, 2005.

[Bel06] G. Beliakov. Interpolation of Lipschitz functions. *Journal of Computational and Applied Mathematics*, 196(1):20 – 44, 2006.

[Bel18] M. Belkin. Approximation beats concentration? an approximation view on inference with smooth radial kernels. *CoRR*, abs/1801.03437, 2018.

[BFN15] J. Barrios, O. P. Ferreira, and S. Z. Nemeth. Projection onto simplicial cones by picard's method. *Linear Algebra and its Applications*, 480:27 – 43, 2015.

[BGK10] Z. Botev, J. Grotowski, and D. Kroese. Kernel density estimation via diffusion. *The Annals of Statistics*, 38, 11 2010.

[Bot16] Z. I. Botev. The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(1):125–148, Feb 2016.

[BP94] A. Berman and R. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Society for Industrial and Applied Mathematics, 1994.

[Bro14] R. F. Brown. *A Topological Introduction To Nonlinear Analysis*. Birkhaeuser Basel, 2014.

[BTA04] A. Berlinet and C. Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2004.

[Cal14] J.-P. Calliess. *Conservative decision-making and inference in uncertain dynamical systems*. PhD thesis, Department of Engineering Science, University of Oxford, 2014.

[Cas02] M. Casini. *Set-Membership Estimation: An Advanced Tool for System Identification*. PhD thesis, Department of Information Engineering and Mathematics, University of Siena, 2002.

[CCC$^+$12] Y. Censor, W. Chen, P. L. Combettes, R. Davidi, and G. T. Herman. On the effectiveness of projection methods for convex feasibility problems with linear inequality constraints. *Computational Optimization and Applications*, 51(3):1065–1088, Apr 2012.

[CCY$^+$15] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann. Cloud RAN for Mobile Networks - A Technology Overview. *IEEE Communications Surveys Tutorials*, 17(1):405–426, Firstquarter 2015.

[Che19] M. Chen. Em algorithm for gaussian mixture model (em gmm). Mathworks File Exchange, Oct. 2019. Retrieved: 20 Oct. 2019.

[CHW04] S. Chen, L. Hanzo, and A. Wolfgang. Nonlinear multiantenna detection methods. *EURASIP Journal on Advances in Signal Processing*, 2004(9):498791, Aug 2004.

[CK94] F. Cozman and E. Krotkov. Truncated Gaussians as tolerance sets. Technical Report CMU-RI-TR-94-35, Carnegie Mellon University, Pittsburgh, PA, September 1994.

[Com93] P. L. Combettes. The foundations of set theoretic estimation. *Proceedings of the IEEE*, 81(2):182–208, 1993.

[CPS$^+$13] R. L. G. Cavalcante, E. Pollakis, S. Stańczak, F. Penna, and J. Bühler. GreenNets deliverables. Technical report, GreenNets Project, FP7.SME.2011.1, 2013.

[CS02] F. Cucker and S. Smale. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39:1–49, 2002.

[CSS16] R. L. G. Cavalcante, Y. Shen, and S. Stanczak. Elementary properties of positive concave mappings with applications to network planning and optimization. *IEEE Transactions on Signal Processing*, 64(7):1774–1783, April 2016.

[CSTY13] S. Chouvardas, K. Slavakis, S. Theodoridis, and I. Yamada. Stochastic analysis of hyperslab-based adaptive projected subgradient method under bounded noise. *IEEE Signal Processing Letters*, 20(7):729–732, July 2013.

[CSY14] R. L. G. Cavalcante, S. Stanczak, and I. Yamada. *Cooperative Cognitive Radios with Diffusion Networks*, chapter Cognitive Radio and Sharing Unlicensed Spectrum in the book Mechanisms and Games for Dynamic Spectrum Allocation, pages 262–303. Cambridge University Press, UK, 2014.

[CYM09] R. L. G. Cavalcante, I. Yamada, and B. Mulgrew. An adaptive projected subgradient approach to learning in diffusion networks. *IEEE Transactions on Signal Processing*, 57(7):2762–2774, July 2009.

[CYS04] R. Cavalcante, I. Yamada, and K. Sakaniwa. A fast blind multiple access interference reduction in DS/CDMA systems by adaptive projected subgradient method. In *2004 12th European Signal Processing Conference*, pages 161–164, Sep. 2004.

[CYY05] R. L. G. Cavalcante, M. Yukawa, and I. Yamada. Set-theoretic ds/cdma receivers for fading channels by adaptive projected subgradient method. In *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.*, volume 4, pages 6 pp.–2275, Nov 2005.

[DLK$^+$17] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. K. Bhargava. A survey on non-orthogonal multiple access for 5G networks: Research challenges and future trends. *CoRR*, abs/1706.05347, 2017.

[DPS14] E. Dahlman, S. Parkvall, and J. Skold. *4G: LTE/LTE-Advanced for Mobile Broadband*. Academic Press, Inc., Orlando, FL, USA, 2nd edition, 2014.

[DRG17] M. Diligenti, S. Roychowdhury, and M. Gori. Integrating prior knowledge into deep learning. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 920–923, Dec 2017.

[DV17] J. Du and R. A. Valenzuela. How much spectrum is too much in millimeter wave wireless access. *IEEE Journal on Selected Areas in Communications*, 35(7):1444–1458, July 2017.

[EG16] R. Evans and J. Gao. Deepmind AI reduces Google data centre cooling bill by 40 percent. Deep Mind Website, July. 2016. Retrieved: 20 Oct. 2018.

[EMM04] Y. Engel, S. Mannor, and R. Meir. The kernel recursive least-squares algorithm. *IEEE Transactions on Signal Processing*, 52(8):2275–2285, Aug 2004.

[FF12] A. J. Fehske and G. P. Fettweis. Aggregation of variables in load models for interference-coupled cellular data networks. In *2012 IEEE International Conference on Communications (ICC)*, pages 5102–5107, June 2012.

[Fro13] F. Froehlich. *Approximation and Analysis of Probability Densities using Radial Basis Functions.* PhD thesis, Technische Universitaet Muenchen Department of Mathematics, 2013.

[GECS$^+$16] M. A. Gutierrez-Estevez, R. L. G. Cavalcante, S. Stanczak, J. Zhang, and H. Zhuang. A distributed solution for proportional fairness optimization in load coupled OFDMA networks. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP' 16)*, 2016.

[GW59] M. Golomb and H. Weinberger. *On Numerical Approximation.* The University of Wisconsin Press, Madison, R.E. Langer edition, 1959.

[HGW$^+$17] T. Han, X. Ge, L. Wang, K. S. Kwak, Y. Han, and X. Liu. 5G converged cell-less communications in smart cities. *IEEE Communications Magazine*, 55(3):44–50, March 2017.

[Hon08] M. L. Honig. *Overview of Multiuser Detection*, chapter 1, pages 1–45. John Wiley Sons, Ltd, 2008.

[Hua16] Huawei. Telefonica and huawei complete world's first proof-of-concept test for 5G UCNC radio access networks. http://www.huawei.com/en/news/2016/11/World-First-Proof-Test-5G-UCNC-Radio-Access-Networks, 2016.

[HYS14] C. K. Ho, D. Yuan, and S. Sun. Data offloading in load coupled networks: A utility maximization framework. *IEEE Transactions on Wireless Communications*, 13(4):1921–1931, 2014.

[IN07] Y. Isik and T. Necmi. Multiuser detection with neural network and PIC in CDMA systems for AWGN and Rayleigh fading asynchronous channels. *Wireless Personal Communications*, 43(4):1185–1194, Dec 2007.

[Jon93] C. Jones. Simple boundary correction for kernel estimation. *Statistics and Computing*, 3(3):135–146, 1993.

[JZB$^+$16] C. Jin, Y. Zhang, S. Balakrishnan, M. J. Wainwright, and M. I. Jordan. Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 4123–4131, USA, 2016. Curran Associates Inc.

[Kot16] W. Kotlowski. Online isotonic regression. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 1165–1189, 2016.

[KP03] S. G. Krantz and H. R. Parks. *The Implicit Function Theorem.* Boston(MA): Birkhaueser, 2003.

[LPV] L. Liberti, , P. Poirion, and K. Vu. Fast approximate solution of large dense linear programs. `http://www.optimization-online.org/DB_FILE/2016/11/5737.pdf`. Accessed: 2018-07-13.

[Lue97] D. G. Luenberger. *Optimization by Vector Space Methods.* John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1997.

[Mar18] G. Marcus. Deep learning: A critical appraisal. *CoRR*, abs/1801.00631, 2018.

[MH99] T. L. Marzetta and B. M. Hochwald. Capacity of a mobile multiple-antenna communication link in Rayleigh flat fading. *IEEE Transactions on Information Theory*, 45(1):139–157, Jan 1999.

[MIM$^+$18] N. Muralidhar, M. R. Islam, M. Marwah, A. Karpatne, and N. Ramakrishnan. Incorporating prior domain knowledge into deep neural networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 36–45, 2018.

[Min10] H. Q. Minh. Some properties of Gaussian reproducing kernel Hilbert spaces and their implications for function approximation and learning theory. *Constructive Approximation*, 32(2):307–338, Oct 2010.

[MK10] K. Majewski and M. Koonert. Conservative cell load approximation for radio networks with Shannon channels and its application to LTE network planning. In *2010 Sixth Advanced International Conference on Telecommunications*, pages 219–225, May 2010.

[MNK$^+$07] P. Mogensen, W. Na, I. Z. Kovács, F. Frederiksen, A. Pokhariyal, K. I. Pedersen, T. Kolding, K. Hugl, and M. Kuusela. LTE capacity compared to the Shannon bound. In *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, number 1, pages 1234–1238, 2007.

[MP94] U. Mitra and H. V. Poor. Neural network techniques for adaptive multiuser demodulation. *IEEE Journal on Selected Areas in Communications*, 12(9):1460–1470, Dec 1994.

[MT85] M. Milanese and R. Tempo. Optimal algorithms theory for robust estimation and prediction. *IEEE Transactions on Automatic Control*, 30(8):730–738, August 1985.

[Mun00] J. Munkres. *Topology (Second Edition)*. Prentice Hall, Inc., 2000.

[MV91] M. Milanese and A. Vicino. Optimal estimation theory for dynamic systems with set membership uncertainty: An overview. *Automatica*, 27(6):997 – 1009, 1991.

[MXZ06] C. A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *J. Mach. Learn. Res.*, 7:2651–2667, December 2006.

[NAY$^+$17] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta. Cell-free massive MIMO versus small cells. *IEEE Transactions on Wireless Communications*, 16(3):1834–1850, March 2017.

[NG95] B. M. Ninness and G. C. Goodwin. Rapprochement between bounded-error and stochastic estimation theory. *International Journal of Adaptive Control and Signal Processing*, 9(1):107–132, 1995.

[NG96] P. Niyogi and F. Girosi. On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. *Neural Computation*, 8(4):819–842, 1996.

[NS3] The network simulator NS3. `https://www.nsnam.org/`. Accessed: 2018-07-13.

[OY17] M. Ohnishi and M. Yukawa. Online Nonlinear Estimation via Iterative L2-Space Projections: Reproducing Kernel of Subspace. *ArXiv e-prints*, December 2017.

[PCS16] E. Pollakis, R. L. G. Cavalcante, and S. Stanczak. Traffic demand-aware topology control for enhanced energy-efficiency of cellular networks. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):1–17, 2016.

[PH00] K. Plataniotis and D. Hatzinakos. *Gaussian mixtures and their applications to signal processing.* 01 2000.

[Pon03] M. Pontil. A note on different covering numbers in learning theory. *Journal of Complexity*, 19(5):665 – 671, 2003.

[PSSS14] S. H. Park, O. Simeone, O. Sahin, and S. S. Shitz. Fronthaul compression for cloud radio access networks: Signal processing advances inspired by network information theory. *IEEE Signal Processing Magazine*, 31(6):69–79, Nov 2014.

[Rav09] W. Rave. Quantization of log-likelihood ratios to maximize mutual information. *IEEE Signal Processing Letters*, 16(4):283–286, April 2009.

[RSF14] Z. Ren, S. Stanczak, and P. Fertl. Activation of nomadic relay nodes in dynamic interference environment for energy saving. *2014 IEEE Global Communications Conference*, pages 4466–4471, 2014.

[Sch68] F. Schweppe. Recursive state estimation: Unknown but bounded errors and system inputs. *IEEE Transactions on Automatic Control*, 13(1):22–28, February 1968.

[Sch11] A. Schindler. *Bandwidth Selection in Nonparametric Kernel Estimation.* PhD thesis, Faculty of Economic Sciences of the Georg-August-Universitaet Goettingen, 2011.

[SF12] P. Skillermark and P. Frenger. Enhancing energy efficiency in LTE with antenna muting. In *2012 IEEE 75th Vehicular Technology Conference (VTC Spring)*, pages 1–5, May 2012.

[She15] Y. Shen. Load coupling model evaluation and feasibility study of power allocation in OFDMA networks, 2015.

[Sil86] B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman & Hall, London, 1986.

[Sio12] I. Siomina. Analysis of cell load coupling for LTE network planning and optimization. *IEEE Transactions on Wireless Communications*, 11(6):2287–2297, June 2012.

[SJ91] S. J. Sheather and M. C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):683–690, 1991.

[SS04] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August 2004.

[SSM98] A. Smola, B. Schölkopf, and K.-R. Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11(4):637–649, June 1998.

[ST08] K. Slavakis and S. Theodoridis. Sliding window generalized kernel affine projection algorithm using projection mappings. *EURASIP Journal on Advances in Signal Processing*, 2008(1):735351, Apr 2008.

[Str73] R. G. Strongin. On the convergence of an algorithm for finding a global extremum. *Engineering in Cybernetics,*, 1973.

[STY09] K. Slavakis, S. Theodoridis, and I. Yamada. Adaptive constrained learning in reproducing kernel Hilbert spaces: The robust beamforming case. *IEEE Transactions on Signal Processing*, 57(12):4744–4764, Dec 2009.

[Suk92] A. G. Sukharev. *Minimax Models in the Theory of Numerical Methods.* Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[SY12] I. Siomina and D. Yuan. Load balancing in heterogeneous LTE: Range optimization via cell offset and load-coupling characterization. In *2012 IEEE International Conference on Communications (ICC)*, pages 1357–1361, June 2012.

[SY15] I. Siomina and D. Yuan. Optimizing small-cell range in heterogeneous and load-coupled LTE networks. *IEEE Transactions on Vehicular Technology*, 64(5):2169–2174, May 2015.

[SYY98] H. Stark, Y. Yang, and Y. Yang. *Vector Space Projections: A Numerical Approach to Signal and Image Processing, Neural Nets, and Optics.* John Wiley & Sons, Inc., New York, NY, USA, 1998.

[The15] S. Theodoridis. *Machine Learning: A Bayesian and Optimization Perspective.* Academic Press, Inc., Orlando, FL, USA, 1st edition, 2015.

[TSY11] S. Theodoridis, K. Slavakis, and I. Yamada. Adaptive learning in a world of projections. *IEEE Signal Processing Magazine*, 28(1):97–123, 1 2011.

[TTQL17] J. Tang, W. P. Tay, T. Q. S. Quek, and B. Liang. System cost minimization in cloud RAN with limited fronthaul capacity. *IEEE Transactions on Wireless Communications*, 16(5):3371–3384, May 2017.

[TV05] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication.* Cambridge University Press, New York, NY, USA, 2005.

[TW80] J. Traub and H. Woźniakowski. *A general theory of optimal algorithms.* ACM monograph series. Academic Press, 1980.

[USDP17] Z. Utkovski, O. Simeone, T. Dimitrova, and P. Popovski. Random access in C-RAN for user activity detection with limited-capacity fronthaul. *IEEE Signal Processing Letters*, 24(1):17–21, Jan 2017.

[Vap95] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, Berlin, Heidelberg, 1995.

[Ver98] S. Verdu. *Multiuser Detection.* Cambridge University Press, New York, NY, USA, 1st edition, 1998.

[Wit68] H. Witsenhausen. Sets of possible states of linear systems given perturbed observations. *IEEE Transactions on Automatic Control*, 13(5):556–558, October 1968.

[WJ94] M. P. Wand and M. C. Jones. *Kernel Smoothing.* Number 60 in Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Chapman & Hall, Boca Raton, FL, U.S., December 1994.

[WRS$^+$16] Y. Wang, B. Ren, S. Sun, S. Kang, and X. Yue. Analysis of non-orthogonal multiple access for 5g. *China Communications*, 13(Supplement2):52–66, N 2016.

[WSEA04] I. C. Wong, Z. Shen, B. L. Evans, and J. G. Andrews. A low complexity algorithm for proportional resource allocation in OFDMA systems. In *IEEE Workshop on Signal Processing Systems*, pages 1–6, Oct 2004.

[WTT$^+$16] X. Wang, S. Thota, M. Tornatore, H. S. Chung, H. H. Lee, S. Park, and B. Mukherjee. Energy-efficient virtual base station formation in optical-access-enabled cloud-RAN. *IEEE Journal on Selected Areas in Communications*, 34(5):1130–1139, May 2016.

[WWJ$^+$16] C. K. Wen, C. J. Wang, S. Jin, K. K. Wong, and P. Ting. Bayes-optimal joint channel-and-data estimation for massive MIMO with low-precision ADCs. *IEEE Transactions on Signal Processing*, 64(10):2541–2556, May 2016.

[Yat95] R. D. Yates. A framework for uplink power control in cellular radio systems. *IEEE Journal on Selected Areas in Communications*, 13(7):1341–1347, Sep 1995.

[YO05] I. Yamada and N. Ogura. Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions. *Numerical Functional Analysis and Optimization*, 25(7-8):593–617, 2005.

[Yuk15a] M. Yukawa. Adaptive learning in cartesian product of reproducing kernel Hilbert spaces. *IEEE Transactions on Signal Processing*, 63(22):6037–6048, Nov 2015.

[Yuk15b] M. Yukawa. Online learning based on iterative projections in sum space of linear and Gaussian reproducing kernel Hilbert spaces. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3362–3366, April 2015.

[YYL$^+$18] L. You, D. Yuan, L. Lei, S. Sun, S. Chatzinotas, and B. Ottersten. Resource optimization with load coupling in multi-cell NOMA. *IEEE Transactions on Wireless Communications*, 17(7):4735–4749, July 2018.