

# Signal Quantization and Approximation Algorithms for Federated Learning

A thesis submitted in partial fulfillment of  
the requirements for the degree of

Doctor of Philosophy

by

**Vijay A**  
**(Roll No. 154078001)**

Under the guidance of:  
**Prof. Animesh Kumar**



Electrical Engineering Department  
Indian Institute of Technology Bombay  
Powai, Mumbai 400076

2022

---

മയൂരസന്ദേശ മണിപ്രവാളവും  
മടിച്ചിടാതങ്ങു രചിച്ച ഞാനുമേ;  
മഹാജനങ്ങൾക്കു രസിക്കുമെങ്കിലീ  
മമശ്രമം നിഷ്ഫലമല്ല കേവലം

–കേരളവർമ്മ വലിയ കോയിത്തമ്പുരാൻ  
1845–1914

mayūrasandēśa maṇipravāḷavum  
maṭicciṭātanīu raciccu ṅānumē;  
mahājananṅaḷkku rasikkumenkili  
mamaśramam niṣphalamalla kēvalam

–kēraḷavarm'ma valiya kēyittampurān  
1845–1914

Mayurasandesam in Manipravalam  
Thus have I penned unabashedly:  
Should it appeal to connoisseurs,  
Then my effort sha'nt go vain!

–Kerala Varma Valiya Koil Thampuram  
1845–1914

---

## Approval Sheet

This dissertation entitled "Signal Quantization and Approximation Algorithms for Federated Learning" by Mr. Vijay A (154078001) is approved for the degree of Doctor of Philosophy.

### Examiners

*Chandra R Murthy*

Chandra R. Murthy  
Professor, Dept. of ECE  
Indian Institute of Science  
Bangalore 560 012

---

Digital Signature  
Sibiraj Bhaskaran Pillai (i09051)  
24-Mar-22 01:04:13 PM

---

### Supervisor (s)

*U K Anandavardhanan*

---

---

### Chairman

---

Digital Signature  
U K Anandavardhanan (i05072)  
24-Mar-22 01:04:45 PM

---

Date : 25 March 2022

Place : IIT Bombay, Mumbai

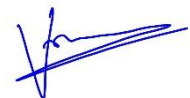
---

## Declaration

I, Vijay A, declare that this written submission represents my ideas in my own words and wherever others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/ data/ fact/ source in my submission. I understand that any violation of the above will be cause for disciplinary action by IIT Bombay and can also evoke penal action from the sources which have thus not been cited or from whom proper permission has not been taken when needed.

Date: 25 March, 2022

Place: IIT Bombay, Mumbai



Vijay A

# Abstract

Distributed signal or information processing using Internet of Things (IoT), facilitates real-time monitoring of signals, for example, environmental pollutants, health indicators, and electric energy consumption in a smart city. Despite the promising capabilities of IoTs, these distributed deployments often face the challenge of data privacy and communication rate constraints. In traditional machine learning, training data is moved to a data center, which requires massive data movement from distributed IoT devices to a third-party location, thus raising concerns over privacy and inefficient use of communication resources. Moreover, the growing network size, model size, and data volume combined lead to unusual complexity in the design of optimization algorithms beyond the compute capability of a single device. This necessitates novel system architectures to ensure stable and secure operations of such networks. Federated learning (FL) architecture, a novel distributed learning paradigm introduced by McMahan et al. [1], can be a promising solution for enabling IoT-based smart city applications by addressing these challenges. In the FL paradigm, a global server orchestrates model training, without raw data being transferred from the participating devices (or clients). Edge-deployed signal processing algorithms, such as sparse approximations and statistical learning methods will be essential for efficient management of compute and communication resources in FL.

In this thesis, we seek answers to three research questions related to distributed signal processing that arise in the context of FL. First, what are the methods to speedup scalar quantizer design in resource-constrained edge devices? Second, given certain system or application-specific constraints, what are the signal representations that lead to near-optimal performance? Third, what are the tradeoffs to be considered while performing federated aggregation of models collected from individual edge devices? These questions

are considered in the context of resource-constrained edge devices in the FL model.

Beginning with the Lloyd-Max quantizer, a well-known algorithm in traditional signal processing, we propose an approximate Lloyd-Max quantizer, which relies on a piecewise linear approximation of the signal source probability density. We show that the proposed quantizer is nearly optimal and also converge to a fixed point close to the limit of Lloyd-Max quantizer levels at an exponential rate. Further, we extend the proposed algorithm to a data-driven setting where the parameters of the piecewise linear representation are learned through batch updates. Through experiments on an Android-based edge device, we show performance improvement of the developed quantizer when compared to the well known  $k$ -means in terms of energy efficiency, runtime, and memory utilization.

Next, we consider some specific application-oriented system constraints and the signal representations useful for such cases. Of particular interest is the overprediction constraint that arise in network capacity planning problems. We develop two solutions: the first based on quantizer design and the second based on signal approximations. The overpredictive quantizer design hinges on the stochastic approximation-based updates that provide an online algorithm for finding the quantizer levels. The designed quantizer will generate a quantized signal which is always greater than or equal to the actual signal. The proposed schemes are verified and compared using an available TV whitespace dataset. The second approach to implement the system level overprediction constraint is through signal approximation, which we describe next in the context of FL.

The final part of this thesis deals with algorithms for overpredictive signal analytics in a client-server architecture motivated by FL. We propose algorithms to find signal representations that satisfy the overprediction constraint using a suitable basis representation (the Fourier basis in our application). Such overprediction constraints are typical in emerging smart grid applications where a central server monitors household electricity consumption. The signal representations computed at the edge devices (or consumer sites) aid the central server in drawing insights into signal analytics, including the time-series demand patterns and other signal statistics. We evaluate the tradeoffs between communication cost, computation cost, and the mean squared error performance through experimental studies on an off-the-shelf residential energy consumption dataset.

# Acknowledgements

During my Ph.D. journey, I have benefited from significant interactions with people both within IIT Bombay and throughout the greater academic community. I am grateful to be able to express my gratitude and acknowledge these people and their contributions.

Firstly, I have been fortunate enough to have enjoyed the support and encouragement of my thesis advisor *Prof. Animesh Kumar*. He initiated me into the world of research and guided me through the peaks and valleys of my Ph.D. life. Without a doubt, his enthusiasm, positive outlook, and unparalleled intellectual energy have always been a source of inspiration. I first met Prof. Animesh as a graduate student at IIT Bombay in his course on Statistical Signal Analysis (EE 601). This course greatly fueled my interest in problems related to statistical learning and signal processing, primarily due to the extraordinary clarity of his teaching. In the following year, I was fortunate enough to work as his Teaching Assistant (TA) for the same course. As a Ph.D. advisor, Prof. Animesh can provide guidance and direction to young graduate students while simultaneously allowing enormous freedom to explore individual interests. He also demonstrates remarkable expertise for rapidly distilling the essence of a problem and suggesting ways for tackling challenges. I also appreciate his assistance in developing my writing skills, especially in writing papers, fellowship applications, and this thesis.

I am immensely grateful to the members of my Research Progress Committee (RPC), *Prof. Vivek Borkar* and *Prof. Sibi Raj B Pillai*, for their invaluable feedback and continuous encouragement. Prof. Borkar had been instrumental in introducing me to the field of stochastic approximations, which is the workhorse of several innovations in AI and ML, and plays an important role in Chapter 3 of this thesis. From him, I have learned how to handle tough situations patiently, and I have benefited from the impressive combination

of intuition and rigor in his thinking. I wholeheartedly thank Prof. Sibi, for his sincere support and encouragement during the period of my Ph.D. As a graduate student, I had the opportunity to have him as my project mentor and I am grateful to him for his simplicity in ideas and humility. His articulate questions and counterexamples have sent me back to the drawing boards more than once.

I would like to acknowledge all the fruitful interactions with potential collaborators: *Prof. Antonio Ortega*, EE department at the University of Southern California (USC); *Prof. Ramesh Annavaajjala*, Adjunct faculty at University of Massachusetts Boston (UMB); *Dr. Hermina Petric Maretic*, currently Applied Scientist at Amazon, Switzerland; *Dr. Arun Venkitaraman*, Postdoctoral Researcher, KTH, Sweden. I am indebted to the faculty members of EE, CSE, IEOR, and Math departments, where I have attended many interesting courses. I thank the staff members of the EE department, especially Mr. Santosh S. Kharat, Ms. Tanvi D. Shelatkar, and Ms. Madhumathi G. Shetty for all the administrative help. I acknowledge with gratitude Bharti Center for Communication and the IRCC, IIT Bombay for supporting my conference travels.

I am extremely grateful for having had the opportunity to be a part of the Students' Reading Group (SRG) at EE department. The weekly student and faculty talks within SRG helped me interact with scholars from various domains, and gave me a world view to tackle problems within my own field. I would like to convey sincere gratitude to all my IIT friends. I am privileged to have shared the lab space with *Sadaf, Meghna, Santosh, Vaibhav, Shreyas, Jithin, Vishwanathan, Haseen, Jinesh, Amitalok, Vikas, Aarti, Feroz* and *Parth*. I will cherish the stimulating discussions, lunch and treats, and all the fun. I also thank *Saurabh, Kaushani, Meera, Kavitha*, and *Parth* for the invaluable brainstorming sessions. I offer gratitude to all my hostel mates, particularly, *Rakesh, Omkar, Santosh, Sandeep, Arjun, Sachin, Vaibhav, Ramachandran* and *Keshav*.

I have been blessed with a very loving and supportive family. My heart-felt gratitude goes to my parents, *Sailaja* and *Mohan*, and my brother, *Vineeth*, who have been a source of constant motivation and unconditional support. Finally, I thank the Almighty for giving me the spiritual strength to pursue the path of excellence.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Abbreviations</b>	<b>xv</b>
<b>List of Symbols</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Information processing at edge devices . . . . .	1
1.2 Distributed processing and inference in IoTs . . . . .	2
1.3 Scope and contributions of the thesis . . . . .	6
1.4 Summary of results and outline of this thesis . . . . .	10
<b>2 Speeding up of Scalar Quantizer Design for Edge Signal Processing</b>	<b>13</b>
2.1 Background . . . . .	13
2.1.1 Related literature . . . . .	16
2.2 Signal source and quantizer models . . . . .	17
2.3 Nearly optimal quantizer designs . . . . .	19
2.3.1 Approximate Lloyd-Max quantizer . . . . .	19
2.3.2 Learning Approximate Lloyd-Max quantizer . . . . .	21
2.4 Convergence of ALM algorithm . . . . .	25
2.4.1 Background concepts for showing ALM convergence . . . . .	26
2.4.2 Insights on ALM convergence using an example . . . . .	27

---

2.4.3	ALM convergence theorem . . . . .	28
2.5	Asymptotic near-LM optimality of ALM . . . . .	29
2.6	Experimental results and discussion . . . . .	32
2.6.1	Experimental results for ALM . . . . .	32
2.6.2	Experimental results for Learning ALM . . . . .	33
2.7	Summary . . . . .	39
<b>3</b>	<b>Data-Driven Approaches for Overpredictive Quantizer Design</b>	<b>41</b>
3.1	Background . . . . .	41
3.1.1	Related works . . . . .	43
3.2	System model and problem formulation . . . . .	45
3.2.1	Signal and quantization models . . . . .	45
3.2.2	Distortion measures and problem formulation . . . . .	45
3.3	Methods motivated by scalar equivalent of LVQ . . . . .	46
3.3.1	Greedy updates for envelope quantizer . . . . .	46
3.3.2	Failure of the greedy approach . . . . .	47
3.4	Stochastic approximation based design for envelope quantizers . . . . .	48
3.4.1	Minimization of mean absolute error . . . . .	48
3.4.2	Minimization of mean squared error . . . . .	50
3.5	Convergence analysis of SANE quantizer . . . . .	50
3.6	Performance evaluation of SANE . . . . .	52
3.6.1	Simulations on synthetic data . . . . .	52
3.6.2	Simulations on TV white Space dataset . . . . .	56
3.7	Approximate envelope quantizer and its variants . . . . .	56
3.7.1	Cost minimization and level updates for AEQ . . . . .	57
3.7.2	Performance of AEQ on sources with beta distribution . . . . .	58
3.7.3	Extension to Learning AEQ . . . . .	61
3.8	Summary . . . . .	62
<b>4</b>	<b>Distributed Quantizer Design and Tradeoffs in Federated Learning</b>	<b>65</b>
4.1	Background . . . . .	65
4.1.1	Related works . . . . .	66
4.2	Federated quantizer model . . . . .	67

4.2.1	Motivating examples for learning a global quantizer . . . . .	68
4.2.2	Quantizer design algorithms considered in this chapter . . . . .	69
4.2.3	Tradeoffs in federated learning . . . . .	70
4.3	Review of quantization schemes . . . . .	70
4.3.1	Equispaced (or uniform) quantizer . . . . .	70
4.3.2	Learning vector quantizer . . . . .	71
4.3.3	Learning Approximate Lloyd-Max quantizer . . . . .	72
4.3.4	Lloyd or $k$ -means quantizer . . . . .	73
4.4	Quantizer designs for federated learning . . . . .	73
4.4.1	Proportional weighted aggregation . . . . .	74
4.4.2	Number based weighted aggregation . . . . .	77
4.4.3	Probability score weighted aggregation . . . . .	78
4.4.4	$k$ -means based aggregation . . . . .	80
4.5	Simulations and experiments . . . . .	80
4.5.1	Simulations on synthetic data . . . . .	81
4.5.2	Simulations on fitness dataset . . . . .	82
4.5.3	Simulations on MNIST dataset . . . . .	86
4.6	Summary . . . . .	88
<b>5</b>	<b>Algorithms for Overpredictive Signal Analytics in Federated Learning</b>	<b>89</b>
5.1	Background . . . . .	90
5.1.1	Prior literature . . . . .	92
5.2	System model and background concepts . . . . .	94
5.2.1	Distributed signal model . . . . .	94
5.2.2	Fourier basis representation . . . . .	95
5.2.3	Mathematical formulation and related definitions . . . . .	95
5.2.4	Distance measures of interest . . . . .	96
5.3	Algorithms for overpredictive analytics . . . . .	96
5.3.1	Envelope approximation at the clients . . . . .	97
5.3.2	Aggregation algorithms at the server . . . . .	97
5.3.3	Algorithm sketch for over-predictive signal analytics . . . . .	98
5.4	Performance bounds on optimal envelope approximation . . . . .	99

5.4.1	A naïve envelope approximation scheme . . . . .	99
5.4.2	Envelope approximation analysis . . . . .	99
5.4.3	CDF of the envelope signal approximation . . . . .	100
5.4.4	Effect of subsampling on CDF estimate . . . . .	101
5.5	Experimental results and discussion . . . . .	102
5.5.1	Simulations on the electricity consumption dataset . . . . .	102
5.5.2	Communication cost and approximation error tradeoff . . . . .	104
5.5.3	Cumulative distribution function based signal analytics . . . . .	105
5.5.4	Effect of subsampling . . . . .	108
5.6	Summary . . . . .	111
<b>6</b>	<b>Conclusions and Future Research Directions</b>	<b>113</b>
6.1	Concluding remarks . . . . .	113
6.2	Directions for future research . . . . .	116
6.2.1	Approximate vector quantization . . . . .	116
6.2.2	Sample complexity of LALM and LAEQ . . . . .	117
6.2.3	Extension of FL methods for generic non-IID signal sources . . . . .	117
6.2.4	Privacy versus utility tradeoff in FL . . . . .	118
<b>Appendix A</b>	<b>: Supplement to Chapter 2</b>	<b>121</b>
A.1	Properties 1-6 for illustration in Sec.2.4.2 . . . . .	121
A.2	Existence of real root for ALM updates . . . . .	122
A.3	Properties of the limiting matrix $P^*$ . . . . .	123
A.4	Proof of the ALM Convergence Theorem 2.1 . . . . .	124
A.5	Proof of Near-LM Optimality Theorem 2.4 . . . . .	127
A.5.1	Proof extension to show order optimality of ALM . . . . .	131
<b>Appendix B</b>	<b>: Supplement to Chapter 3</b>	<b>133</b>
B.1	Implementation of wavelet density estimation . . . . .	133
B.2	SANE quantizer design based on mean squared error minimization . . . . .	134
B.3	Roots corresponding to AEQ . . . . .	135
B.4	Proof that AEQ optimality condition results in a positive derivative . . . . .	136

---

<b>Appendix C : Supplement to Chapter 4</b>	<b>139</b>
C.1 General IID datasets in federated aggregation . . . . .	139
C.2 General non-identical but independent datasets in federated aggregation .	142
<b>Appendix D : Supplement to Chapter 5</b>	<b>145</b>
D.1 Order optimality of envelope approximation . . . . .	145
D.2 Communication vs accuracy tradeoff in envelope CDF estimation . . . . .	147
D.3 Effect of subsampling on the CDF . . . . .	149
<b>Bibliography</b>	<b>151</b>
<b>List of Publications</b>	<b>163</b>



# List of Tables

2.1	Coefficients of the equation $r(u) = r_0 + r_1u + r_2u^2 + r_3u^3$ , to determine optimal level updates of ALM . . . . .	22
2.2	A comparison of different data driven quantizer design algorithms in emulator	37
2.3	A comparison of different data driven quantizer design algorithms in Android device for the Beta(4,2) signal source probability distribution . . . .	37
3.1	Performance of SANE with varying sliding window length . . . . .	55
3.2	Coefficients of the cubic polynomial equation $p(u) = p_0 + p_1u + p_2u^2 + p_3u^3$ , to determine optimal level updates of AEQ . . . . .	58
4.1	A summary of performance of various scalar quantization methods possible at the client devices, and their properties. . . . .	69
4.2	A summary of various federated aggregation schemes possible at the central server, and their communication cost. . . . .	70
4.3	Illustration of the improvement in the MSE performance of the federated schemes when number of devices increase. . . . .	83
4.4	A statistical description of the number of data samples across 14 users in the fitness dataset. . . . .	84
4.5	Mean squared error of different federated quantization schemes measured with respect to various train and test sizes. . . . .	85

5.1	A comparison of the quantiles of the true signal with the envelope approximation signal . . . . .	107
5.2	Comparison of different error metric for the $\mathcal{L}_1$ and the $\mathcal{L}_2$ cost functions, assuming $S = 2$ subsampling . . . . .	110
D.1	Bounds on the approximation errors . . . . .	147

# List of Figures

1.1	The projected growth trend in internet-connected devices for the forecast period 2015-2025 [2] . . . . .	2
1.2	Illustration of the federated learning architecture . . . . .	4
2.1	(a) Error-bitrate tradeoff for LM and ALM (b) Error performance on symmetric distributions (c) Error performance on truncated distribution . . . .	34
2.2	(a) Quantizer evolution (b) Simulation runtime of LM and ALM (c) Relative quantization levels . . . . .	35
2.3	A comparison of the CDF estimate obtained using LALM and the Glivenko Cantelli empirical CDF . . . . .	38
3.1	An illustration of TV whitespace protection contours . . . . .	42
3.2	The one-step greedy procedure for envelope quantization. . . . .	46
3.3	Performance of SANE quantizer on synthetic dataset . . . . .	53
3.4	Performance of SANE quantizer on TV whitespace dataset . . . . .	55
3.5	Performance of AEQ algorithm for Beta(4,2) distribution . . . . .	60
3.6	An example of LAEQ failure for the TV whitespace dataset with a non-unimodal probability density. . . . .	63
4.1	Federated learning architecture for the design of a global quantizer . . . . .	67

---

4.2	A comparison of different federated quantizers design schemes with 100 devices. . . . .	82
4.3	The figure illustrates the communication cost vs MSE tradeoff in quantizer design using federated learning. . . . .	84
4.4	$k$ - $k$ -means clustering performance on MNIST data when $b$ bits per pixel quantized communication is used. . . . .	87
4.5	$k$ - $k$ -means clustering performance as compared to the $k$ -means clustering implemented at individual edge devices. . . . .	88
5.1	The figure illustrates federated learning for overpredictive signal analytics in a smart city with $D$ electricity consumers. . . . .	91
5.2	The federated learning system model for distributed signal analytics. . . . .	94
5.3	The sum of signals obtained by distributed envelope approximation schemes using $\mathcal{L}_1$ and $\mathcal{L}_2$ cost functions. . . . .	103
5.4	Tradeoff between normalized root mean-squared (RMS) error and the number of Fourier coefficients transmitted per device. . . . .	104
5.5	Plot illustrating the convergence of the empirical CDFs of the envelope signal. . . . .	106
5.6	The figures depict the effect of subsampling on the $\mathcal{L}_1$ cost based envelope approximation. . . . .	109

# List of Abbreviations

<b>5G</b>	Fifth Generation
<b>ADC</b>	Analog to Digital Converter
<b>AEQ</b>	Approximate Envelope Quantizer
<b>ALM</b>	Approximate Lloyd-Max Quantizer
<b>CDF</b>	Cumulative Distribution Function
<b>EM</b>	Expectation Maximization
<b>ES</b>	Equispace Quantizer
<b>EV</b>	Electric Vehicles
<b>FL</b>	Federated Learning
<b>HM</b>	Heronian Mean
<b>IID</b>	Independent and Identically Distributed
<b>IoT</b>	Internet of Things
<b>IoV</b>	Internet of Vehicles
<b>LAEQ</b>	Learning Approximate Envelope Quantizer
<b>LALM</b>	Learning Approximate Lloyd-Max Quantizer
<b>LM</b>	Lloyd-Max algorithm
<b>LVQ</b>	Learning Vector Quantizer

<b>M2M</b>	Machine to Machine
<b>MAE</b>	Mean Absolute Error
<b>MATLAB</b>	Matrix Laboratory
<b>MB</b>	Mega Bytes
<b>MISE</b>	Mean Integrated Square Error
<b>MNIST</b>	Modified National Institute of Standards and Technology database
<b>MSE</b>	Mean Squared Error
<b>PDF</b>	Probability Density Function
<b>RMS</b>	Root Mean Squared error
<b>SANE</b>	Stochastic Approximation based Envelope Quantizer
<b>SAX</b>	Symbolic Aggregate Approximation
<b>SGD</b>	Stochastic Gradient Descent
<b>SOM</b>	Self Organizing Maps
<b>SQNR</b>	Signal to Quantization Noise Ratio
<b>SRAM</b>	Static Random Access Memory
<b>SVM</b>	Support Vector Machine
<b>UAV</b>	Unmanned Aerial Vehicle
<b>VC</b>	Vapnik Chervonenkis dimension
<b>WDE</b>	Wavelet Density Approximation

# List of Symbols

We have listed below some notations used globally in this thesis. Additional notations are introduced within the chapters as required.

## Vector

---

$\vec{a}$	a column vector
$a_k^{(i)}$	$k^{\text{th}}$ component of vector $\vec{a}$ at iteration $i$
$\ \cdot\ _1$	$\ell_1$ -norm of a vector
$\ \cdot\ _2$	Euclidean norm of a vector
$\ \cdot\ _\infty$	Infinity norm of a vector
$\text{Diag}\{\cdot\}$	Diagonal matrix formed with entries of a vector on the diagonal

## Matrix

---

$A_{ij}$	$(i, j)^{\text{th}}$ entry of matrix $A$
$A_i$	$i^{\text{th}}$ column of matrix $A$
$A_{\mathcal{S}}$	Set of columns of matrix $A$ indexed by set $\mathcal{S}$
$(\cdot)^T$	Transpose of a matrix
$\det\{\cdot\}$	Determinant of a matrix
$\text{Rank}\{\cdot\}$	Rank of a matrix

## Number Sets

---

$\mathbb{Z}$	Set of integers
$\mathbb{Z}_+$	Set of non-negative integers
$\mathbb{R}$	Set of real numbers

**Probability**

---

$\mathbb{P}\{.\}$	Probability of an event
$\mathbb{E}\{.\}$	Expectation of a random variable
$f_X(.)$	PDF of a random variable $X$
$F_X(.)$	CDF of a random variable $X$

**Miscellaneous**

---

$ \cdot $	Cardinality of a set
$(.)^c$	Complement of a set
$\cup$	Union of sets
$\cap$	Intersection of sets
$\mathbf{0}$	All zero vector or matrix
$\mathbf{1}$	All ones vector
$\mathbf{I}$	Identity matrix
$\mathbb{1}_{\{.\}}$	Indicator function
$\lambda_P$	Perron eigenvalue
$\mathcal{O}(.)$	Big O notation
$o(.)$	Little O notation

# Chapter 1

## Introduction

*“If we did all the things we are capable of, we would literally astound ourselves.”* - Thomas A. Edison [4]

### 1.1 Information processing at edge devices

The Internet of Things (IoTs) is driving several technology innovations, by equipping things (devices) with necessary communication and computation power and appropriate protocol stacks to communicate with a server. These IoTs, for example, will enable smart cities, through pervasive monitoring, automated actuation and optimized feedback. Distributed sensing applications involving monitoring structural health of buildings, waste management systems, noise monitoring, air quality sensing, traffic congestion management, smart lighting, and city energy consumption can be seamlessly managed and operated from remote locations in these emerging smart cities [5]. IoTs will assist governments, regulators, and businesses to derive actionable insights that allow end-users to optimize system processes and maximize performance.

We see a massive upsurge in the number of devices connected to form the IoT infrastructure and the amount of data generated at the edge devices. In Cisco’s Internet Report whitepaper, 2018-2023, the machine-to-machine (M2M) IoT segment is projected to grow from 30% to 50%. Connected home applications will have the largest share, and connected cars will be the fastest-growing application type in the forecast period (2018-2023). With the emerging 5G rollout, about 1.4 billion internet-connected things will be

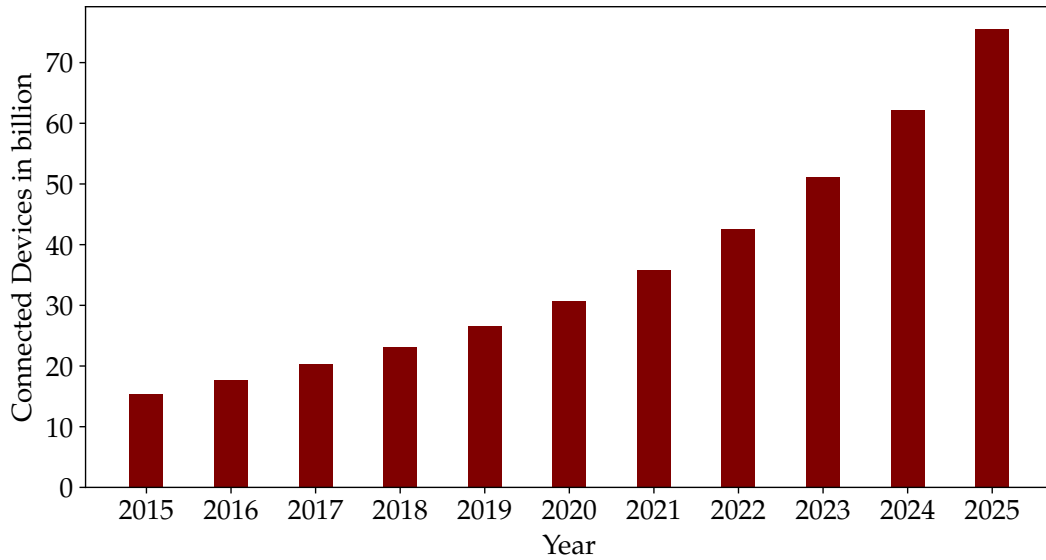


Figure 1.1: The projected growth trend in internet-connected devices for the forecast period 2015-2025 [2]

added to the current IoT infrastructure worldwide [6]. The projected growth trend in IoT connected devices for 2015-2025 is shown in Figure 1.1.

Edge computing is envisaged as a core enabler of IoT as it allows processing at the edge where the devices collect signals. Therefore, not all sensor signals need to be sent to the cloud, reducing communication costs and improving robustness and privacy. Notably, we can think about IoTs enabled with edge intelligence as distributed signal processors capable of putting the signal analytics near the signal sources. In this way, each edge device can learn its own customized signal models from all observed signal samples and only send out aggregated analytics. There are several reasons to prefer edge processing over cloud processing, and these range from latency and bandwidth cost to robustness and resilience, and to privacy and data protection. By enabling intelligence at the edge devices, we reduce the communication cost and preserve data privacy.

## 1.2 Distributed processing and inference in IoTs

The distributed sensing paradigm of IoTs opens up several applications such as environmental monitoring, industrial automation, smart homes, energy consumption monitoring, Internet of Vehicles (IoVs), healthcare, and several other use cases. In particular, IoT networks for the environment have gained popularity due to rapid urbanization in both

developed and developing countries. These intelligent networks, for example, measure air quality, humidity, temperature, ambient noise levels, and earthquakes. With the push for ensuring greener living spaces, more cities are stepping up to provide accurate distributed environmental sensing. Fixed and mobile sensors are deployed around the observed area, and the pollutant concentration data are continuously collected and relayed to the city pollution regulators for processing. For ease of deployment, most sensors use wireless communications to relay sensor data and are often battery or solar-powered. Thus, IoTs provide a fertile space for research in distributed signal processing and allied areas such as signal modeling, analytics, and inference.

The distributed nature of IoT network brings several challenges in both implementation and deployment. The key challenges are identified as [7, 8]:

- Low-power and limited compute resources
- Low-bandwidth communication channel
- Data aggregation methods
- Privacy and data protection
- Large-scale deployment of distributed sensors

To address these challenges, we need innovative solutions capable of performing low-power signal processing on resource-limited edge devices. The communication constraint arising from low-power and low bandwidth necessitates a parsimonious signal representation extracted from the raw signals. These distributed signal representations need to be combined using novel data aggregation methods to achieve targeted inference performance. Often IoT networks are vulnerable to expose privacy-sensitive data. Applications involving data from health records, energy consumption reports, smart assistants, etc., are susceptible to privacy. We need to address this challenge by communicating only a processed summary of the raw data, which is minimal for performing the inference task. The distributed nature of IoTs is yet another challenge for large-scale adoption and deployment. However, crowdsourcing solutions using sensors attached to moving objects such as buses,

motorbikes, and bicycles offer favorable options to scale distributed coverage of the IoT deployment.

A natural choice for implementing the IoT networks, with a distributed client-server model, is through *Federated Learning* (FL). This distributed learning architecture has been proposed by McMahan et al. [1], by considering the challenges discussed above. FL is part of the more general framework of “*bringing the code (algorithms) to the data, instead of the data to the code (algorithms)*” and addresses the key challenges of the locality of data, privacy, and communication resource constraints. In other words, FL promotes signal processing at the participating edge devices through energy-efficient algorithms and aids in updating a global model maintained by the server. Only these updates are communicated, thereby ensuring efficient communication complying with the principle of *data minimization* [9]. Thus, by choosing the FL architecture, we can organically address the inherent communication and privacy challenges of IoTs.

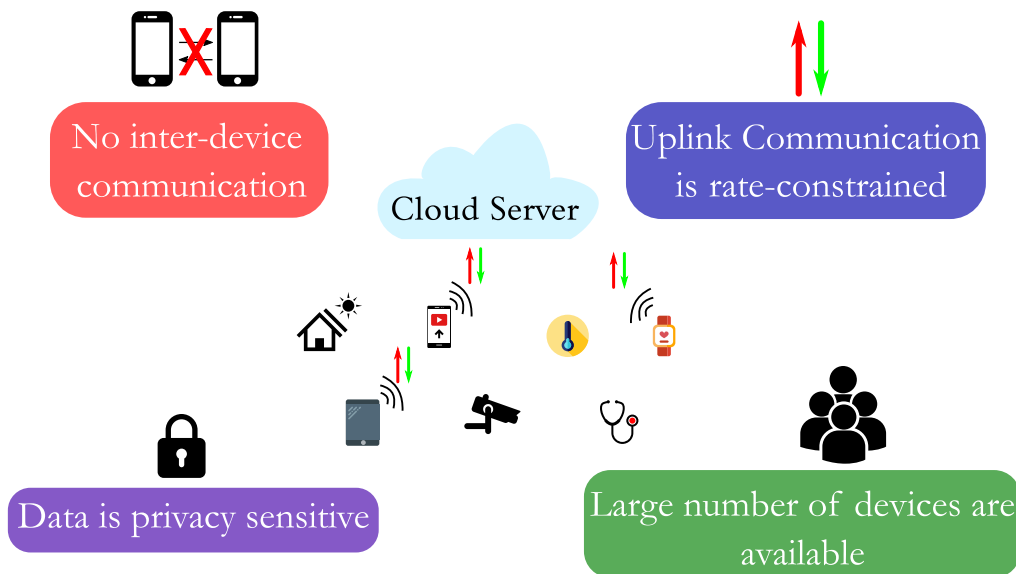


Figure 1.2: The Federated Learning (FL) architecture consisting of several participating edge devices and a central server. The figure summarizes the key aspects of FL.

FL, although having distributed processing units, is different from the distributed learning framework in data centers. The traditional data centers still have raw data available at a single location, and server-client communication is possible through reliable links. However, distributed clients (i.e., edge devices) in FL often communicate with the central server through unreliable wireless links. Additionally, the signals sensed at clients

are more privacy-sensitive compared to the data center framework. More generally, the edge devices in FL have heterogeneous compute and communication resources, i.e., these devices could have different hardware configurations depending on the manufacturers. The spatial distribution of the edge devices also implies non-identical signal source probability distribution, which adds to the existing challenges. In summary FL approach demands a holistic optimization framework that considers the following aspects (also see Fig. 1.2).

- **Limited Communication.** Edge devices frequently have rate-constrained uplink connections.
- **Heterogeneous Devices.** There are disparities in the (training) data volume available for distributed learning and inference. Moreover, the devices have different hardware capabilities depending on the manufacturer.
- **Massively Distributed.** The number of participating edge device is often larger than the number of training data samples per device.
- **Non Identical Distribution.** The signal source distribution of the local dataset at edge devices is heavily dependent on usage patterns and thus non-identical across devices.

Low power devices are vital for the widespread adoption of FL in IoTs. These edge devices can belong to the category of embedded systems, wearables, or smartphones that are primarily battery-powered. Apart from the small device dimensions, these devices are constrained by processor capabilities and storage. For instance, a class of embedded machine learning devices called Tiny machine learning or TinyML devices can perform on-device sensor data analytics at extremely low power, typically in the mW range and below, enabling a variety of always-on use-cases in battery operated devices. A typical TinyML device (like Arduino Nano 33 BLE Sense) has a program memory (flash) of the order of few Megabytes (MB) and working memory (SRAM) of a few hundred kilobytes. Thus pervasive sensing using TinyML poses a challenging landscape that motivates the need for efficient and optimized on-device signal processing. Quantization and sparse signal representation schemes relying on the signal source statistics are the desired ways to customize edge intelligence into these low-power devices.

Building from the challenges posed by the FL architecture, we formulate the following key research questions:

- Q1. *Is it possible to design scalar quantizers with nearly optimal performance under computational resource constraints at the edge devices?*
- Q2. *Can the edge device quantizer design accommodate system or application-specific constraints, such as signal overprediction, in addition to resource constraints?*
- Q3. *Does the edge processing algorithms scale with the number of devices? If yes, what is the tradeoff between inference accuracy and communication cost?*
- Q4. *Can we design signal representations for learning signal analytics in network capacity planning applications using federated learning?*

We answer the above questions in the affirmative. In the following section, we will explore these research questions in much more detail by considering the attributes of the federated learning model and the resource constraints in the low-power edge devices.

### 1.3 Scope and contributions of the thesis

The research in this thesis is generously supported by IIT Bombay and the Bharti Center for Communication (Electrical Engineering Department). The application problems which led to the research questions posed in the previous section are listed below:

- **Lloyd-Max quantization in resource constrained devices.** Quantization is a fundamental signal processing operation for signal acquisition, storage, and communication. The traditional Lloyd-Max ( $k$ -means) quantizer, although optimal, is computationally intensive due to the centroid computation step. The goal here is to develop a data-driven quantizer design that is nearly optimal and computationally feasible in resource-constrained hardware.
- **Electricity Network Capacity Planning in Smart Cities.** Smart energy meter deployments will enable tracking of energy consumption in upcoming smart cities. The city planner may desire to estimate the instantaneous energy demand

by collecting signal representations from the household smart meters. These signal representations must be designed so that the estimated demand will always be greater than the actual demand.

The above IoT use-cases provide the platform to probe further into deeper aspects of quantization and signal representation. We next pose the subquestions related to Q1, Q2, Q3, and Q4 that we have answered in this thesis. Our idea here is to highlight a selection of interesting FL-related problems without explicitly detailing the mathematical model.

In order to reduce the computational cost while designing quantizers at edge devices, it is imperative to adopt efficient approximation methods. Since the Lloyd-Max is the optimal quantizer for a known signal source probability distribution, we will build approximation algorithms based on this framework. By approximating the known signal source density, we speedup the quantizer level updated iteratively. A more interesting question to ask here is whether we can update the quantizer in a data-driven manner. Convergence and optimality of the approximate design procedure are natural questions, which we pose as extensions of Q1:

- Q1.1. What effect does piecewise linear approximation of the signal source probability density have on the mean squared error performance of the Lloyd-Max quantizer?
- Q1.2. Does the mean squared error converge w.r.t number of iterations for the piecewise linear approximation scheme? If yes, what is the convergence rate?
- Q1.3. How is the optimality of the Lloyd-Max design affected by the approximation? What is the relationship with the number of quantizer levels?

We will answer the above subquestions assuming that the signal source is continuous and the probability density is differentiable. We analyze the quantizer for a known statistical model and later extend it for a data-driven quantizer design.

Certain IoT applications will require specific cost functions to suit system requirements. Overprediction or underprediction of the quantized signals is one such system requirement. For instance, the ADC design for the depth estimation application in autonomous vehicles will require the quantized estimate of the depth signal to be underes-

estimated [10]. If  $x$  is the actual depth signal, and  $Q(x)$  is the quantized estimate of the depth, then  $x \geq Q(x)$ . This is an important design criterion in order to maintain inter-vehicle distance for collision avoidance. Other applications, for example, TV whitespace geolocation-based spectrum allocation, require a quantized representation of relevant geolocation distance signal to be overestimated. The goal here is to facilitate these system requirements while considering the inherent resource constraints. To this end, we pose the following subquestions for Q2:

Q2.1 How to extend the “approximate Lloyd-Max” type quantizer design to handle overprediction and underprediction constraints?

Q2.2 Can the quantizer design with system constraints be performed using an online update algorithm?

The first subquestion is addressed using the extension of the piecewise linear probability density approximation developed as part of the solution to Q1. The second subquestion is addressed using a variant of stochastic approximation.

Next, we look into the scalability of the quantizer design with the number of IoT devices in the federated learning setting. We analyze various quantizer design schemes along with different aggregation methods for the case of non-identical but independent distributed datasets across edge devices. Notably, we are interested in performing these quantizer designs for low-power devices with privacy and communication constraints. This leads to the following subquestions for Q3:

Q3.1 What is the tradeoff between the communication cost and mean squared error of the federated quantizer ?

Q3.2 By how much or by what factor does the mean squared error performance improve upon choosing the federated learning approach?

Using the approximately optimal Lloyd-Max quantizers discussed as part of Q1, we characterize the fundamental tradeoff between communication cost and mean squared error. This subquestion is answered based on experiments on an available fitness tracker dataset.

Further, the solution to second subquestion follows from a mathematical analysis hinging on the linearity of the aggregation function and the convexity of the mean squared error cost function. We also demonstrate the efficacy of the designed quantizer and federated aggregation on an image classification example.

Finally, we consider the application of federated learning in electricity network capacity planning for smart cities. As motivated in the beginning of this section, the city planner (server) desires to estimate an overpredictive estimate of the cumulative consumer demand time-series based on the smart-meter recordings and each consumer (edge device). Due to the communication constraints the consumer side edge devices only sends a concise signal representation, considering the signal overprediction constraint. At the server, the city planner can derive several useful signal analytics based on the reconstructed time-series. The related subquestions in Q4 are:

Q4.1 What algorithms can be implemented at the consumer edge devices to determine a signal representation considering the signal overprediction constraint?

Q4.2 Can we suggest analytical bounds on the error performance of the signal representations derived from these algorithms?

Q4.3 How does the estimate of the signal analytics vary with the number of signal approximation coefficients (communicated by the edge devices) ?

In this exposition to Q4, we use the Fourier basis representation to build algorithms for signal overprediction. We rely on mean-absolute error (or  $\mathcal{L}_1$ ) and mean squared error (or  $\mathcal{L}_2$ ) cost functions and provide analytical guarantees on the error performance for the class of  $p$ -times differentiable signals. These bounds depend on the number of communicated Fourier coefficients and the sampling rate chosen while recording the energy consumption time-series. To answer Q4.3, we estimate the empirical Cumulative Distribution Function (CDF) from the reconstructed time-series of individual consumers and analyze the pointwise difference of the estimate with the actual CDF using concentration bounds. To verify the above analytical guarantees, we present experiments findings on an available residential energy consumption dataset. We bring out the communication vs. accuracy tradeoff inherent in the FL approach and probe into subsampling effects.

## 1.4 Summary of results and outline of this thesis

The rest of the thesis is organized into three parts. In the first part of the thesis (i.e., in Chapter 2-3), the algorithms and analysis focussed on resource-constrained edge devices is discussed in detail. This part handles the subquestions Q1.1-Q1.3 and Q2.1-Q2.2. In the second part (i.e., in Chapter 4-5), we address applications of federated learning concerning the subquestions Q3.1-Q3.2 and Q4.1-Q4.3. Finally, the thesis concludes with the third part (i.e., Chapter 6), where we pose some interesting open problems for future research. The list of publications from this thesis is available at the end of the thesis. Below we provide the outline of the chapters by summarizing the main results.

- **Chapter 2** on *Approximate Lloyd-Max* (ALM) quantizer design provides the foundation to develop quantizers for low-power resource-constrained edge devices. Utilizing a piecewise linear probability density approximation, we show that the quantizer updates can be performed by solving the roots of a quadratic equation. The ALM design, which assumes the probability density function to be known, is then extended to the data-driven case termed as *Learning ALM* (LALM). LALM is agnostic to the source probability distribution and is shown to converge to ALM in probability. The subquestions Q1.1-Q1.2 related to mean squared error convergence is addressed with the help of mathematical analysis that relies on Perron-Frobenius theory. For Q1.3 on the optimality of ALM, we show that the ALM quantizer levels converge to a point “near” the Lloyd-Max quantizer. The degree of “nearness” is characterized mathematically for a chosen number of quantizer levels. For the LALM, we comment on the sample complexity of the quantizer design, taking insights from statistical learning theory.
- **Chapter 3** is focussed on the *overprediction system constraint* which requires the quantized signal to be always greater than the actual signal. For brevity, we call the overprediction criterion as the *envelope constraint*. To this end, we propose two approaches, the first that relies on stochastic approximation and the second that is based on the piecewise linear approximation of the signal source density. The Stochastic Approximation based Envelope (SANE) quantizer is a counterpart of the Learning Vector Quantizer (LVQ) algorithm, a well-known clustering technique.

We propose an online update method using a two-time stochastic gradient descent algorithm, thus answering Q2.2 in the affirmative. Experiments on an available TV whitespace geolocation dataset are used to study the convergence property of the SANE quantizer. The second approach mentioned above is a clone of the ALM algorithm (studied in Chapter 2) with the additional envelope constraints. Akin to ALM, the Approximate Envelope Quantizer (AEQ) provides quantizer level updates based on polynomial root solving and exhibits similar convergence properties (subquestion Q2.1). Learning AEQ (LAEQ) is the data-driven equivalent of AEQ, which also learns the piecewise linear approximate probability density.

- **Chapter 4** is dedicated to the discussion of *distributed quantizer design* in federated learning. Building from known quantizer design algorithms such as equispaced (uniform), LALM, LVQ, and  $k$ -means, we suggest federated aggregation methods based on linear weighted functions. For the class of non-identical but independent distributions assumed over the edge devices, we analyze the mean-squared error performance of the global quantizer along with the quantizers at individual devices. The federated quantizer offers better mean-squared error performance compared to all the individually designed quantizers (thus answering Q3.2). To probe further into the engineering tradeoffs posed in Q3.1, we perform experiments on a canned fitness tracker dataset. We empirically characterize the mean-squared error (MSE) vs. communication cost tradeoff for the developed federated quantization schemes. Finally, we examine the vector extension of the federated quantizer in the MNIST image classification dataset and observe the improvement in clustering accuracy obtained by scaling the number of edge devices.
- **Chapter 5** addresses subquestions of Q2, related to *Overpredictive Signal Analytics in Federated Learning*. Signal representations with an overpredictive constraint are desired in applications such as electricity network capacity planning. We propose algorithms to determine the envelope constrained signal representation using  $\mathcal{L}_1$  and  $\mathcal{L}_2$  cost functions with the Fourier basis representation. We also compute the analytical upper bound on the signal approximation error for these algorithms, thereby answering Q2.2. We consider two signal analytics at the central server, viz. (i) sum of signals at the edge device and (ii) CDF of the individual time-series

signals. By relying on concentration bounds, we derive upperbounds on the CDF estimates derived from overpredicted signal representations. Experimental studies were conducted on a public residential electricity consumption dataset consisting of several consumer time-series data over 30 days. It is observed that the  $\mathcal{L}_1$ -optimized signal approximation has better mean squared error performance as compared to  $\mathcal{L}_2$ -optimized signal approximation. On the other hand, the  $\mathcal{L}_2$  envelope is more resilient to envelope errors when subsampling is performed. We also characterize the effect of the number of communicated signal coefficients on the Wasserstein distance between the estimated CDF and the actual CDF, therefore addressing Q4.3. Finally, we evaluate the tradeoff between Wasserstein distance and the subsampling rate chosen at the edge devices.

- **Chapter 6** summarizes the thesis and provide directions for future research. We raise open research questions related to extension to vector quantization and the feasibility of near optimal design in resource constrained edge devices. Further, we provide some insight into analyzing sample complexity of LALM and LAEQ. In the context of FL, we outline possible challenges while extending the existing quantizer design algorithms for generic non-IID datasets. Finally, we provide a flavor of inherent privacy challenges in FL and motivate the necessity of a mathematical framework to analyze the privacy vs. utility tradeoff.

# Chapter 2

## Speeding up of Scalar Quantizer Design for Edge Signal Processing

In this chapter, we will discuss quantizer design for resource-constrained edge devices using signal processing focussed on the efficient use of energy, communication, and hardware resources. We propose a model-driven approach, termed Approximate Lloyd- Max (ALM) design, based on a piecewise linear approximation of the signal-source probability density to address these resource challenges. From the principles of the ALM design, we develop a data-driven quantizer, or Learning ALM (LALM), using statistical learning methods. We show convergence of the ALM quantizer near the limit of the Lloyd-Max quantizer and characterize its convergence rate. Using an Android-based edge device, we compare the performance of LALM with traditional  $k$ -means in terms of algorithm speedup, energy usage, and memory utilization.

### 2.1 Background

In emerging machine learning applications, edge signal processing is done in battery constrained low power devices such as smart phones or wearables. To reduce the communication cost, these devices are required to perform on-device signal quantization with reliable performance guarantees [11]. Optimal quantizer design at the edge device is challenging in both model-driven (where the signal source probability density is known) and data-driven cases, due to the battery constraint (or equivalent energy constraint). In the

model-driven case, classical quantizer design based on Lloyd-Max (LM) algorithm, will incur higher computational cost due to the integration step for computing the centroid (conditional mean). Similarly, for known data-driven quantizer design techniques such as  $k$ -means, the common criticisms are over its high memory utilization [12, 13], and relatively large convergence time [14]. These attributes of the existing quantizer designs are unfavorable for edge devices, and therefore require novel design algorithms.

In this work, we propose a model-driven solution for quantizer design termed *Approximate Lloyd-Max (ALM)*, which will be further developed to function in a data-driven case. ALM quantizer design allows to bypass the integration step in the LM algorithm, by performing a *piecewise-linear approximation of the source probability density* [15]. We show that the ALM design involves *nearly-linear* update steps, leading to speedup and computational advantages, in addition to hardware reduction and energy saving. Besides these, we show the ALM algorithm converges to a limit point near the quantization levels of the optimal LM algorithm. The convergence rate of ALM design compares with the LM algorithm, and we show this through mathematical analysis.

For edge signal processing applications, where the signal source probability distribution is unknown, ALM quantizer design is modified to a data-driven setup. This is termed as *Learning Approximate Lloyd-Max (LALM)* algorithm in the chapter. Unlike the ALM model, the LALM quantizer is *oblivious* to the signal source probability density, and hence trained using the signal observations alone. The proposed learning algorithm uses statistical approaches that convert the data samples to a *sparse representation*, in the form of piecewise-linear probability density parameters. This approach leads to both space and sample complexity reduction as compared to the  $k$ -means algorithm. In this work, we also establish the (weak) consistency of the LALM algorithm with ALM; that is, the quantizer levels of LALM method converges to the ALM levels as the number of training samples become large. Added to this, the LALM design learns the probability density function (PDF) of the source, useful in deriving signal analytics and statistics.

*Remark:* While designing the quantizers we have considered fixed rate encoding, where each quantization level is represented by a fixed number of bits. Although variable rate encoding is known to minimize the entropy, it requires coding over multiple blocks [16,

17, 18]. This is especially undesired in edge devices, given the energy and hardware constraints. Further, in multi-user systems such as federated learning with several edge nodes, the decoder design at the central server becomes complex and often impossible in real-time applications. Considering these facts, we adhere to the fixed rate encoding model throughout this chapter.

The focus in this work has been to optimize edge signal processing [19, 20, 21] for enabling distributed learning applications such as federated learning. In this first exposition, we have restricted the scope of discussions to the design of near-optimal quantizers at individual edge devices. Thus, the aspects of federated aggregation and orchestration [1] for determining a global quantizer will be considered in a future work. Further, we note that higher order polynomial approximations (including other non-linear kernel based methods), instead of piecewise linear approximation, would result in better probability density approximation, with increase in the cost of computations. We demonstrate that piecewise linear probability density approximation lead to a near-optimal quantizer design for the class of signal sources with continuously differentiable PDFs.

The main contributions are summarized below.

- For resource-scarce edge devices, we propose an Approximate Lloyd-Max (ALM) quantizer, which reduces the computational complexity of Lloyd-Max quantizer.
- A data-driven, Learning ALM (LALM) quantizer is designed, for edge devices that has no a priori signal source probability density. LALM design also learns the approximate signal source PDF, useful for deriving several signal analytics.
- For the class of differentiable source PDF, we show that the ALM quantizer converges at an exponential rate to a fixed point near the LM quantizer.
- Using simulation study, we demonstrate the speedup obtained for ALM algorithm for various bounded support signal source distributions. Experiments performed on an Android device shows the lower energy and memory utilization of LALM when compared to  $k$ -means.

An important contribution that distinguishes this work from the prior works, is the analysis framework for showing the ALM quantizer convergence for the finite rate (i.e.

non-asymptotic) regime. Since the ALM quantizer updates are almost linear, we can express each level update step as a linear (matrix) transformation. This enables us to invoke the Perron-Frobenius theory [22, 23], which is well-known in Markov chain analysis. Specifically, we show that the ALM algorithm converges to a fixed point at an exponential (decay) rate, which depends on the Perron eigenvalues of the transformation.

### 2.1.1 Related literature

Fixed-rate optimal scalar quantization with known signal source distribution and mean square error cost function, was first studied in the independent works by Lloyd and Max [24, 25]. Sharma extended the Lloyd-Max method to a general class of convex and semi-convex distortion measures [26], by employing a combination of dynamic programming and fast search. The vector extension of LM quantizer was introduced by Linde, Buzo and Gray [27]. This quantizer is well known by the name LBG algorithm. Gray and Neuhoff have summarized the historical evolution of the quantization schemes, both scalar and vector cases, in their comprehensive review paper [28]. Ziv proposed a variable rate universal quantizer for vector data, that achieves the optimal performance within a constant gap [29]. Another well studied model is the entropy coded quantizer, which is based on mean square error minimization with an entropy constraint [17]. The convergence analysis of the LM algorithm is also extensively dealt in literature. Convergence with exponential decay rate to the global minima is known, under the assumptions of a convex cost and a log-concave probability distribution [30]. In another work, Sabin and Gray explains the absolute convergence of the Lloyd algorithm and its empirical density consistency on training data [31]. A paper by Wu shows the convergence of the Lloyd method for continuous, positive PDFs defined over a finite interval, using the concept of finite state machines [32]. Some of the above mentioned convergence results are valid for the high rate regime. However, in this work we have analyzed the convergence of the developed quantizers in all finite rate (or non-asymptotic) regimes.

Quantization using data-driven methods are relevant in (adaptive) signal processing and machine learning. Some well-known data-centric quantizers include, the  $k$ -means clustering and the LVQ [33, 34]. The  $k$ -means clustering algorithm was first reported by MacQueen [33]. It has been extensively used in statistical learning and pattern recogni-

tion [35]. The convergence analysis of  $k$ -means using notions of uniform consistency has been broadly examined by Pollard [36, 37]. The most acclaimed version of  $k$ -means, known as Elkan’s  $k$ -means, is prominent in many software packages [12]. Learning Vector Quantization (LVQ) is another popular stochastic approximation based algorithm proposed by Kohonen [34]. The method uses a *competitive learning* approach, which is suitable for online (learning) applications. For comparison, we have used the scalar version of LVQ in this work. Design of adaptive scalar quantizers using piecewise linear approximations has been studied for dynamic signal sources by Ortega and Vetterli [38]. The authors suggest an adaptive quantizer design by using mean-squared error optimization with a linear regularizer, tuned using a hyperparameter. In addition, the work provides guarantees on signal to quantization noise (SQNR) for a class of image signals, based on experiments.

Recent research works in machine learning have stressed on edge intelligence for improving communication efficiency and privacy [19, 20, 21]. The federated learning architecture [39, 1, 40] promises several solutions for addressing challenges in edge signal processing and machine learning. In wireless communication, channel aware quantizer design for learning deep neural network models have seen recent interests [41, 42]. Another related domain is TinyML, where several ultra-low-power IoT devices are deployed for data collection [43]. Computationally efficient quantized neural networks have emerged as an important candidate in such applications [11, 44, 45].

**Organization:** The source and quantizer model are introduced in section 2.2. In section 2.3, we develop the cost function, optimality criteria and update rule corresponding to nearly optimal quantizers, *viz.*, the model-driven ALM and the data-driven LALM. This section also treats the consistency and sample complexity aspects of the LALM quantizer. The main result, showing the ALM convergence theorem is presented in section 2.4. Further, in section 2.5, the asymptotic optimality of the ALM quantizer is considered. Section 2.6 discusses the experiment results, and finally concluding remarks are mentioned in section 2.7.

## 2.2 Signal source and quantizer models

Consider a signal source having a continuous probability density  $f_X(x)$ . In this chapter, we examine two separate signal source models. The first one assumes the signal probability

density is *known* at the source, while the second is *oblivious* to the underlying probability density. It is presumed that  $f_X(x)$  is positive, differentiable and supported on a finite interval  $\mathcal{I}$ . Without loss of generality, we consider the unit interval,  $\mathcal{I} = [0, 1]$ . In addition to this, the following *smoothness condition* is assumed,

$$|f'_X(x)| \leq m \in [0, \infty) \quad \text{for all } x \in \mathcal{I}. \quad (2.1)$$

The derivative condition ensures that the slope of the probability density is bounded and hence the probability density is smoothly varying.

Hereafter we describe the quantizer system model. Depending on the knowledge about the signal source, there are two kinds of quantizer models. The first is termed as the *model-driven* quantizer, which has complete knowledge of the underlying source probability density. The second kind is the *data-driven* quantizer, which is oblivious to the source probability density. We denote the two quantizer models using the quantizer functions  $\mathcal{Q}_M(\cdot, \cdot)$  and  $\mathcal{Q}_D(\cdot, \cdot)$  respectively. A fixed length quantizer (with finite bit allocation) having  $K$  representative levels is assumed. Each level is encoded at the quantization rate given by  $R = \log_2 K$ . For ease of notation, we represent the quantization levels using the vector  $\vec{q} := [q_1, q_2, \dots, q_K]$ . These levels are ordered as,  $q_1 < q_2 < \dots < q_K$ .

Quantization performance is measured using the mean squared error (MSE) distortion. For the model-driven quantizer, the distortion is

$$\mathcal{D}_{\mathcal{Q}_M}(f_X) := \mathbb{E} [(\mathcal{Q}_M(X, K) - X)^2]. \quad (2.2)$$

For the data-driven quantizer, we employ the empirical distortion measure for the data  $X_1, X_2, \dots, X_N$  given by,

$$\mathcal{D}_{\mathcal{Q}_D}(X_1, X_2, \dots, X_N) := \frac{1}{N} \sum_{i=1}^N (\mathcal{Q}_D(X_i, K) - X_i)^2 \quad (2.3)$$

A quantizer,  $\mathcal{Q}^*$  is called (*globally*) *optimal* if it results in the *minimum distortion* among all quantizers chosen from the *feasible set*,  $\mathcal{S}$ . That is,  $\mathcal{Q}_M^* = \arg \min_{\mathcal{Q}_M(\cdot, K) \in \mathcal{S}} \mathcal{D}_{\mathcal{Q}_M}(f_X)$ , when the quantizer is model-driven. For a data-driven quantizer  $\mathcal{Q}_D^* = \arg \min_{\mathcal{Q}_D(\cdot) \in \mathcal{S}} \mathcal{D}_{\mathcal{Q}_D}(X_1, \dots, X_N)$ . Further, we describe the near optimality property, which will be used in the course our work. Two quantizers defined by the functions  $\mathcal{Q}(\cdot)$  and  $\mathcal{Q}_{\text{app}}(\cdot)$  (each having  $K$  levels) are said to be *asymptotically near optimal* if  $\lim_{K \rightarrow \infty} |\mathcal{Q}(x, K) - \mathcal{Q}_{\text{app}}(x, K)| = 0$  for  $x \in \mathcal{I}$ .

The challenge in designing an optimal quantizer is equivalent to a search problem over a  $K$  dimensional space ( $[0, 1]^K$  for instance). In practice, the optimal quantizer levels are obtained via recurrent (iterative) algorithms. The efficiency of a quantizer design algorithm is thus characterized by the number of iterations required to attain convergence.

## 2.3 Nearly optimal quantizer designs

In this section, we introduce the optimization framework for the model-driven quantizer as well as the data-driven quantizer. At first we derive the optimality condition based on the MSE distortion, and later use it to develop a quantization algorithm based on the approach of piecewise linear approximation.

### 2.3.1 Approximate Lloyd-Max quantizer

The ALM algorithm is a model-driven scheme, where the signal source probability density is known. For a fair comparison with LM, we consider the signal sources having a continuous and positive probability density defined on a bounded support. The LM convergence for this class has been effectively dealt by Wu [32]. Akin to the LM algorithm, the ALM quantizer is implemented using recursive level updates. Nevertheless, the ALM design bypasses the integration operation in the centroid update step, thereby reducing the computational complexity. Moreover, it offers considerable speedup for the quantizer design, making it a favorable choice for distributed signal sensing applications.

For elucidating the analysis of the ALM, we introduce two reference levels,  $s_{\text{start}} := 0$  and  $s_{\text{stop}} := 1$ , each representing the boundary of the bounded interval. Consider the number of levels,  $K \geq 2$ . Let  $\{s_k; 1 \leq k \leq K\}$  be the quantization set (representation points) and  $\{d_j; 1 \leq j \leq K + 1\}$  be the boundary set (quantization regions). The MSE distortion for ALM optimization can be expressed as,

$$\begin{aligned} \mathcal{D}_{\mathcal{Q}_M}(f_X) &:= \mathbb{E}[\mathcal{Q}_M(X) - X]^2 \\ &= \int_0^1 (\mathcal{Q}_M(x) - x)^2 f_X(x) dx \\ &= \sum_{k=1}^K \int_{d_k}^{d_{k+1}} (s_k - x)^2 f_X(x) dx. \end{aligned} \tag{2.4}$$

The boundary set  $\{d_j; 1 \leq j \leq K+1\}$  is defined as,  $d_j = (s_j + s_{j-1})/2$  for  $j \in \{2, \dots, K\}$ ,  $d_1 := s_{\text{start}}$  and  $d_{K+1} := s_{\text{stop}}$ . The total MSE cost in (2.4) can be minimized by optimizing the MSE cost in the left and right decision neighborhood of each quantization level. Since the MSE distortion is a differentiable convex function, we perform minimization by taking partial derivatives with respect to the levels  $\{s_i; 1 \leq i \leq K\}$ . Using Leibniz rule (for differentiation under integral sign), we obtain the following optimality condition for the quantization levels [46]:

$$0 = 2 \int_{d_k}^{d_{k+1}} (s_k - x) f_X(x) dx, \quad \text{where } 1 \leq k \leq K. \quad (2.5)$$

In general, the above equation is devoid of a closed form expression for  $s_k$ . The LM algorithm solves this issue by fixing the boundary levels  $d_k$  and  $d_{k+1}$  as per the previous iterate values of  $\{s_k\}_{k=1}^K$ , followed by the centroid update step. However, the ALM uses the alternate approach of (piecewise-linear) probability density approximation to realize a *nearly optimal* quantizer. This allows us to obtain the revised quantization level,  $s_k$  by evading the integral computation. Besides this, the ALM enables to attain a closed-form solution for  $s_k$ .

The modified optimality criteria for the ALM is thus obtained by replacing the known probability density  $f_X(\cdot)$  by the piecewise linear function  $f_{\text{app}}(\cdot)$  in (2.5). For each segment a first order approximation applies. That is,

$$f_{\text{app}}(x) = m_k x + c_k, \quad \left\{ \begin{array}{l} \text{for } x \in [s_{k-1}, s_{k+1}] \\ \text{and } 2 \leq k \leq K-1, \end{array} \right\} \quad (2.6)$$

where  $m_k$  and  $c_k$  corresponds to the slope and the intercept parameters of the approximation. These parameters are determined using the endpoint conditions  $f_{\text{app}}(s_{k-1}) = f_X(s_{k-1})$  and  $f_{\text{app}}(s_{k+1}) = f_X(s_{k+1})$ . The linear approximation described above, helps us to glean a computable expression for the optimal  $s_k$ . On replacing the probability density function  $f_X(x)$  by its approximation  $f_{\text{app}}(x)$  in the optimality condition, a cubic equation,  $r(u) = r_0 + r_1 u + r_2 u^2 + r_3 u^3$  is obtained, which has a real root in the interval  $[s_{k-1}, s_{k+1}]$  (See Appendix A.2 for the proof). For  $2 \leq k \leq K-1$ , the equation becomes quadratic, as the coefficient  $r_3 = 0$ . In Table. 2.1, the coefficients  $r_0, r_1, r_2$  and  $r_3$  are tabulated for the different quantization levels. It is observed that the ALM algorithm recursively updates the quantization levels by solving the (cubic) polynomial equation. More details about

ALM quantizer is described in the algorithm sketch, Algorithm 1 and Table 2.1. For the stopping rule we consider a **Threshold** parameter which is constructed as  $x\%$  of the MSE of the equispaced quantizer, where  $x$  is chosen by the designer.

---

**Algorithm 1** ALM Algorithm
 

---

**Input:** Density function  $f_X(x)$ ,  $K = \text{No. of levels}$ , **Threshold**

**Initialize:** Set  $\vec{s}^{(0)} = [0, \frac{1}{K}, \frac{2}{K}, \dots, 1]$ , **stop condition** = FALSE, [Iteration index]  
 $i = 0$

**while** **stop condition** = FALSE **do**

- (step-1)  $\vec{s}^{(i)}$  is partitioned into odd and even sets,

$$\mathcal{Q}_{\text{odd}} = \{s_1^{(i)}, s_3^{(i)}, \dots\} \text{ and } \mathcal{Q}_{\text{even}} = \{s_2^{(i)}, s_4^{(i)}, \dots\}$$

- (step-2) All levels  $s_k^{(i)} \in \mathcal{Q}_{\text{odd}}$  are updated (*concurrently*) to the real root of  $r(u) = 0$  in  $[s_{k-1}, s_{k+1}]$  (see Table 2.1), using parameters  $m_k, c_k$  chosen according to (2.6).
- (step-3) All levels  $s_k^{(i)} \in \mathcal{Q}_{\text{even}}$  are updated (*concurrently*) similar to step 1.
- (step-4)  $i \leftarrow i + 1$  and recompute MSE.

**If** (MSE < Threshold):

set **stop condition** = TRUE

**Else:** jump to step 1.

**end while**

**Output:** Quantization levels  $\vec{s}$

---

### 2.3.2 Learning Approximate Lloyd-Max quantizer

Because ALM quantizer cannot be designed without the source probability density information, we consider a data-driven equivalent of the ALM algorithm, termed as Learning-ALM (LALM). The only input fed to the LALM algorithm is a collection of  $N$  data samples generated from an unknown source distribution. Analogous to the ALM optimization framework, the goal here is to obtain a fixed length quantizer that minimizes the empirical MSE distortion. However, since the probability density information is unknown, alternate

	$r_0$	$r_1$	$r_2$	$r_3$
$k = 1$	$\frac{m_1}{3} \left( s_{\text{start}}^3 - \frac{s_2^3}{8} \right) + \frac{c_1}{2} \left( s_{\text{start}}^2 - \frac{s_2^2}{4} \right)$	$-\frac{m_1}{2} s_{\text{start}}^2 + \frac{c_1}{4} s_2 - c_1 s_{\text{start}}$	$\frac{1}{8} m_1 s_2 + \frac{3}{8} c_1$	$\frac{1}{12} m_1$
$k \neq 1, K$	$-\frac{m^k}{24} \left( s_{k+1}^3 - s_{k-1}^3 \right) - \frac{c^k}{8} \left( s_{k+1}^2 - s_{k-1}^2 \right)$	$\frac{c^k}{4} \left( s_{k+1} - s_{k-1} \right)$	$\frac{m^k}{8} \left( s_{k+1} - s_{k-1} \right)$	0
$k = K$	$\frac{m_K}{3} \left[ \frac{s_{K-1}^3}{8} - s_{\text{stop}}^3 \right] + \frac{c_K}{2} \left[ \frac{s_{K-1}^2}{4} - s_{\text{stop}}^2 \right]$	$\frac{m_K}{2} s_{\text{stop}}^2 - \frac{c_K}{4} s_{K-1} + c_K s_{\text{stop}}$	$\frac{1}{8} m_K s_{K-1} - \frac{3}{8} c_K$	$-\frac{1}{12} m_K$

Table 2.1: Coefficients of the equation  $r(u) = r_0 + r_1 u + r_2 u^2 + r_3 u^3$ , to determine optimal level updates of ALM

statistical methods have to be devised to determine the optimal data-driven quantizer.

LALM will estimate the piecewise linear source probability density in each iteration. If  $\widehat{m}_k$  and  $\widehat{c}_k$  denote the learned parameters for the interval  $[s_{k-1}, s_{k+1}]$ , then the learned probability density is  $\widehat{f}_{\text{app}}(x) = \widehat{m}_k x + \widehat{c}_k$ . These estimated parameters are *plugged-in* to the ALM algorithm, described in Table 2.1, to learn the quantization levels. Each iteration in the algorithm will estimate the  $\widehat{m}_k$  and  $\widehat{c}_k$  parameters, as explained below.

For the interval  $[s_{k-1}, s_{k+1}]$ , we denote the probability density estimates at the endpoints as  $\widehat{f}_{k-1} := \widehat{f}_{\text{app}}(s_{k-1})$  and  $\widehat{f}_{k+1} := \widehat{f}_{\text{app}}(s_{k+1})$ . These estimates can be computed using the empirical probability and the conditional statistical mean restricted to the same interval. If  $n_{k-1}$  and  $n_k$  stands for the number of data points falling in the interval  $[s_{k-1}, s_k]$  and  $[s_k, s_{k+1}]$  respectively, then the following linear relations holds true,

$$(\widehat{f}_{k+1} + \widehat{f}_{k-1}) \frac{s_{k+1} + s_{k-1}}{2} = \frac{n_{k-1} + n_k}{N}, \quad (2.7)$$

$$\begin{aligned} \widehat{f}_{k+1} \left( \frac{s_{k+1}^2}{3} - \frac{s_{k-1}^2}{6} - \frac{s_{k+1}s_{k-1}}{6} \right) + \\ \widehat{f}_{k-1} \left( \frac{s_{k+1}^2}{6} - \frac{s_{k-1}^2}{3} + \frac{s_{k+1}s_{k-1}}{6} \right) = \frac{\sum_i X_i \mathbb{1}_{I_k}(X_i)}{N}. \end{aligned} \quad (2.8)$$

The notation  $\mathbb{1}_{\{\cdot\}}(\cdot)$ , represents the 0 – 1 indicator function and  $I_k := [s_{k-1}, s_{k+1}]$ . After solving for this two-variable linear system, the endpoint estimates of the approximate probability density are obtained. Thus, the slope and intercept parameter estimates of the piecewise-linear probability density approximation are

$$\widehat{m}_k = \frac{\widehat{f}_{k+1} - \widehat{f}_{k-1}}{s_{k+1} - s_{k-1}}, \quad \widehat{c}_k = \frac{s_{k+1}\widehat{f}_{k-1} - s_{k-1}\widehat{f}_{k+1}}{s_{k+1} - s_{k-1}}. \quad (2.9)$$

These parameters are further fed into the level update (root finding) step in the ALM design. Analogous to the model-driven ALM approach, the LALM algorithm runs until convergence condition (or stopping criteria) is met. The algorithmic sketch for LALM will follow the steps shown in Algorithm. 1, with the minor change in **step-2**, where the estimates  $\widehat{m}_k$  and  $\widehat{c}_k$  will be used in place of actual slope and intercept. The plug-in update for the  $k$ -th quantization level is,

$$\widehat{q}_k = \frac{1}{\widehat{m}_k} \left( \text{HM} \left( \widehat{f}_{k-1}, \widehat{f}_{k+1} \right) - \widehat{c}_k \right), \quad (2.10)$$

where  $\text{HM}(\cdot, \cdot)$  represents the Heronian mean function defined as [47],

$$\text{HM}(a, b) := \frac{1}{3} \left( a + \sqrt{ab} + b \right).$$

The performance of LALM algorithm can be gauged using the aspects of *consistency* and *sample complexity*. We observe that, for a large dataset, the worst-case gap between LALM levels and the ALM levels becomes negligible. This implies that, there is a growing consistency with the number of data samples. However, increasing data size comes at the cost of sample complexity. In what follows, we describe the mathematical sketch of LALM consistency, and suggest a *one-shot* LALM that will lower the sample complexity.

### Consistency of LALM quantizer

Since LALM is a data-driven algorithm, it is of interest to verify its consistency with the model-driven ALM algorithm. For consistency to hold, the LALM quantizer levels need to converge to the ALM quantizer levels, as the data sample size,  $N \rightarrow \infty$ . By invoking the law of large numbers, the empirical probability  $\frac{n_{k-1}+n_k}{N} \xrightarrow{\mathbb{P}} \mathbb{P}(X \in [s_{k-1}, s_{k+1}])$ . Similarly, the fractional sample mean  $\frac{n_{k-1}+n_k}{N} \frac{\sum_j X_j \mathbb{1}_{[s_{k-1}, s_{k+1}]}(X_j)}{n_{k-1}+n_k} \xrightarrow{\mathbb{P}} \mathbb{P}(X \in [s_{k-1}, s_{k+1}]) \times \mu_{[s_{k-1}, s_{k+1}]}$ , where  $\mu_{[s_{k-1}, s_{k+1}]}$  is the conditional mean of the approximate probability density in the range  $[s_{k-1}, s_{k+1}]$ . Since the set of equations – (2.7)-(2.8), has a unique solution, the LALM probability density parameters  $(\widehat{m}_k, \widehat{c}_k) \xrightarrow{\mathbb{P}} (m_k, c_k)$ . In other words, the probability density estimate  $\widehat{f}_k = \widehat{m}_k s_k + \widehat{c}_k$  converges to  $f_k = m_k s_k + c_k$ . Because the probability density parameters converge, the quantization levels of LALM converge to the ALM levels, which explains the consistency.

### Sample complexity reduction of LALM quantizer

From the description of LALM quantizer, it is observed that the bottleneck step is the one that involves repeated parameter estimations. This *repeated learning* of the piecewise probability density parameters slows down the algorithm execution. From an algorithmic point of view, the redundant estimation operation is unnecessary, if the source considered is stationary. To rectify this shortcoming, a *one-shot LALM*, that intermittently learns the probability density parameters, is proposed. Between any two successive probability density learning operations, the piecewise linear approximation available at that instant is used for quantization level updates. This periodic estimation procedure performs faster

when compared to conventional  $k$ -means algorithm. The speedup is attributed to the ability of one-shot LALM to encapsulate the data statistics into a sparse representation, available in the form of piecewise-linear slopes and intercepts. The frequency of one-shot ALM updates is determined by the target MSE and the speedup requirement specified for an application. Thus, the one-shot LALM has the advantage of jointly minimizing space and time complexity, as opposed to  $k$ -means where repeated distance computations are necessary. More details on one-shot LALM can be seen in the experiments section 2.6.

In literature, determining the sample complexity of optimal data-driven quantizers has been shown to be an NP-complete problem [48]. Thus most often only a finite sample upper bound is available to describe the sample complexity. Such finite sample bounds (or in other words sample complexity) of empirically optimal quantizers defined over a bounded support would require additional assumptions on the probability density class, such as finite second moment. Moreover, learning theoretic tools like Vapnik-Chervonenkis (VC) dimension is often necessary [14] to establish statistical consistency. Considering these aspects, determining the sample complexity of LALM algorithm for the general class of signal sources with continuously differentiable probability density is expected to be challenging, given the scope of the current manuscript.

In [49], an upper bound on the expected distortion gap has been characterized for the nearest-neighbor class of quantizers. For the  $k$ -means quantizer, the expected distortion gap with respect to the LM quantizer is given by an approximate upperbound,  $C \times \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ , where constant  $C$  is determined from the VC dimension of the probability density function class and  $n$  is the number of data samples. Since the *one-shot* LALM (which is a nearest neighbor quantizer) acts as a bounded plug-in estimate over the approximate probability density function, we expect similar upperbound result to characterize the expected distortion gap between LALM and ALM quantizers. We intend to do a detailed analysis in a future work.

## 2.4 Convergence of ALM algorithm

In the current section, the convergence aspect of the ALM algorithm is dealt in detail. We discuss about the convergence of quantization levels of ALM and seek methods to quantify the rate of convergence. Such a convergence analysis will be useful to determine

the speedup performance of the ALM algorithm parameterized by a finite rate  $K$  and a variable  $i$  number of iterations. The convergence analysis of ALM hinges on the Perron-Frobenius theory, which is well-known in the context of Markov chains. As an added benefit, this method provides the convergence rate of the ALM algorithm. The section covers a proof for ALM convergence, starting with a few background concepts.

### 2.4.1 Background concepts for showing ALM convergence

For showing the convergence of the ALM algorithm, we first express the level update steps as successive linear transformations. Recall that the optimal (level update) solution at any iteration is realized as the root of (2.5) in the interval  $[s_{k-1}, s_{k+1}]$ . This valid root, for the iteration index  $i$ , can be expressed as a convex (linear) combination,

$$s_k^{(i+1)} = \theta_k^{(i)} s_{k-1}^{(i)} + (1 - \theta_k^{(i)}) s_{k+1}^{(i)}, \quad (2.11)$$

where  $\theta_k^{(i)} \in [0, 1]$ . The linear form of the above update equation will aid in the convergence analysis of the proposed algorithm. In vector notation the same can be expressed as,

$$\vec{s}^{(i+1)} = P_{\text{odd}}^{(i)} P_{\text{even}}^{(i)} \vec{s}^{(i)} \quad \text{where } i \in \mathbb{Z}_+. \quad (2.12)$$

Here,  $P_{\text{even}}^{(i)}$  and  $P_{\text{odd}}^{(i)}$  are square matrices having dimension  $K + 2$  (due to the inclusion of two reference levels). These matrices are constructed based on (2.11). For instance, if  $K = 3$ ,

$$P_{\text{odd}}^{(i)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \theta_1^{(i)} & 0 & 1 - \theta_1^{(i)} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \theta_3^{(i)} & 0 & 1 - \theta_3^{(i)} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \text{ and} \quad (2.13)$$

$$P_{\text{even}}^{(i)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & \theta_2^{(i)} & 0 & 1 - \theta_2^{(i)} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.14)$$

We notice that the two matrices,  $P_{\text{odd}}$  and  $P_{\text{even}}$  are row stochastic. Further, a (row) symmetry on the locations of zeros is observed. Also, the first and last rows of matrices

in (2.14) are set accordingly, so as to preserve reference levels,  $s_{\text{start}}$  and  $s_{\text{stop}}$ , in every iteration. Thus, the vector update described in (2.12)-(2.14) gains the requisite structure for applying the Perron-Frobenius theory [22, 23].

### 2.4.2 Insights on ALM convergence using an example

Consider a uniform (probability density) source in  $[0, 1]$ . On applying the LM and ALM algorithms, the same quantizers are attained, since the piecewise linear approximation can exactly track the uniform probability density. For illustration, consider  $K = 3$  and let the initial  $\vec{s}^{(0)}$  be equispaced levels. Then, by using the linear update operator  $\vec{s}^{(1)} = P_2 P_1 \vec{s}^{(0)}$ , we see that the update matrices constituted by the coefficients  $\theta_k^{(i)}$  are invariant across iterations; i.e.  $\theta_1^{(i)} = 2/3$ ,  $\theta_2^{(i)} = 1/2$  and  $\theta_3^{(i)} = 1/3$  for all  $i \in \mathbb{Z}_+$  (using (2.5)). Then, the matrix  $P_2 P_1$  has the structure,

$$P = P_2 P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{2}{3} & 0 & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{3} & 0 & \frac{2}{3} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.15)$$

Observe that  $P_1$  and  $P_2$  are row stochastic with non-negative entries. This enables us to use the Perron-Frobenius theory [22, 23] to find the fixed points of  $P$ . The fixed points will correspond to the optimal quantizer. Recall that, the quantization level at the  $n$ -th stage is given by the relation,  $\vec{q}^{(n)} = (P_2 P_1)^n \vec{q}^{(0)}$ . As  $n \rightarrow \infty$ ,  $(P_2 P_1)^n$  converges to a rank 2 matrix with two non-zero columns. In the next subsection, we show that these non-zero column vectors correspond to the fixed points of  $P$ . By imposing an ordering on the fixed points, a unique solution can be recovered. The following properties of the  $P$  matrix are of interest.

1.  $P$  is row stochastic.
2. Eigenvalues of  $P$  satisfy  $|\lambda| \leq 1$ .
3.  $\lambda = 1$  is an eigenvalue and  $\mathbf{1} = [1, 1, \dots, 1]^T$  is a corresponding eigenvector.
4. All eigenvectors of  $P$  are either symmetric or antisymmetric with respect to the flip operator (see Appendix A.1).

5. The geometric multiplicity of  $\lambda = 1$  is 2; i.e. there are 2 eigenvectors corresponding to the eigenvalue 1.
6. If  $\vec{v}_1 \neq \mathbf{1}$  is an eigenvector of  $\lambda = 1$ , then  $\vec{v}_2 = \mathbf{1} - \vec{v}_1$  is an independent eigenvector of  $\lambda = 1$ .

Details of the above results are described in [23]. Using these properties we can show the existence of a fixed point such that  $\vec{q}_{\text{opt}} = \lim_{n \rightarrow \infty} \vec{q}^{(n)}$  and  $P\vec{q}_{\text{opt}} = \vec{q}_{\text{opt}}$ . Besides finding the quantizer levels, the Perron-Frobenius formulation allows us to obtain the rate of convergence. The lead eigenvalue of the update operator matrix will determine the convergence rate towards the fixed point. This fact implies that the ALM quantizer has an exponential (decaying) rate. In this example, the convergence rate is  $O(1/3^n)$ , as the lead eigenvalue of  $P$  is  $1/3$ . Using these insights derived from the illustration, we will discuss the ALM convergence theorem.

### 2.4.3 ALM convergence theorem

In this section, we detail the steps leading to convergence of the ALM algorithm. Using the fact, the product of two row stochastic matrices is row stochastic, we see that  $P^{(i)} := P_{\text{even}}^{(i)} P_{\text{odd}}^{(i)}$  has every row adding up to unity. For the  $K = 3$  case, this product matrix has the following structure:

$$P^{(i)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \theta_1^{(i)} & 0 & \bar{\theta}_1^{(i)} & 0 & 0 \\ \theta_2^{(i)}\theta_1^{(i)} & 0 & \bar{\theta}_1^{(i)}\theta_2^{(i)} + \bar{\theta}_2^{(i)}\theta_3^{(i)} & 0 & \bar{\theta}_2^{(i)}\bar{\theta}_3^{(i)} \\ 0 & 0 & \theta_3^{(i)} & 0 & \bar{\theta}_3^{(i)} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad (2.16)$$

where  $\bar{\theta}_j^{(i)} = 1 - \theta_j^{(i)}$  for  $1 \leq j \leq K$ . Before proceeding for the formal proof, we list a few observations related to the matrix  $P^{(i)}$ . Firstly, notice the symmetry in the location of zeros across the rows. Also note that, the first and last rows of the matrix are independent of the scale parameters  $\{\theta_j^{(i)}\}_{j=1}^3$ , as these correspond to the reference levels. Further observe, the zero (column) vector appearing at alternate columns of the above matrix (which is due to linear updates acting on alternate entries of  $\vec{s}^{(i)}$ ). Finally note that, each entry of  $P^{(i)}$  lies within the  $[0, 1]$  range.

As illustrated earlier, the ALM convergence theorem leans on the Perron-Frobenius theory. This analysis is similar to the convergence results for gossip algorithms and consensus models [50, 51]. Nevertheless, our case differs in the aspect of number of fixed points, where we get two fixed points against one for the former schemes. A few notations will be useful in proving the main convergence result stated below. The symbol  $P^{(i)}$  represents the update matrix for the  $i^{\text{th}}$  iteration (having a structure similar to (2.16) for  $K = 3$ ). The column vectors of the finite product sequence will be denoted as,  $\prod_{i=1}^L P^{(i)} := [\vec{c}_{\text{start}}^{(L)}, \vec{c}_1^{(L)}, \vec{c}_2^{(L)}, \dots, \vec{c}_K^{(L)}, \vec{c}_{\text{stop}}^{(L)}]$ .

**Theorem 2.1** (*The ALM Convergence Theorem*). *Let  $\vec{s}^{(0)}$  be the initial quantization levels, and  $P^* := \lim_{L \rightarrow \infty} \prod_{i=1}^L P^{(i)}$  be the asymptotic product matrix. Then, the ALM algorithm (see Algorithm. 1) converges to the quantization levels,*

$$\vec{s}^* = P^* \vec{s}^{(0)}, \quad (2.17)$$

*independent of the initialized vector  $\vec{s}^{(0)}$ .*

For the complete description of the proof refer to [Appendix A.4](#).

The major points to take away from the ALM convergence result, are explained in the following remarks.

**Remark 2.2** (*Uniqueness of ALM quantizer levels*). *The ALM quantization levels is unique up to a sorted order. This is true since the eigenvectors of  $P^*$ , corresponding to the unit eigenvalue satisfy  $\vec{c}_{\text{start}} + \vec{c}_{\text{stop}} = \mathbf{1}$ .*

**Remark 2.3** (*Exponential rate of convergence of ALM*). *The ALM algorithm (see Table. 1) achieves exponential rate of convergence, that is, the  $\ell_2$  gap between the levels  $s_k^{(i)}$  and  $s_k^*$  drops at a rate  $O(\lambda_{(2)}^n)$ , where  $\lambda_{(2)}$  represents the maximum over the second largest eigenvalues of  $P^{(i)}$ .*

## 2.5 Asymptotic near-LM optimality of ALM

The objective of this section is to analyze the asymptotic near-LM optimality of the ALM quantizer, when the quantization rate (or equivalently the number of levels) is allowed to grow. Such an analysis will be useful in quantifying the tradeoff between the quantizer

level mismatch of ALM and LM, and the number of representation levels,  $K$ . Thereby, it benefits the quantizer designer by providing a choice to optimize between quantizer performance (i.e. MSE) and representation (or communication) cost, for any given finite number of iterations of ALM. Using a sequence of examples, we motivate the need for studying the asymptotic optimality. First, consider a ramp (source) PDF, i.e.,  $f_X(x) = x$  for  $x \in [0, 1]$ . The piecewise-linear approximation step of the ALM quantizer would exactly match the ramp function, since the source PDF is linear. More precisely, the ALM levels,  $s_k^{(i)}$  would match with the LM levels  $q_k^{(i)}$  for every iteration. Next, consider a triangular PDF,  $f_X(x) = 4x\mathbb{1}_{[0,0.5]}(x) + (4 - 4x)\mathbb{1}_{[0.5,1]}(x)$  with  $K = 2$  levels (the notation  $\mathbb{1}_{[\cdot]}(\cdot)$  is the 0 – 1 indicator function). In this case, the ALM algorithm approximates the PDF using three piecewise linear regions, making it imprecise. However, upon increasing the levels to  $K = 3$ , with sufficient iterations, ALM can perform at par with LM. This is attributed to the ability of ALM to capture the piecewise-linear parts of the triangular probability density, with increase in quantization granularity. Extending this argument to a generic continuous PDF, we observe that the approximation accuracy of ALM improves, with increasing quantization levels (or rate). In an asymptotic sense, that is as  $K \rightarrow \infty$  we expect the ALM quantizer to converge to the optimal LM quantizer. Building from the insights discussed here, we analyze the asymptotic near-LM optimality ALM in terms of its convergence rate for a finite  $K$  and a finite number of iterations  $i$ .

Let  $\bar{q}^*$  be the optimal LM quantizer and  $\bar{s}^*$  be the ALM quantizer. The asymptotic convergence property deals with the question, *how close are the levels of ALM quantizer with respect to the LM quantizer for a chosen number of quantization levels  $K$* . In our analysis, it is observed that the ALM quantizer converges to the levels of the optimal LM quantizer for higher quantization rates. While deriving distance bounds on the quantization error, we bank on the Taylor series expansion of the PDF. The linear approximation error at  $x = q_k$  can be bounded using the Taylor expansion about the interval  $x \in [q_k - \delta/2, q_k + \delta/2]$  and  $\delta > 0$ , and is given by,

$$\begin{aligned} f_X(x) &= f_X(q_k) + f'_X(q_k)(x - q_k) + \mathcal{O}((x - q_k)^2) \\ &= m_k x + c_k + \mathcal{O}((x - q_k)^2) \\ &= f_{\text{app}}(x) + \mathcal{O}((x - q_k)^2). \end{aligned} \tag{2.18}$$

For simplicity of notations, we restrict our attention to the level  $q_2$ . The two neighboring levels of interest are  $q_1$  and  $q_3$ . Let  $q_2^{(i)}$  and  $s_2^{(i)}$  denote the optimal levels at iteration index  $i$ , for LM and ALM quantizers respectively (see (2.5) for optimality condition). Then, the Taylor expansion at  $x = q_2^{(i)}$  is,

$$f_X(q_2^{(i)}) = f_{\text{app}}(q_2^{(i)}) + \mathcal{O}(\varepsilon_K), \quad (2.19)$$

where  $\varepsilon_K < 1$  denotes the maximum squared error in any quantization bin (or interval)  $[q_{k-1}, q_{k+1}]$ . It can be mathematically defined as,

$$\varepsilon_K := \max_{i \in \mathbb{Z}_+} \left[ \max_{1 \leq k \leq K-1} |q_{k+1}^{(i)} - q_{k-1}^{(i)}|^2 \right] \quad (2.20)$$

This can be computed by evaluating the pairwise square distance between neighboring levels in the case of LM algorithm. Since  $K$  signifies the number of quantization levels, we expect the neighboring levels to get closer, and thus  $\varepsilon_K$  goes to zero as  $K \rightarrow \infty$ . Using this fact,  $|q_2^{(i)} - x|^2 \leq |q_3^{(i)} - q_1^{(i)}|^2 \leq \varepsilon_K$  for all  $x \in [q_1^{(i)}, q_3^{(i)}]$ . For a concrete example, consider a uniformly distributed source, which has  $K$  equispaced levels. We discover that  $\varepsilon_K = \frac{4}{K^2}$ , since LM algorithm converges to equally spaced intervals. It is noted that the term  $\mathcal{O}(\varepsilon_K)$  is primarily dependent on the curvature (i.e. the second derivative) of the PDF  $f_X(x)$ .

The asymptotic optimality of ALM algorithm is examined further in the forthcoming result. Recall that,  $q_2^{(i)}$  and  $s_2^{(i)}$  represent the second quantization level corresponding to the  $i$ -th iteration of LM and ALM respectively. The asymptotic levels,  $q_2^* := \lim_{i \rightarrow \infty} q_2^{(i)}$  and  $s_2^* := \lim_{i \rightarrow \infty} s_2^{(i)}$ .

**Theorem 2.4** (*Near-LM optimality of ALM*). *There exists number of quantization intervals,  $K \geq K_0$  such that  $|q_2^* - s_2^*| \leq \varepsilon$ , where  $K_0$  is a positive integer and  $\varepsilon$  is an arbitrary positive real number. In particular, the convergence rate of ALM quantizer level,  $s_2^*$ , to the LM quantizer level,  $q_2^*$ , is characterized by the upperbound,*

$$|q_2^* - s_2^*| \leq C \times \mathcal{O}(\varepsilon_K^{1.5}), \quad \text{where } C = \frac{1}{f_{\text{app}}(s_2^*)}.$$

See [Appendix A.5](#) for the detailed proof. The key aspects of the result are summarized below.

**Remark 2.5.** *Combining the results of Theorem. 2.1 and 2.4, we see that at any finite iteration  $i$ , the difference  $|q_2^* - s_2^{(i)}| = C_i \mathcal{O}(\varepsilon_K^{1.5}) + \mathcal{O}(\lambda_{(2)}^i)$ , where  $C_i = \frac{1}{f_{\text{app}}(s_2^{(i-1)})}$  and  $\lambda_{(2)}$  is the Perron eigenvalue. This result follows by the triangle inequality.*

**Remark 2.6.** *This result also holds for all quantization levels  $q_k^*$  and  $s_k^{(i)}$  where  $1 \leq k \leq K$ , with the assumption the  $f_X(x) > 0$  for all  $x \in [0, 1]$  satisfying the smoothness condition in (2.1).*

## 2.6 Experimental results and discussion

To showcase the analytical results obtained in the previous section we performed extensive simulations and experiments on various synthetic datasets. These results are discussed separately for ALM and LALM. Broadly our results characterize the designed quantizers with respect to three aspects; viz. *error-bitrate tradeoff*, *convergence* and *accuracy*.

### 2.6.1 Experimental results for ALM

Hardware and Software Specifications: For the ALM based simulations we have used Python 3.6 along with the NumPy (ver 1.16.4) and SciPy (ver 1.1.0) packages. A general purpose Windows 10 PC with hardware specification - Processor: Intel(R) Core i5 CPU @ 2.2 GHz, RAM: 8 GB was used.

Error-Bitrate Tradeoffs: Our simulations are performed on signal sources with the Beta distribution. This choice was made due to two main advantages- viz. unimodality and bounded support [30, 31]. In addition, the Beta distribution class enables us to compare the existing results for LM quantizer with the proposed ALM quantizer. In Fig. 2.1, we have characterized the error-bitrate tradeoff of ALM quantizer for different source distributions. Observe the convergence of the ALM quantizer to the limit of the LM quantizer with increasing quantization levels. The plots in Fig. 2.1(b) show the dependence of the quantizer MSE on the variance of the source. The variance of the symmetric Beta distribution monotonically decreases with the parameters  $\alpha$  and  $\beta$  (where  $\alpha = \beta$ ). A similar dependence is shown in Fig. 2.1(c) for truncated normal and exponential sources, which are commonly used sensor data models.

Convergence and Accuracy: The convergence aspect of the ALM algorithm is depicted using the evolution of the quantization levels (with iteration) for a Beta(4,2) source (see Fig. 2.2(a)). In Fig. 2.2(c) the relative accuracy of ALM with respect to LM quantizer is shown for a Beta(4,2) source. Being an asymmetric distribution, the ALM levels close the left deviates from the LM, perhaps due to the negative skewness of the distribution.

This deviation results in an MSE difference of the order of  $10^{-4}$ , which is equivalent to 13% relative error.

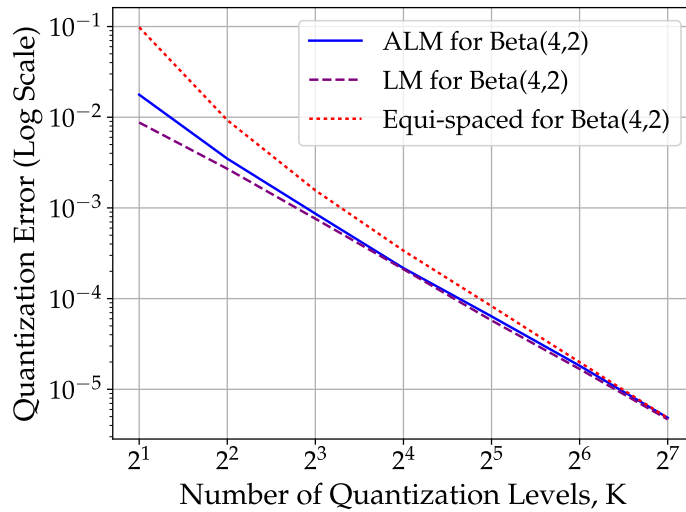
*Measuring the Speedup of ALM:* In order to understand the computational complexity aspect of the ALM algorithm, we use the algorithm runtime in Python (*Version-3.6* using the `time` library and the `clock()` function) . The stopping rule for runtime computation is chosen as the iteration until which the computed MSE is within 1% of the optimal MSE. Experiments for the Beta(4,2) source probability density shows a computation time speedup of **3.4X** for the ALM over LM at  $K = 8$  (i.e. bitrate of 3 *bits per level*; see Fig. 2.2(b)). A similar experiment on a truncated exponential source (with mean 0.5), provides an average performance improvement of **3.3X** over the bitrate choices  $\{1, 2, \dots, 7\}$  per quantizer level. At low bitrates, speedups are seen to be less than unity, which is probably due to the larger number of iterations required for ALM to settle within the 1% error criteria.

## 2.6.2 Experimental results for Learning ALM

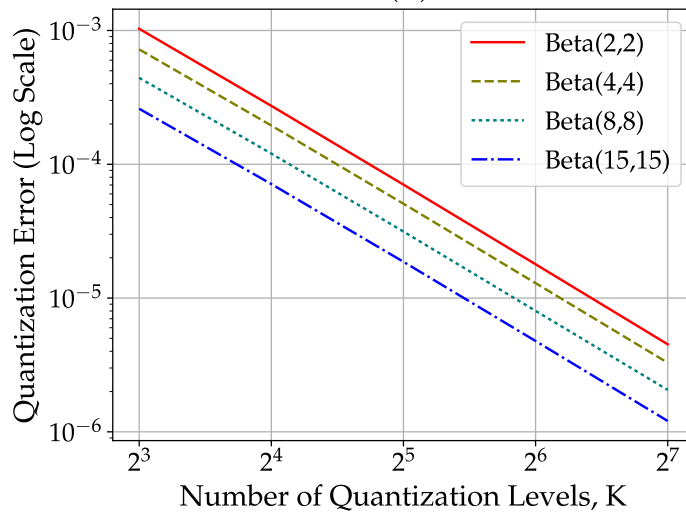
Learning ALM (LALM), which is a data-driven version of the ALM quantizer, is analyzed in terms of the sample complexity, speedup and energy (or computational) efficiency. The following experiments for LALM brings out its advantage over the known quantizers such as LVQ and  $k$ -means, especially when deployed in resource constrained edge devices. For this section, we have generated synthetic datasets from unimodal Beta distribution, which is representative of source distributions that appear in several IoT applications [52]

*Hardware and Software Specifications:* LALM based experiment were conducted in two segments. First, the energy and memory usage characterization was performed in an Android SDK emulator environment in a Linux machine (Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz, RAM: 16 GB) running Python 3.6 and using the *energyusage* and *memory-profile* packages. Next we measured the algorithm runtime in an Android device (Android 9 API 28, Qualcomm Snapdragon 632 Octa-core Max 1.80 GHz processor) running PyDroid.

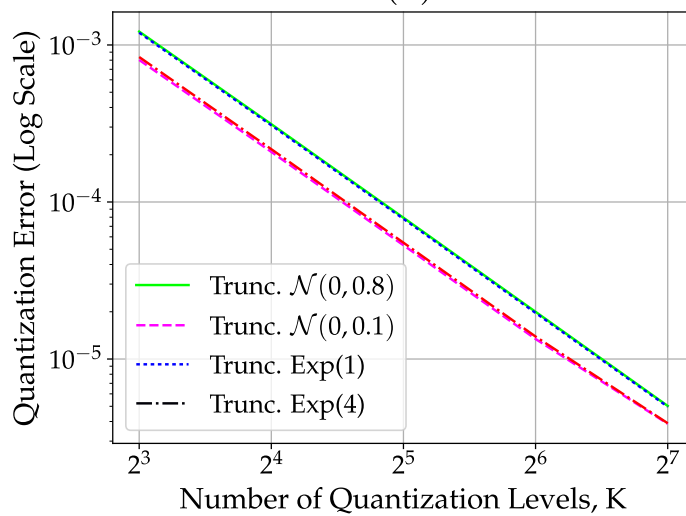
*Energy and Memory Profiling of LALM:* We have used Android SDK to invoke the LALM algorithm written as Python code in the Android Studio emulator. The *energyusage* package in Python allows for power measurement via the RAPL (Running Aver-



(a)

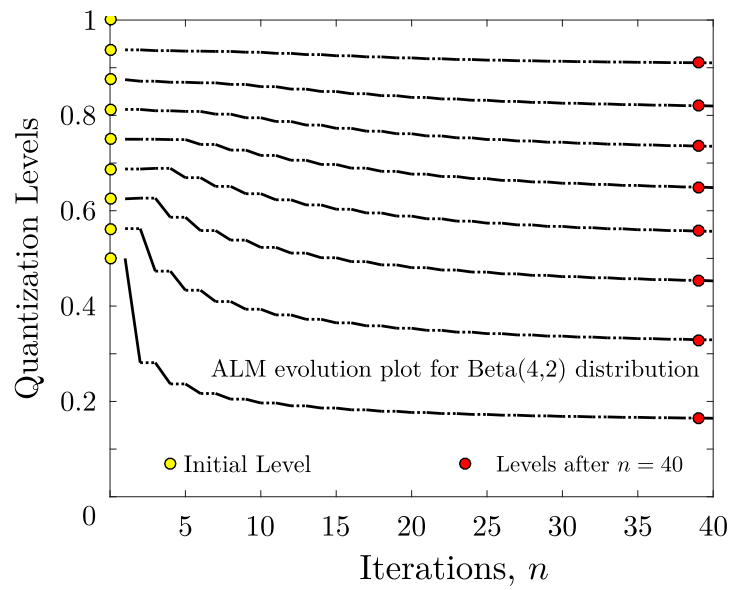


(b)

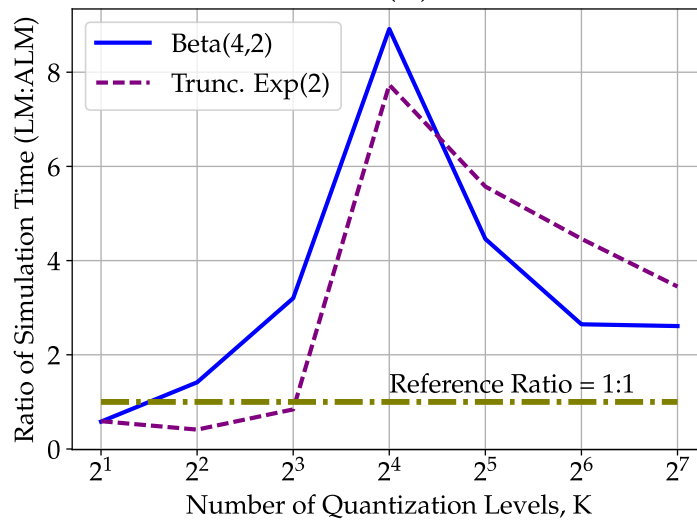


(c)

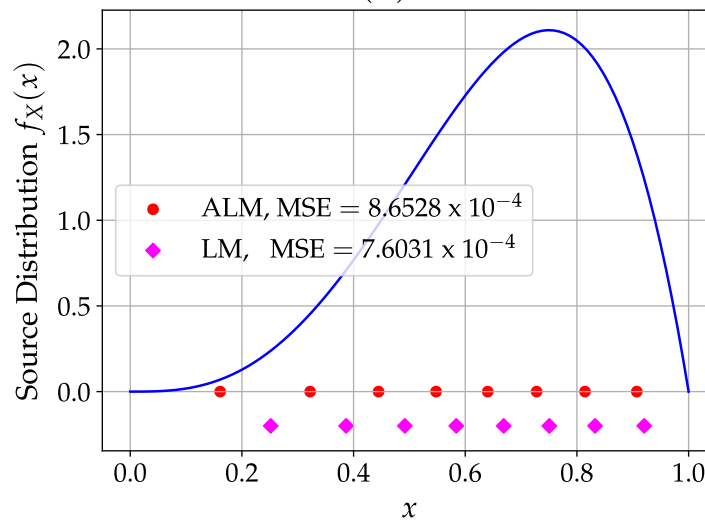
Figure 2.1: (a) Error-bitrate tradeoff for LM and ALM schemes for Beta(4,2) source distribution. (b) Error performance of ALM for symmetric Beta distribution having different variance. (c) ALM quantization error vs bitrate for truncated normal and truncated exponential distributions.



(a)



(b)



(c)

Figure 2.2: (a) Quantizer evolution of ALM algorithm for Beta(4,2) source. (b) Ratio of ALM to LM simulation runtime for Beta(4,2) and truncated exponential source distributions (c) Relative quantization levels of ALM and LM for Beta(4,2) distribution.

age Power Limit) interfaces available on Intel processors [53]. Associated registers of these interfaces provides power-related information about the CPU. In particular the Energy Status register (`MSR_PKG_ENERGY_STATUS`) allows for power measurement for the processes running in the CPU. For computing the average wattage for the Python process, the *energyusage* package subtracts the average baseline wattage obtained 10 seconds prior to the code execution. The energy in KWhr is obtained by multiplying the resultant wattage with the code execution time.

The memory usage of LALM is obtained using the *memory\_profiler* package. This package has the functionality to estimate the time series of the memory utilized by the python code. For this standardized Python profiler the memory sampling time is 0.1 seconds. In addition, the package helps us to track the memory of individual child processes (or function calls), which is essential to offset the estimation error due to the background code. Apart from the memory usage statistics, the profiler also reports the profiling duration which is proportional to the execution time of the algorithm.

In Table 2.2, a comparison of the LALM quantizer design algorithm with equi-spaced quantizer (entropy coded online using Huffman coding), LVQ and  $k$ -means implementation is shown. It is observed that for a data size of  $1.25 \times 10^5$  and a bitrate of 4 *bits per level*, LALM gives 10x energy reduction over entropy coded quantizer and  $k$ -means. The LVQ algorithm, without any parameter tuning (i.e. for a fixed value of learning rate  $\gamma$ ), has energy utilization comparable to LALM. However, in practice LVQ requires repeated iterations for obtaining near-optimum performance, thus taking its energy utilization beyond  $k$ -means. In terms of memory used, LALM requires lesser (storage) space than  $k$ -means, because of its online implementation. This is facilitated by transforming the data into piecewise linear representation in each time window of the LALM algorithm execution. Speedup of LALM over  $k$ -means and LVQ is indicated through the profiling duration of each algorithm. To note, this speedup is obtained without much deviation from the optimal MSE.

*Remark:* Though the entropy coded equi-spaced quantizer is known to be optimal for high data rates, the need to code over large blocklengths makes it unfit for real-time applications. This was verified using the Lempel-Ziv-Welch coding scheme, which

is a universal data compression algorithm [54]. For implementation we fed the discrete symbols corresponding to the quantization bins of the equi-spaced quantizer. During training, the algorithm learns a dictionary, that helps in the encoding of the symbols. It is observed that for block lengths less than 20000 data samples, there is no coding advantage (or gain), owing to the overheads of the learned dictionary.

Table 2.2: A comparison of different data driven quantizer design algorithms in emulator

Algorithm	Compute Energy (in KWhr)	Average Memory Utilized (in MiB)	Memory Profiling Duration (in sec)	Mean Squared Error (MSE)
Equi-spaced (Huffman coded)	$4.20 \times 10^{-4}$	100.36	55.9	$2.62 \times 10^{-4}$
LVQ (with $\gamma = \frac{1}{\lceil n/100 \rceil}$ )	$1.84 \times 10^{-5}$	62.61	12.8	$4.24 \times 10^{-4}$
k-Means	$1.59 \times 10^{-4}$	86.43	43.3	$2.06 \times 10^{-4}$
LALM (1-shot)	$1.32 \times 10^{-5}$	69.48	10.8	$2.14 \times 10^{-4}$

Table 2.3: A comparison of different data driven quantizer design algorithms in Android device for the Beta(4,2) signal source probability distribution

Data Size	Algorithm Runtime (in sec)			
	Equi-spaced (Huffman coded)	LVQ (parameter tuned)	k-Means	LALM (1-shot)
1000	1.76	0.35	4.21	1.31
5000	8.56	2.47	8.27	1.38
25000	42.60	12.23	27.73	2.21
125000	311.91	61.49	117.76	3.57

Runtime performance of LALM in an Android device: In Table. 2.3, we get a first hand measure of the speedup of LALM through the algorithm execution time measurement

(using the *timeit* library in Python). On comparing with the SciPy's in-built  $k$ -means algorithm, we see that LALM has a significant reduction, while ensuring near-optimality. For the LVQ algorithm the results are tabulated by considering 10 iterations of parameter tuning. This involved cross validation with different values of the learning parameter  $\gamma$ , which was varied in the form  $\frac{1}{|n/S|}$  where  $S \in \{10, 20, \dots, 100\}$ . At lower data sizes, although LVQ executes faster, the MSE of the quantizer is more than LALM. For the entropy coded equi-spaced quantizer, we see the larger algorithm runtimes, which is mainly attributed to the long blocklength requirement while doing entropy coding. It is also noted that for low data sizes the coding gains are not substantial. Table 2.3 summarizes the speedup advantage of LALM in edge devices.

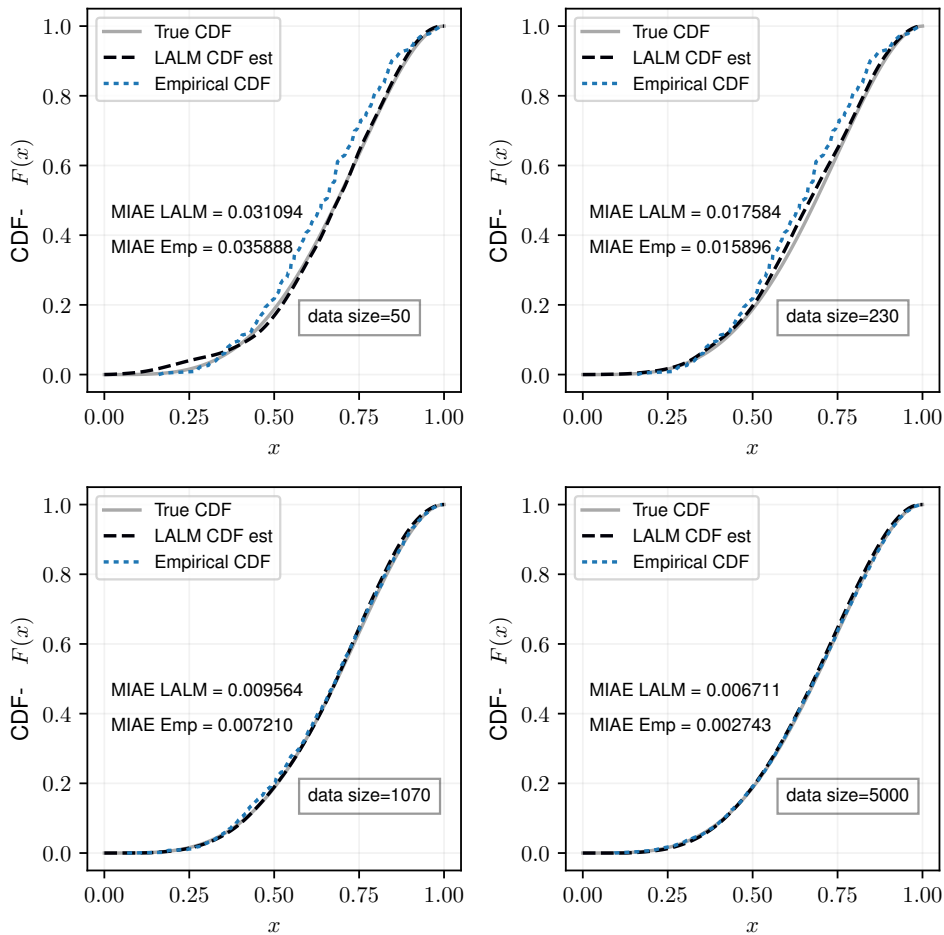


Figure 2.3: A comparison of the CDF estimate obtained using LALM algorithm and the Glivenko Cantelli empirical CDF estimate. The mean integrated absolute error for four different data sizes are shown.

*CDF estimation with LALM:* Apart from quantizer design, the LALM algorithm estimates the signal source PDF. The integral of the PDF, that is, the source distribution or Cumulative Distribution Function (CDF) is a vital quantity for deriving signal statistics and analytics. In Fig. 2.3, we benchmark the performance of the LALM CDF estimate with the fundamental Glivenko-Cantelli empirical CDF estimate [55]. The plots indicate that piecewise linear approximation of PDF obtained by the LALM algorithm (for bitrate 4 bits per level), provides a CDF estimate which is nearly order optimal to the Glivenko-Cantelli estimate. For comparison, we have used the Mean Integrated Absolute Error (MIAE) as the cost metric, i.e.  $\text{MIAE} = \int_0^1 |F(x) - F_n(x)| dx$ , where  $F(x)$  and  $F_n(x)$  represent the true and the empirical CDF. The Glivenko-Cantelli estimate has the best known MIAE decay rate of  $\mathcal{O}\left(\sqrt{\frac{1}{n}}\right)$ , as given by the DKW inequality [56].

## 2.7 Summary

In this chapter, we considered quantizer design algorithms for energy constrained and hardware-limited systems. A model-driven ALM quantizer was developed, based on the piecewise linear probability density approximation. The ALM quantizer demonstrated computational and speedup advantages over classical Lloyd-Max quantizer, while being nearly-optimal. In the data-driven setting, we introduced a Learning ALM (LALM) algorithm that is consistent with the ALM quantizer, and additionally estimates the signal source probability density. The LALM quantizer has faster convergence, less energy usage and efficient memory utilization when compared to the  $k$ -means quantizer. Through analysis, the proposed quantizers were shown to have exponential convergence rate and quantizer levels are shown to converge to a unique fixed point near the optimal Lloyd-Max quantizer. Using extensive simulations and experiments on an Android-based edge device, we validated the efficacy of ALM and LALM quantizer designs. In future, we wish to extend the proposed algorithms to learn a global quantizer using federated aggregation methods, and analyze the sample complexity of data-driven LALM quantizer.



# Chapter 3

## Data-Driven Approaches for Overpredictive Quantizer Design

The objective of this chapter is to motivate application-specific constraints while performing quantizer design for edge devices. The focus will mainly be on overprediction constraints that arise in several applications, for example, the representation of TV whitespace protection regions. First, we develop a stochastic approximation based online quantizer design based on a two time-scale stochastic gradient design. Later in the chapter, we extend the piecewise linear approximation ideas developed for the ALM quantizer design (Chapter 2). We show the proposed schemes' convergence and demonstrate the mean-squared error performance on synthetic and real-world data.

### 3.1 Background

Signal retrieval and storage are viewed as fundamental tasks for several emerging applications such as TV white spaces and smart grids [57, 58, 59, 60]. Recent implementation and design efforts emphasize the need for efficient signal representations to address the resource constraints arising in such applications [61, 62, 63, 64]. It is believed that signal quantization will play a vital role in enabling these systems' deployment.

This chapter focuses on data-driven scalar quantizer design for specific applications where the quantized signal should always overpredict the actual signal. There are several applications where such overprediction of signals is required. For instance, consider a

cognitive radio system that uses the TV white spaces spectrum [57, 65]. The secondary (unlicensed) user devices in this setup are required to consult a TV white space database to retrieve the protection regions of TV transmitters. An example of such protection regions is shown in Fig. 3.1, where these contours are obtained from the iconectiv website [3] for Channel 22 in the New York region. Often these protection contours are closed regions of irregular shapes, thus requiring complicated signal representations for storage. Therefore, these protection regions are stored in the database as circular approximations, parameterized by the protection radius [57, 66]. In particular, a circular approximation to the protection region number 2 is illustrated in Fig. 3.1. For the approximation purpose, a quantizer is needed to convert the protection radius to finite precision (bit) representation. While the quantizer is designed, the represented radius should always overpredict the actual radius. This criterion is to ensure that the primary (licensed) user spectrum is always protected. Because the represented protection regions in the database will form an ‘envelope’ over the actual protection regions, we will term such designed quantizers as *envelope quantizers*.

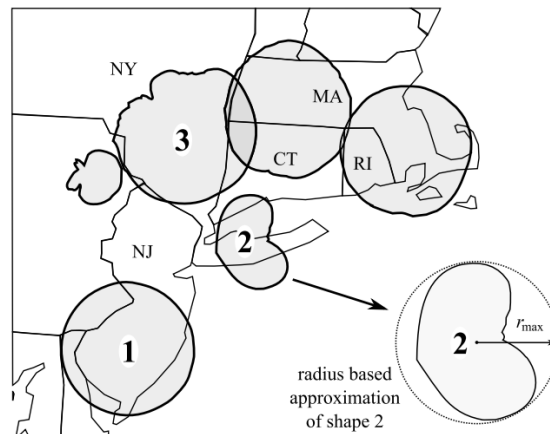


Figure 3.1: The graph represents the contours of the protection regions for Channel 22 as obtained from the website of a certified TV white space service provider iconectiv in the United States [3]. Figure is taken from [66]

Although envelope quantizer design by exhaustive search is possible, it may be expensive to implement, especially when the dataset involved is large. Taking this into account, we propose an online data-driven quantizer design using stochastic approximation methods [67, 68]. The proposed quantizer is learned by following stochastic gradient

updates while ensuring that the overprediction constraint is always met. For brevity, we will term the proposed quantizer as *Stochastic Approximation based Envelope quantizer* and abbreviate it as SANE. The stochastic approximation approach uses a two-time scale update, which concurrently estimates the source’s probability density and modifies the quantizer levels. The iterative nature of SANE provides an online update algorithm for the envelope quantizer design with excellent convergence properties. In the context of the TV whitespace example, SANE quantizer provides a better utilization of whitespace resources and increases the capacity of the secondary (unlicensed) users.

An alternate method to design envelope quantizers is using piecewise linear approximations of the probability density function, as discussed in the previous chapter. This method uses the envelope constraint while optimizing the quantization error. We term this quantizer design, which assumes a known signal probability density function, as *Approximate Envelope Quantizer* or AEQ. Each level update in AEQ can be done by solving a cubic polynomial whose coefficients depend on piecewise linear approximation parameters. Additionally, we extend AEQ to a data-driven setup, where the quantizer levels are learned from the signal samples alone. The Learning AEQ (LAEQ) is a one-shot approach, which requires a chunk of data samples to be available for each iteration, unlike the online updates available in SANE. However, LAEQ is devoid of hyperparameters and is suitable for quantizer designs involving a moderate number of data samples.

### 3.1.1 Related works

Classical literature has explored several extensions and improved designs [24, 25, 32, 28, 15] of Lloyd-Max quantizer algorithm. The data-centric approach towards MSE quantization is attributed to the works of Pollard related to the method of  $k$ -means [33, 31, 37]. This scheme is analogous to the famed expectation-maximization (EM) algorithm in adaptive signal processing [69]. With the advent of neural networks in the early ’90s, Kohonen introduced an online quantization scheme termed learning vector quantization (LVQ) [34]. The LVQ takes a ‘winner-take-all’ approach, assigning the nearest-neighbor quantization level as the ‘winner’ corresponding to an arrived data observation. In neural network literature, this method has applications in supervised classification and self-organizing maps (SOM). In principle, the algorithm provides a specific weight update to the ‘winner’ for

the underlying neural network using a gradient-based scheme. Almost sure convergence and the self-organizing property of the LVQ have been extensively studied [70]. Stochastic approximation methods are widely applicable in machine learning, autonomous control, adaptive signal processing, and others [68, 71]. Although the field has a long history [67], its relevance is felt in many modern applications. An extensive study on stochastic approximation and its variants used in machine learning has been reviewed by Netrapalli [72], where finite sample and finite time bounds are discussed.

Recent interest in TV white space applications [57, 65, 58, 61, 62] has produced algorithms for overpredictive quantizer design using improved implementations of exhaustive search. In the work by Maheshwari and Kumar [66], the authors study the feasibility of a broadcast-based geolocation database transmission over rate constrained satellite links. For this application, the quantization process of primary services protection regions has to be such that any point in the assigned protection region must not be declared as a white space region. An 'even-odd' algorithm has been suggested where the even (odd) indexed quantization levels are fixed, and the odd (even) indexed quantization levels are exhaustively searched. The iterative procedure performed here provides a template for the design of overpredictive quantizers in this chapter. A key difference of our work from the even-odd algorithm suggested in [66] is that we rely on the signal source probability density and do not perform an exhaustive search. Thereby we reduce the computational complexity involved and help in scaling the algorithm for larger datasets.

In this chapter, we aim to develop a stochastic approximation approach, akin to LVQ, to design the envelope quantizers motivated in the introduction. A summary of the key contributions is listed below.

- We propose the SANE quantizer design for two quantizer cost functions - namely, the mean absolute error and the mean squared error.
- We show the convergence of the proposed stochastic approximation scheme using the ODE approach [68].
- We evaluate the performance of the SANE quantizer using a synthetic dataset and an already available TV white spaces dataset.

- We propose a piecewise linear approximation-based envelope quantizer and discuss its data-driven extension. A comparative study is performed between SANE and LAEQ on a synthetically generated dataset.

## 3.2 System model and problem formulation

### 3.2.1 Signal and quantization models

Let  $X_1, X_2, \dots, X_N$  represent  $N$  IID scalar signal samples drawn from an unknown probability density function  $f(x)$ . We will assume that  $f(x)$  has a bounded support in the interval  $[0, 1]$  and this function is continuous and differentiable in this domain.

For the quantizer model, we consider a fixed rate scalar quantizer with  $K$  quantization levels. The quantizer is denoted by the map  $Q : [0, 1] \rightarrow \{q_1, q_2, \dots, q_K\}$ , where an ordering  $q_1 < q_2 < \dots < q_K$  is assumed. Distinct from classical quantizers, in this work, we discuss the design of data-driven quantizers that overpredicts the signal samples. That is,  $Q(x) \geq x$  for all  $x \in [0, 1]$ . Since the signal source probability distribution is unknown, these overpredictive quantizers are designed based on the signal observations  $\{X_n : n \in [1 : N]\}$  alone. Such an overpredictive quantizer that minimizes a certain distortion criterion is of interest. As stated earlier, we will term these designed quantizers as envelope quantizers.

### 3.2.2 Distortion measures and problem formulation

In the design of envelope quantizers, we consider two distortion measures, viz. mean absolute error (MAE) and mean squared error (MSE). We denote these distortion measures as  $\mathcal{D}_{\text{MAE}} := \mathbb{E}[|X - Q(X)|]$  and  $\mathcal{D}_{\text{MSE}} := \mathbb{E}[|X - Q(X)|^2]$  respectively. The goal of this work is to develop envelope quantizers that minimize these distortions while satisfying the overprediction constraint. More formally this can be stated as,

$$\begin{aligned} & \arg \min_Q \mathcal{D}_Z \quad \text{where } Z \in \{\text{MAE}, \text{MSE}\} \\ & \text{subject to } Q(X) \geq X. \end{aligned} \tag{3.1}$$

We note that the expectation operation in these distortion measures is taken with respect to the unknown probability density  $f(x)$ . To address this concern in the proposed algorithms, we will use approximation methods to estimate the underlying probability

density. Further, it is noted that the objective function  $\mathcal{D}_{\text{MAE}}$  is not differentiable when the argument is zero, but is almost surely differentiable because a density function exists.

### 3.3 Methods motivated by scalar equivalent of LVQ

In this section, we describe a greedy envelope quantization approach motivated by the LVQ scheme [34]. Consider a dataset of  $N$  observations -  $X_1, X_2, X_3, \dots, X_N$ . Without loss of generality, we ascertain that the quantization points,  $\{q_1, q_2, \dots, q_K\}$  lie on top of data points (see Sec. 2; [66]). We explain the one-step greedy approach below.

#### 3.3.1 Greedy updates for envelope quantizer

Let  $q_1 < q_2 < \dots < q_K := 1$  be the quantization levels, initialized randomly in the interval  $[0, 1]$  (except  $q_K$  which is fixed to the extreme point  $x = 1$ ). For ease of exposition, consider the update of the quantization level  $q_2$ . We denote  $x$  as the level  $q_2$ ,  $y$  as the nearest left neighbor of  $q_2$  and  $z$  as the nearest right neighbor of  $q_2$ . Fig. 3.2 illustrates the notation above.

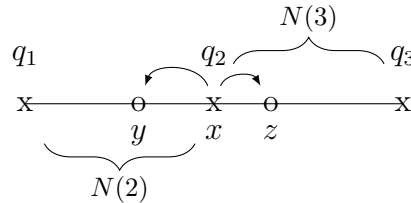


Figure 3.2: The one-step greedy procedure for envelope quantization. The quantization level  $q_2$  modifies to data location  $y$  or  $z$  or remains at  $x$  depending on the decrease in the absolute error cost function. The quantities  $N(2)$  and  $N(3)$  indicates the number of data points in left and right bin intervals  $(q_1, q_2]$  and  $(q_2, q_3]$  respectively.

We use the notation  $N(i)$ ;  $1 \leq i \leq K$  to represent the number of data observations in bins defined by the interval  $(q_{i-1}, q_i)$  (Note : For  $i = 1$ , we define  $q_0 := 0$ ). Specifically for  $i = 2$ , the adjacent bins of level  $q_2$  will have  $N(2)$  and  $N(3)$  data points respectively. The cost function decrease due to a left jump (i.e from  $x$  to  $y$ ) is given by,

$$c_2 = N(2)(x - y) - (q_3 - x). \quad (3.2)$$

Similarly, a jump towards right (i.e from  $x$  to  $z$ ) would result in a decrease,

$$d_2 = (q_3 - z) - (N(2) + 1)(z - x). \quad (3.3)$$

The one-step greedy approach will perform an operation based on one of the following cases:

**case 1** if  $c_2 \leq 0$  and  $d_i \leq 0$ ,

do nothing, remain at  $x$

**case 2** if  $c_2 > 0$  and  $d_i \leq 0$ ,

$q_2 \rightarrow y$

**case 3** if  $c_2 \leq 0$  and  $d_i > 0$ ,

$q_2 \rightarrow z$

**case 4** if  $c_2 > 0$  and  $d_i > 0$ ,

$q_2 \rightarrow y$  if  $c_2 > d_2$ ; else  $q_i \rightarrow z$

### 3.3.2 Failure of the greedy approach

The greedy algorithm performs the same even with MSE cost since it considers only the distance between a pair of points. The one-step greedy algorithm can be modified to update multiple quantization levels simultaneously. Since the adjacent levels,  $q_1$  and  $q_3$  are not affected due to  $q_2$  update; we can perform a joint update for all the even and odd indexed levels. Another extension is to consider multiple updates in a single iteration. However, this multi-step scheme is not scalable to large datasets or when the data streams in an online manner. In many instances, there are chances that the greedy algorithm falls in a local minimum. One possible alternative to counter this is to adapt the greedy scheme by randomizing points chosen as left and right neighbors. However, the online implementation of such schemes is not feasible. Hence we look into online recursive methods, which are based on stochastic approximation.

### 3.4 Stochastic approximation based design for envelope quantizers

To design the data driven envelope quantizer, we consider the minimization of the mean absolute error (MAE) and the mean-squared error (MSE) distortions. The error minimization will be based on stochastic approximation method, where the cost function is minimized using stochastic gradient descent.

#### 3.4.1 Minimization of mean absolute error

We consider the mean absolute error when the probability density function of the signal source is known. That is,

$$\mathcal{D}_{\text{MAE}} = \sum_{k=1}^K \int_{q_{k-1}}^{q_k} |q_k - x| f(x) dx. \quad (3.4)$$

The gradient (partial derivative) of the cost with respect to each quantization level,  $q_k$  for  $k = \{1, 2, \dots, K\}$  is given by,

$$\begin{aligned} \frac{\partial \mathcal{D}_{\text{MAE}}}{\partial q_k} &= \int_{q_{k-1}}^{q_k} f(x) dx - (q_{k+1} - q_k) f(q_k), \\ &= \mathbb{P}\{X \in (q_{k-1}, q_k]\} - (q_{k+1} - q_k) f(q_k) \end{aligned} \quad (3.5)$$

A possible method to minimize the MAE cost is to set the above gradient to zero. However, it is not possible to solve for  $q_k$  without the knowledge of the probability density of the signal source. Moreover a closed form solution is infeasible in this case. Hence, we propose a stochastic gradient descent based quantizer update, utilizing the stochastic approximation scheme. The quantizer level updates can be represented as,

$$q_k(n+1) = q_k(n) - a(n) \frac{\partial \mathcal{D}_{\text{MAE}}}{\partial q_k} \quad (3.6)$$

where the learning rate  $a(n)$  is chosen such that  $\sum_n a(n) = \infty$  and  $\sum_n a(n)^2 < \infty$  [67]. Viewed as a noisy discretization of the limiting o.d.e., these criteria ensure that the learning parameter covers the entire time axis, as well as makes the effect of noise in the observations asymptotically negligible [68].

There are two unknown terms in the gradient expression in the quantization update in (3.6). The first is the probability that the data point falls in the quantization bin

$(q_{k-1}, q_k]$ , that is  $\mathbb{P}(q_{k-1} < X \leq q_k)$ . The second is the probability density function  $f(x)$  at  $x = q_k$ . We estimate these unknown terms by using a two time-scale stochastic approximation scheme and a wavelet density estimation algorithm respectively.

**Two time-scale stochastic approximation :** The proposed two scale stochastic approximation update, with learning rates  $a(n)$  and  $b(n)$  can be expressed as,

$$q_k(n+1) = q_k(n) - a(n) \left[ z_k(n) - \Delta q_k(n) \widehat{f}(q_k(n)) \right], \quad (3.7)$$

$$z_k(n+1) = (1 - b(n))z_k(n) + b(n) \mathbb{1}_{X(n)} \{(q_{k-1}(n), q_k(n+1)]\}, \quad (3.8)$$

where  $\Delta q_k(n) := q_{k+1}(n) - q_k(n)$ , and  $\widehat{f}(x)$  is the estimate of the probability density. In (3.8)  $z_k(n+1)$  represents the estimate of the bin probability  $\mathbb{P}\{X \in (q_{k-1}, q_k]\}$ , and the notation  $\mathbb{1}_{\{Y\}}\{I\}$  denotes the 0-1 indicator function for the event  $\{Y \in I\}$ . This two time-scale update process is akin to a nested loop. That is, the outer loop with a slower learning rate appears to be quasi-stationary to the inner loop having a faster learning rate. For this design, the learning rate  $b(n)$  is chosen such that  $\sum_n b(n) = \infty$ ,  $\sum_n b(n)^2 < \infty$  and  $a(n) = o(b(n))$ .

**Wavelet Density Estimation:** The remaining unknown term in the update rules presented in (3.7) and (3.8) is the density estimate  $\widehat{f}(x)$ . In order to estimate the density we use a sliding window based wavelet density estimation scheme. In addition to reducing the sample complexity, this scheme also better approximates the unknown density of the stationary source in comparison to the histogram and kernel based methods [73, 74]. The estimation scheme also allows to fix the resolution of the wavelet approximation in a data driven manner. For this, we use an adaptive technique that compares the energy of signal samples in time and wavelet domains (based on principles in [75]). We provide details of this method in the supplementary material (Appendix B.1).

The algorithm for the MAE based envelope quantizer design is summarized as follows.

1. Signal,  $X(n)$  for  $n \geq 0$  and  $n \in \mathbb{Z}$ , is taken as input.
2. Envelope quantization levels,  $\vec{q} := [q_1, q_2, \dots, q_K]$  are initialized to be equi-spaced in the interval  $[0, 1]$ .

3. For each sample  $X(n)$ , the quantization levels are updated using the two time scale stochastic approximation steps – (3.7) and (3.8) and learning rates  $a(n)$  and  $b(n)$  are chosen appropriately.
4. The unknown density in the update rule is estimated using a sliding window based wavelet density estimation, with an adaptively learned resolution (see Appendix B.1).
5. Steps 3 and 4 are repeated until the stopping criteria, (either maximum iteration or an error criterion), is met.

### 3.4.2 Minimization of mean squared error

Drawing insights from the stochastic approximation scheme for mean absolute error, we extend it for the case of mean squared error (MSE) distortion. The MSE distortion,  $\mathcal{D}_{\text{MSE}} = \mathbb{E}[|X - Q(X)|^2]$ , is minimized with respect to the  $Q(\cdot)$ , while ensuring the envelope constraint,  $Q(X) \geq X$ . To this end, the gradient (partial derivative) of the MSE distortion with respect to the level  $q_k$  is computed, and is given by,

$$\frac{\partial \mathcal{D}_{\text{MSE}}}{\partial q_k} = 2(q_k - \mu_{[q_{k-1}, q_k]})\mathbb{P}\{X \in (q_{k-1}, q_k]\} - (q_{k+1} - q_k)^2 f(q_k), \quad (3.9)$$

where the notation  $\mu_I$  represents the conditional mean of  $X$  in the interval  $I$ . Note that the terms in MSE gradient, except for the conditional mean, are similar to the terms in MAE gradient. Thus, the level update step for MSE based envelope quantizer follows the two-time scale procedure akin to MAE case. Due to space constraints, we provide the details of this update rule in the supplementary section (Appendix B).

## 3.5 Convergence analysis of SANE quantizer

In this section we discuss the convergence analysis of the two-time scale stochastic approximation algorithm described earlier. Due to space constraints, we will restrict the proof for the mean absolute error cost function. However, the fundamental steps remain the same for the mean squared error cost as well. Because two-time step processes are well studied [68, 71], we only sketch the relevant parts pertaining to the convergence analysis of SANE.

**Theorem 3.1.** *The iterates of the two-scale stochastic approximation described in (3.7)*

and (3.8) converge the solution trajectory of the ODE,

$$q'_k(t) = \frac{\partial \mathcal{D}_{MAE}}{\partial q_k} \quad \text{for } k \in \{1, 2, \dots, K\}.$$

Thus, in the limit  $t \rightarrow \infty$  (or equivalently  $n \rightarrow \infty$ ) the SANE quantizer levels converge to the optimum envelope quantizer levels obtained by the limiting ODE,  $0 = \frac{\partial \mathcal{D}_{MAE}}{\partial q_k}$ .

*Proof.* To keep the analysis simplified, we shall assume the wavelet density estimate closely tracks the true probability density. The main ideas of the proof related to the ODE approach are borrowed from Borkar (Sec. 5.1 [68]), and Kushner and Yin (Sec. 5.2 [71]). First, we recall the two-time scale update step in (3.8), and rewrite the same by adding and subtracting the expectation of the indicator function,  $\mathbb{1}_{X(n)}\{(q_{k-1}(n), q_k(n+1))\}$ . Therefore, we obtain,

$$z_k(n+1) = (1-b(n))z_k(n) + b(n)e(n) + b(n)\mathbb{E}(\mathbb{1}_{X(n)}\{I_n\}), \quad (3.10)$$

where the notation  $e(n) := \mathbb{1}_{X(n)}\{I_n\} - \mathbb{E}(\mathbb{1}_{X(n)}\{I_n\})$ , and the interval  $I_n := (q_{k-1}(n), q_k(n+1)]$ . On repeated substitution using previous iterates, we analyze the fast component (or the update equation (3.8)) taking slow one (or the update equation (3.7)) as frozen at a constant value, parameterized by  $q_k(n)$ . Thus, when  $b(n) = b$  (a constant), equation (3.10) becomes

$$z_k(n+1) = (1-b)^n + \sum_{i=1}^n \xi_i + b \sum_{i=1}^n (1-b)^{n+1-i} \mathbb{P}_i, \quad (3.11)$$

where  $\xi_i := b(1-b)e(i)$  for  $i = \{1, 2, \dots, n-1\}$ ,  $\xi_n := e(n)/(1-b)$ , and  $\mathbb{P}_i := \mathbb{P}\{X(i) \in I_i\}$ .

Next, we analyze the slow component treating the fast one as equilibrated at the unique equilibrium of the fast component that is parametrized by the slow one. As  $n \rightarrow \infty$ , in (3.11) the first term diminishes to zero and the last term converges to the true probability of a data sample falling in the interval  $\lim_{n \rightarrow \infty} I_n$ . This convergence holds only if  $a(n)$  is chosen such as  $\frac{a(n)}{b} \rightarrow 0$ . Further, the middle term,  $\sum_i \xi_i$  in the equation represents the zero mean stationary noise arising due to the stochastic approximation, whose contribution dies down to zero. Since zero mean stationary noise can still cause oscillations (explained in [71]), we have to choose a decreasing sequence  $b(n)$  such that  $\frac{a(n)}{b(n)} \rightarrow 0$ , to ensure non-oscillatory convergence.

Such a choice of  $b(n)$  would result in a residual noise to be a martingale difference, which dies down to zero if  $\sum_n b(n) = \infty$  and  $\sum_n b(n)^2 < \infty$  [68]. Thus, for a sufficiently large  $n$ , the quantizer level update equation can be rewritten as,

$$q_k(n+1) = q_k(n) - a(n) \left\{ \mathbb{P}\{X_n \in \mathcal{I}_n\} + M(n) - (q_{k+1}(n) - q_k(n)) \widehat{f}(q_k(n)) \right\}, \quad (3.12)$$

where  $\mathcal{I}_n := (q_{k-1}(n), q_k(n)]$ , and  $M(n)$  represents the martingale difference noise.

For completeness, we consider the wavelet density estimation error obtained, by assuming differentiability of the density function  $f(x)$ . The wavelet density estimate,  $\widehat{f}(x)$  is known to satisfy the condition,  $\|\widehat{f} - f\|_2 \leq Cn^{-\alpha}$ , where  $C > 0$  and  $\alpha \in (0, 1]$  [76]. Using this fact, we observe that the proposed stochastic approximation scheme is a noisy discretization of the ODE  $q'_k(t) = -\frac{\partial \mathcal{D}_{\text{MAE}}}{\partial q_k}$  (refer [68]). Thus, as  $n \rightarrow \infty$ , the SANE quantizer converges to the optimal envelope quantizer, where the gradient condition,  $\frac{\partial \mathcal{D}_{\text{MAE}}}{\partial q_k} = 0$ , holds.  $\square$

*Remarks:* The ODE approach applied in the above analysis, has been extensively used to determine the convergence rate of gradient descent algorithms (both deterministic and stochastic). Some recent works have evaluated the finite sample convergence concentration bounds (non-asymptotic) of two time-scale stochastic approximation schemes [77, 78, 72].

## 3.6 Performance evaluation of SANE

To evaluate the performance of SANE quantizer, we conducted simulations on synthetic as well as real-world data. For simulations a Windows 10 Notebook running MATLAB 2015b was used along with a wavelet density estimation toolbox [79].

### 3.6.1 Simulations on synthetic data

In this simulation, we consider signal samples taken from the Beta(2,2) distribution which has the support set  $[0, 1]$ . The performance of SANE quantizer is analyzed for  $K = 8$  (or equivalently 3 bit) quantization levels. For studying the influence of the learning rate on the distortion, we choose  $a(n)$  and  $b(n)$  to satisfy the stochastic approximation criteria discussed in Sec. 3.4.1. In particular,  $a(n)$  is chosen in the form  $\frac{1}{\lceil n/S \rceil}$  where  $S$  is a positive integer, and  $b(n)$  is chosen so as to satisfy  $\frac{a(n)}{b(n)} \rightarrow 0$ . For instance, three forms of  $b(n)$

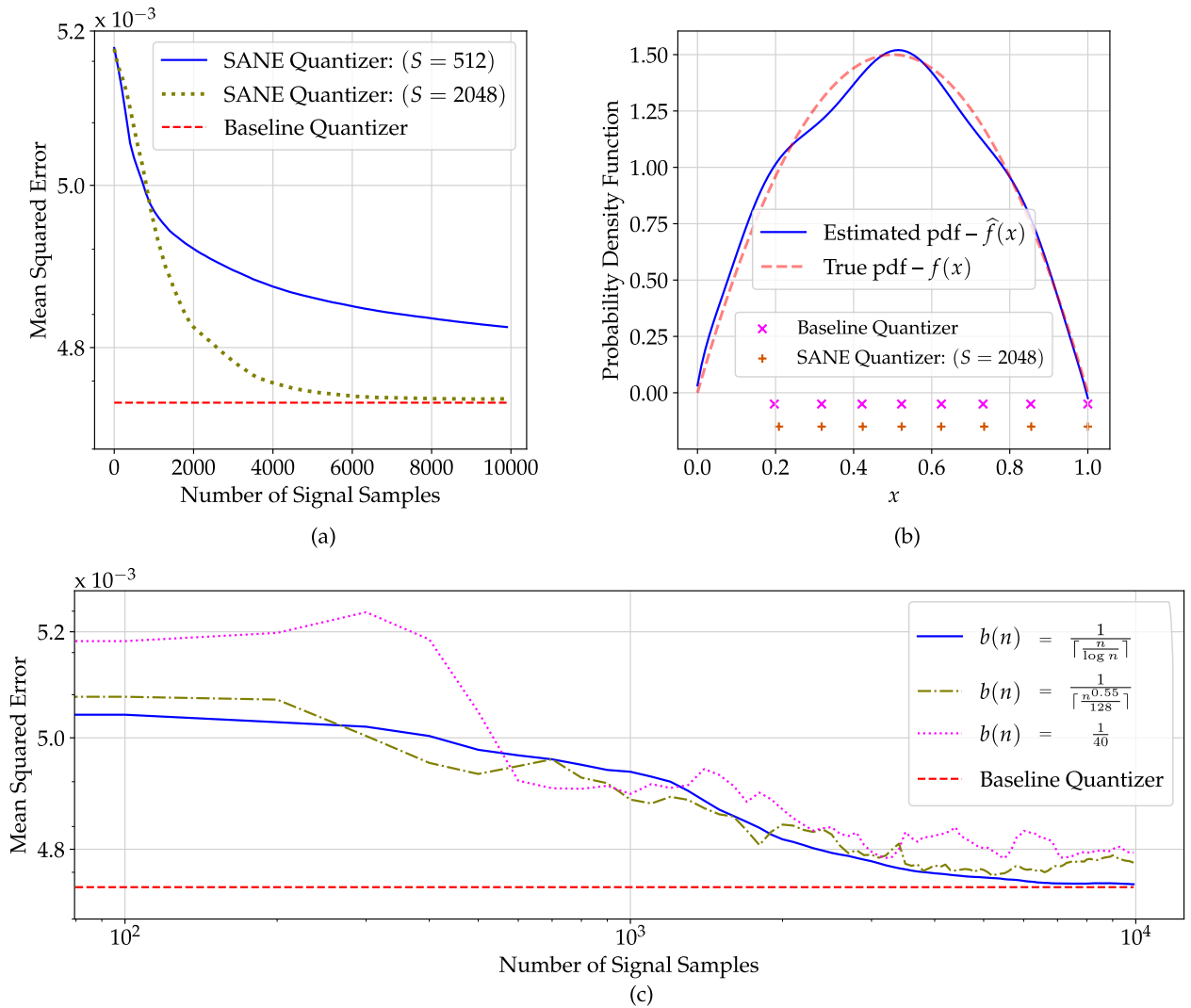


Figure 3.3: The performance of SANE quantizer is studied on synthetic data generated from Beta(2,2) distribution. Plot (a) depicts the MSE decay and convergence to the baseline quantizer with increasing data size. In plot (b), the learned probability density and the envelope quantizer levels are shown. Plot (c) depicts the convergence with different choices of  $b(n)$ .

considered in this simulation are  $\frac{1}{\lceil \frac{n}{\log n} \rceil}$ ,  $\frac{1}{\lceil n^\gamma/K \rceil}$  where  $\frac{1}{2} < \gamma < 1$  and  $K > 0$ , and  $b(n) = b$  (constant).

**Sample complexity performance.** Comparison of the distortion performance (with the MAE criterion) is analyzed using the various choices of learning rates  $a(n)$  and  $b(n)$  discussed above. In Fig. 3.3 (a), we observe the sample complexity plot for the choice of the learning rates,  $a(n) = \frac{1}{\lceil n/S \rceil}$  where  $S \in \{512, 2048\}$  and  $b(n) = \frac{1}{\lceil \frac{n}{\log n} \rceil}$  for  $n \geq 2$ .

It is noted that the quantization error (that is, MSE) of the SANE quantizer approaches the baseline quantizer performance, which is obtained by numerical evaluation of the envelope quantizer with the signal source density assumed to be known. For  $S = 2048$ , the plot is seen to achieve the optimal envelope quantizer performance with nearly  $N = 6000$  signal samples. Generally the subsampling rate is set according to the expected number of training samples. For the experiments considered, we have chosen the subsampling rate,  $S = N/5$ , where  $N$  is the number of training examples, as it holds the value of  $a(n)$  for about 20% of the training size. However this should be taken as a rule of thumb.

Another aspect while designing the SANE quantizer is the choice of  $b(n)$ . In Fig. 3.3(b) we show the convergence trend for three choices of  $b(n)$ . In the first case where  $b(n) = \frac{1}{\lceil \frac{n}{\log n} \rceil}$ , we note a steady convergence as it satisfies the convergence criteria. For the second case  $b(n) = \frac{1}{\lceil \frac{n^{0.55}}{128} \rceil}$ , we see that convergence is oscillatory due to the poor choice of the subsampling rate, leading to the violation of convergence criteria. Finally in the case where  $b(n) = \text{constant}$ , the oscillations are prominent although there is an overall trend of convergence.

**Quantizer convergence performance.** In Fig. 3.3 (b), the envelope quantizer levels obtained through the stochastic approximation scheme is compared with the optimal (baseline) levels, which considers the underlying signal source pdf to be known. This baseline quantizer is computed numerically (for the Beta(2,2) pdf) using a procedure similar to the Lloyd's algorithm [24]. It is noted that the envelope constraint forces the highest quantizer level to be assigned to the largest signal value in the support set. Thus, in the considered simulation with  $K = 8$  quantization levels, the highest level  $q_8 = 1$ . In the plot, the wavelet density estimate is also depicted, for  $N = 10^4$  signal samples and sliding window length  $M = 500$ .

**Wavelet density estimation performance** The effect of the sliding window size of the wavelet based density estimator on the quantizer distortion is studied in Table. 3.1. We considered window sizes varying from  $M = 10$  to  $M = 1000$ . It is noted that the wavelet density estimation error (measured as MISE) diminishes with increasing window size. In contrast, this diminishing trend shows little influence on the mean-square error distortion of the envelope quantizer. This motivates us to look for approximate density

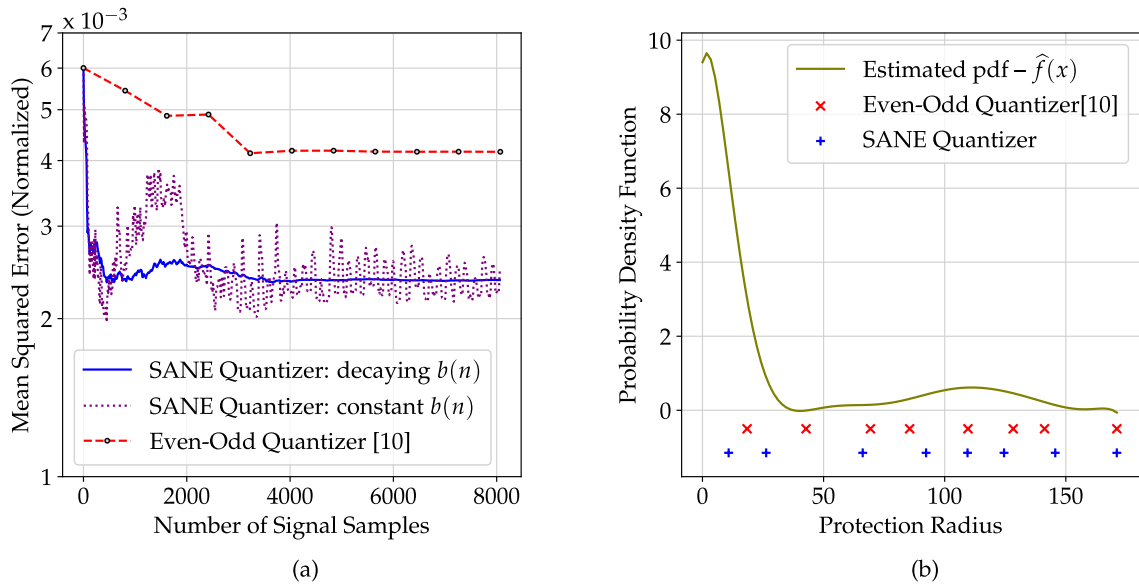


Figure 3.4: SANE quantizer is applied on the TV white spaces protection radius dataset. Plot (a) illustrates the convergence of SANE quantizer compared to other envelope quantizers. Further in plot (b), the obtained quantization levels of SANE are seen to cluster near the local maxima in the estimated density.

representations [80, 15], that will reduce the computational complexity, in a future work.

Table 3.1: Performance of SANE with varying sliding window length

Sliding Window length (data samples)	Quantization Error of SANE (MSE)	Wavelet Density Estimation Error (MISE)
10	$4.68 \times 10^{-3}$	$87.69 \times 10^{-3}$
100	$4.67 \times 10^{-3}$	$10.23 \times 10^{-3}$
600	$4.66 \times 10^{-3}$	$1.54 \times 10^{-3}$
1000	$4.65 \times 10^{-3}$	$2.17 \times 10^{-3}$

### 3.6.2 Simulations on TV white Space dataset

To study the performance of SANE quantizer on real-world data, we consider the TV white space protection radius dataset [57]. This dataset has information on 8076 primary TV transmitters across channels 2 to 51 in the United States. For the SANE quantizer design, we set the end point  $q_K = 171.05$ , which is the maximum protection radius in the dataset. By fixing the learning rates  $a(n) = \frac{1}{\lceil n/256 \rceil}$  and  $b(n) = \frac{1}{\lceil \frac{n}{\log n} \rceil}$  for  $n \geq 2$ , and using an equi-spaced initialization, we plot the MSE performance in Fig. 3.4 (a). We note that SANE quantizer converges to a lower MSE compared to the 'alternating even-odd' quantizer design proposed in [66]. Further, on setting  $b(n) = \frac{1}{40}$  an oscillatory behaviour of MSE is observed. This is because of the violation of the condition,  $\sum b(n)^2 < \infty$ .

In Fig. 3.4 (b), the quantizer levels of the 3-bit (that is  $K = 8$ ) SANE quantizer is shown relative to the estimated probability density. The probability density estimate plot suggests that a majority of the TV transmitters have protection radius in the range 0-35 Km. Unlike the synthetic dataset, we see that this real-world dataset has a bi-modal distribution (two peaks). For the data driven quantizer, the bi-modal setting is challenging due to the chances of the quantizer levels getting trapped in a local minima. However, for the SANE quantizer design, the levels tend to move closer to the peaks in the probability density function. This indicates the ability of SANE quantizer to come out such local traps. In contrast, the even-odd quantizer has higher chance of giving a suboptimal performance, if the quantization levels are not carefully initialized.

## 3.7 Approximate envelope quantizer and its variants

In this part we introduce a model driven quantizer design which is motivated by the Approximate Lloyd-Max (ALM) quantizer discussed in Chapter 2. Piecewise linear approximation of the source probability density will be used to update the quantizer levels in addition to considering the envelope constraints. This modified quantizer is termed as Approximate Envelope Quantizer (AEQ), and we will assume the probability density function of the source signal is known.

### 3.7.1 Cost minimization and level updates for AEQ

Let the mean square error cost with the envelope constraint imposed be denoted by  $\mathcal{D}_{\text{env}}(Q)$ . We define,

$$\begin{aligned} \mathcal{D}_{\text{env}}(Q) &:= \mathbb{E} [(Q(X) - X)^2] \quad \text{where } Q(X) \geq X, \\ &= \int_0^1 (Q(x) - x)^2 f_X(x) dx \quad \text{where } Q(x) \geq x, \\ &= \sum_{k=1}^K \int_{q_{k-1}}^{q_k} (q_k - x)^2 f_X(x) dx. \end{aligned} \quad (3.13)$$

The simplification in the cost function above is performed by substituting  $Q(x) = q_k$  for  $q_{k-1} \leq x \leq q_k$ . It is observed that the total cost can be minimized by taking partial derivative with respect to the each quantization level  $q_k$ ;  $k = 1, \dots, K$ . The minima corresponds to equating the partial derivative to zero. That is,

$$\begin{aligned} 0 &= \frac{\partial \mathcal{D}_{\text{env}}(Q)}{\partial q_k} \\ &= \frac{\partial}{\partial q_k} \int_{q_{k-1}}^{q_k} (q_k - x)^2 f_X(x) dx + \frac{\partial}{\partial q_k} \int_{q_k}^{q_{k+1}} (q_{k+1} - x)^2 f_X(x) dx \\ &= \int_{q_{k-1}}^{q_k} 2(q_k - x) f_X(x) dx - (q_{k+1} - q_k)^2 f_X(q_k) \end{aligned} \quad (3.14)$$

In the above equation, we note that the nearest neighbor levels of  $q_k$  are  $q_{k-1}$  and  $q_{k+1}$ , which can be assumed to be fixed while updating  $q_k$ . This implies that the quantizer level updates can be performed simultaneously for all even (or odd) quantizer indices, while fixing the odd (or even) quantizer indices. Since the modified quantization levels can be determined by concurrent updates of even and odd sets, we term this procedure as *Alternating Between Evens and Odds (ABEO)* [66]. This update rule will considerably speed up the proposed envelope quantizer design. In general (3.14) does not have a closed-form solution, hence we suggest a linear approximation based algorithm for envelope quantizer updates. This method is similar in spirit to piecewise linear approximation considered in ALM. However, we will need to accommodate the envelope constraint, that is  $Q(x) > x$ , while updating the quantizer levels.

#### Linear approximation based algorithm for AEQ

The linear approximation method described for ALM (in (2.6)) provides a template for finding a closed-form expression for the optimal AEQ levels,  $q_k$ . We rewrite the sufficient

Table 3.2: Coefficients of the cubic polynomial equation  $p(u) = p_0 + p_1u + p_2u^2 + p_3u^3$ , to determine optimal level updates of AEQ

Coeff.	$1 \leq k \leq K$
$p_0$	$\frac{2}{3}m_k q_{k-1}^3 + c_k (q_{k-1}^2 - q_{k+1}^2)$
$p_1$	$-2c_k(q_{k-1} - q_{k+1}) - m_k(q_{k+1}^2 + q_{k-1}^2)$
$p_2$	$2m_k q_{k+1}$
$p_3$	$-\frac{2}{3}m_k$

condition for optimality using the approximate density function  $f_{\text{app}}(x)$  as

$$0 = \int_{q_{k-1}}^{q_k} 2(q_k - x)f_{\text{app}}(x)dx - (q_{k+1} - q_k)^2 f_{\text{app}}(q_k). \quad (3.15)$$

On substituting

$$f_{\text{app}}(x) = m_k x + c_k, \quad \text{for } x \in [q_{k-1}, q_{k+1}] \text{ and } 1 \leq k \leq K - 1, \quad (3.16)$$

we obtain a third order polynomial equation,  $p(u) = p_0 + p_1u + p_2u^2 + p_3u^3$ . The coefficients  $p_j$ ;  $j = 0, 1, 2, 3$  depends on the nearest neighbor levels,  $q_{k-1}$  and  $q_{k+1}$ . We list these coefficients in Table 3.2. The roots of the cubic equation  $p(u) = 0$  in the interval  $[q_{k-1}, q_{k+1}]$ , corresponds to the optimum level update of  $q_k$ . We show the existence of atleast one real root of  $p(u)$  in  $[q_{k-1}, q_{k+1}]$  in Appendix B.3. Also, we observe that  $p(u)$  has a positive slope at  $u = q_k$  (see Appendix B.4). The proposed linear approximation based quantization scheme is described in Algorithm 2.

### 3.7.2 Performance of AEQ on sources with beta distribution

We now consider simulations to evaluate the performance of AEQ on known distributions. Particularly, we consider the signal sources with a unimodal PDF defined over a bounded support [30, 31]. Beta distribution is one such signal source, which is parameterized by  $\alpha$  and  $\beta$ . In this section we consider  $\alpha = 4$  and  $\beta = 2$ , which is an asymmetric PDF.

**Algorithm 2** Scalar Envelope Quantizer Algorithm**Input** : Input distribution  $f_X(x)$ ,  $K =$  Number of levels, MaxIter, Threshold**Output** : List of quantization levels,  $\vec{q}$ **Initialization:**  $\vec{q}^{(0)} = [0, \frac{1}{K}, \frac{2}{K}, \dots, 1]$ , stop condition = False,  $i = 0$ , dist = 0**1 while** !stop condition **do****2**  $\mathcal{Q}_{\text{odd}} = \{q_1, q_3, \dots, q_{2m+1}\}$  $\mathcal{Q}_{\text{even}} = \{q_2, q_4, \dots, q_{2l}\}$ % where  $\max\{2l, 2m + 1\} = K - 1$ **for** (In Parallel)  $q_k$  in  $\mathcal{Q}_{\text{odd}}$  **do****3**

| Set linear approximation parameters :

$$\text{Slope: } m_k = \frac{f_X(q_{k+1}) - f_X(q_{k-1})}{q_{k+1} - q_{k-1}},$$

Intercept:  $c_k = f_X(q_{k+1}) - m_k q_{k+1}$ , and

$$p(u) = p_0 + p_1 u + p_2 u^2 + p_3 u^3 \text{ (see Table 3.2)}$$

$$q_k^{(i+1)} \leftarrow \{v \in [q_{k-1}, q_{k+1}] : p(v) = 0\} \quad \text{Note:}(\text{Im}(r) = 0)$$

**4** **end****5** **for** (In Parallel)  $q_k$  in  $\mathcal{Q}_{\text{even}}$  **do****6**| Update  $q_k$  with steps in  $\mathcal{Q}_{\text{odd}}$  loop above**7** **end****8** dist  $\leftarrow \mathcal{R}(\vec{q}^{(i+1)})$ ;  $i \leftarrow i + 1$ **9** **if** (dist < Threshold) or (iter > MaxIter) **then****10** | stop condition = True**11** **end****12 end**

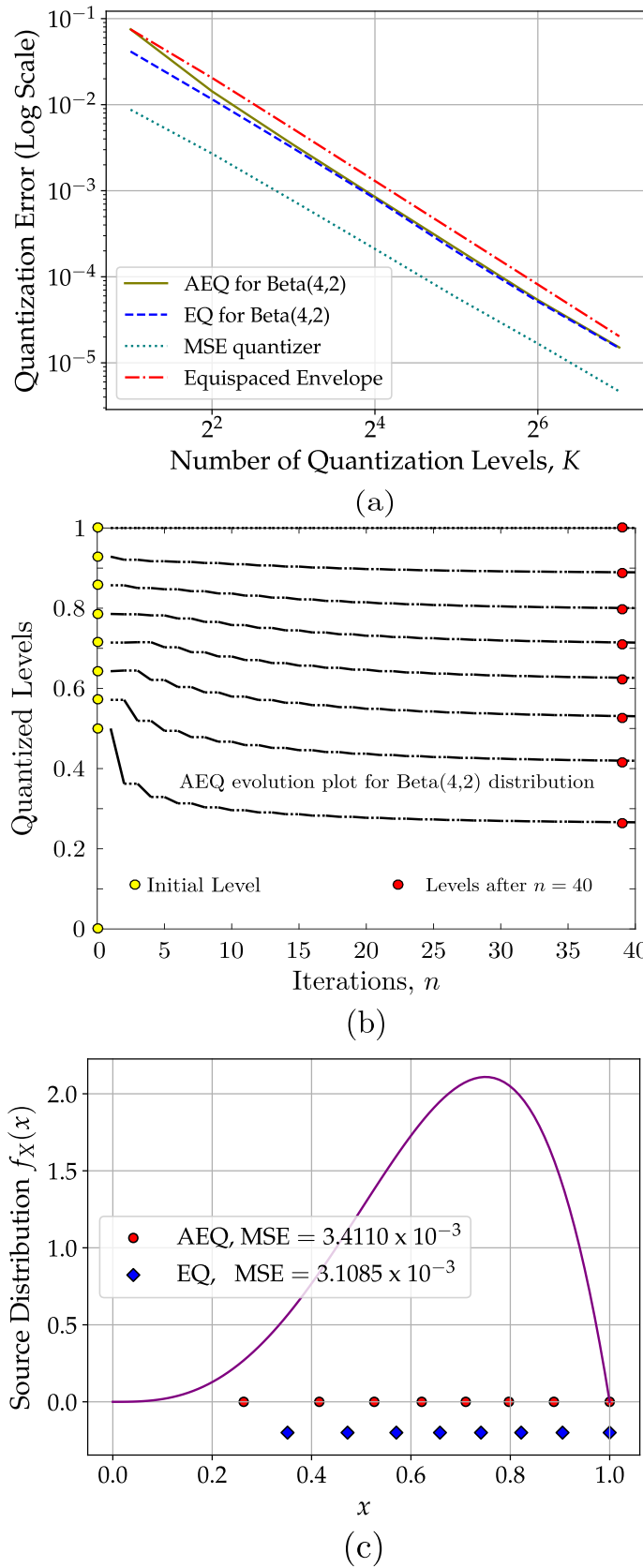


Figure 3.5: AEQ algorithm performance for Beta(4,2) source. In (a) the MSE of AEQ is compared with the optimal EQ, the equispaced envelope quantizer, and the MSE optimal quantizer. Plot (b) shows the evolution of the AEQ algorithm for  $n = 40$  iterations. (c) shows the alignment of the AEQ levels with respect to the optimal EQ levels.

In Fig. 3.5 we study MSE performance and convergence of AEQ levels. The quantization error or MSE of AEQ is shown in comparison to the optimal Envelope Quantizer (EQ) and the equispaced envelope quantizer (see Fig. 3.5 (a)). Here, EQ corresponds to the equivalent of the Lloyd-Max quantizer which accounts for the envelope constraint and the equispaced envelope quantizer is a linear quantizer with uniform step size in the interval  $[0, 1]$ . It is seen that AEQ performance approach the EQ performance with the increase in the quantization levels,  $K$ . For reference, we also show the performance of the optimal MSE quantizer which is designed without considering the envelope constraints.

Next, we analyze the evolution of the AEQ levels at different iterations, beginning with a equispaced initialization. It is observed that the convergence of levels happens at  $n \approx 20$  when  $K = 8$ . In general, we see experimentally that the convergence of levels happen for  $n = \mathcal{O}(cK)$  where  $c \leq 5$ . In Fig. 3.5 (b), we observe that AEQ aligns the quantization levels closer to the regions where Beta(4,2) PDF has a larger value, i.e. in the interval  $[0.6, 0.9]$ , thus indicating the near-optimal behaviour. Finally, Fig. 3.5 (c) considers a comparison of the quantization levels obtained from AEQ and the EQ quantizers. The plot depicts higher conformity of AEQ and EQ near the peak of the PDF, and deviation of the levels for regions where the PDF is close to zero. Reason for the deviation is the approximation error due to piecewise linear fitting, which is large when  $f_X(x) \approx 0$ . This deviation of the quantization levels has only a small influence in MSE performance and thus the quantization errors of AEQ and EQ are of the same order.

*Remark:* The convergence of AEQ can be shown by theory developed for ALM based on Perron-Frobenius analysis. The procedure developed in Sec. 2.4 accommodates the level updates of AEQ by expressing them as convex combination parameterized by  $\theta_k^{(i)}$ . To avoid repetition, we avoid a separate analysis for establishing the exponential convergence rate and asymptotic near-optimality of AEQ.

### 3.7.3 Extension to Learning AEQ

By using a density estimate similar to LALM (see equation (2.7) and (2.8)), we can estimate the piecewise linear estimate and use the AEQ as a plugin estimate over the

approximate density function. The slope and intercept estimates, i.e.

$$\hat{m}_k = \frac{\hat{f}_{k+1} - \hat{f}_{k-1}}{q_{k+1} - q_{k-1}}, \quad \hat{c}_k = \frac{q_{k+1}\hat{f}_{k-1} - q_{k-1}\hat{f}_{k+1}}{q_{k+1} - q_{k-1}},$$

can be plugged into the cubic polynomial described in Table 3.2 and solved even in a hardware constrained devices using well known numerical solvers such as the Newton's method [81].

For signal sources with a unimodal probability density the Learning AEQ (LAEQ) weakly converges to the model driven AEQ with increasing data samples. The convergence claim follows since the estimate of the slope and intercept approaches the actual slope and intercept respectively of the piecewise linear approximation. Thus LAEQ has similar convergence properties as studied under LALM (see Sec. 2.3.2). Next, we will see an example where LAEQ fails when the source probability density is non-unimodal with a zero a probability region between the modes of the PDF.

Consider the TV whitespace dataset introduced in Sec. 3.6.2 which has a bimodal probability density as shown in Fig. 3.6 (b). Observe the zero probability region in the interval  $[30, 50]$ , representing the TV whitespace protection radius,  $r$ , where  $30 \leq r \leq 50$ . As the LAEQ algorithm initialized with equispaced quantizer levels, it fails to approximate the probability density in the range  $[0, 50]$  and result in a suboptimal design. Further, in Fig. 3.6, we observe that the number of quantization levels required for LAEQ to match the MSE performance of SANE is very high. More precisely, LAEQ requires  $K \approx 17$  to reach the same performance of SANE, which requires only  $K = 8$  levels.

### 3.8 Summary

In this chapter, we motivated the need for envelope quantizers, which always overpredicts the observed signal samples. A stochastic approximation-based quantizer design (known as SANE) was proposed by accounting for the overprediction constraint. The convergence properties of the proposed quantizer were analyzed, and these properties were validated through simulations on synthetic and real-world data. Further, we proposed the AEQ algorithm by extending the piecewise linear approximation idea developed for ALM and incorporating the additional envelope constraints. A comparative study was performed to understand the benefits and failures of SANE and AEQ. We envisage vector extensions

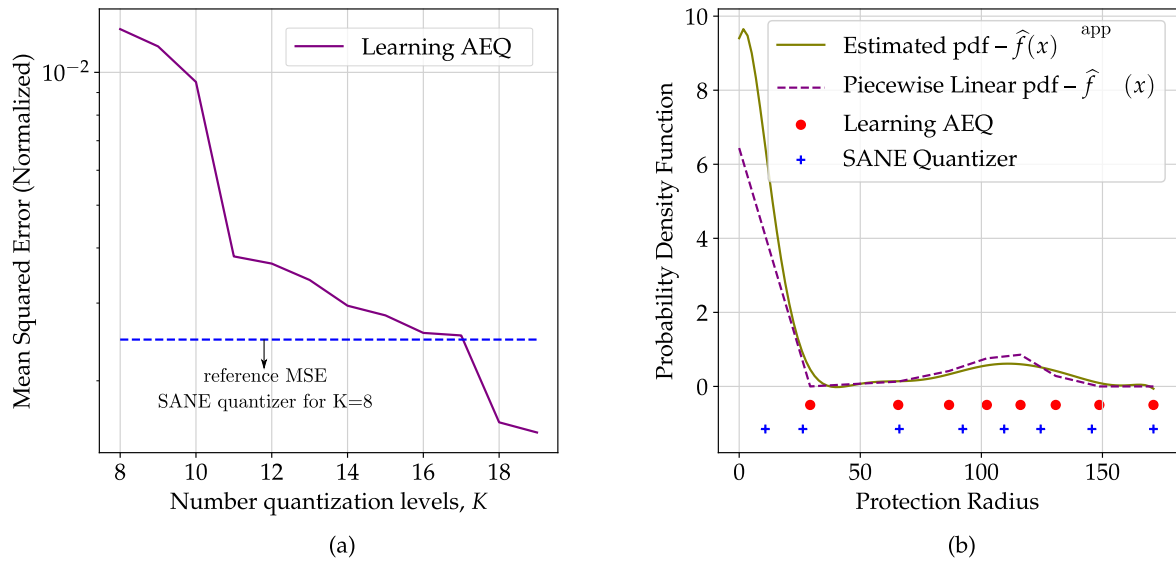


Figure 3.6: An example of LAEQ failure for the TV whitespace dataset with a non-unimodal probability density. (a) shows the decay of mean square error for LAEQ with respect to the reference MSE of SANE for  $K = 8$ . In (b), we depict the piecewise linear approximation of LAEQ and compare its quantization levels with SANE.

of these algorithms in future work.



# Chapter 4

## Distributed Quantizer Design and Tradeoffs in Federated Learning

In this chapter, we develop data-driven quantizer designs for a distributed client and server architecture proposed in federated learning by hinging on the algorithms proposed in the previous chapters. Due to the rising concerns related to data communication and privacy, the edge devices (clients) may not send their raw training data but only a compressed representation relevant for learning a global quantizer. For the quantizer schemes considered, we study the tradeoffs between the mean-squared error due to quantization, the client-server communication cost, and the training data size (sample complexity). Analytically we show that the mean-squared error of the proposed quantizer schemes is less than a (weighted) average of the mean-squared errors for quantizers at respective clients, and this is verified using an available fitness device-based dataset.

### 4.1 Background

Distributed quantizer design and aggregation methods in a federated learning setting [1], having decentralized training data across several clients, will be the main focus of this chapter. A global quantizer, designed using FL, will allow all client devices to access a better quantizer in terms of distortion metrics such as the mean-squared error while incorporating training samples from several client devices. Moreover, the federated design reduces the client-server communication cost by sending only the quantizer levels instead

of the entire training data. Thus, federated learning-based quantizers naturally preserve privacy. For instance, consider Lloyd’s algorithm or  $k$ -means quantizer [24] implemented at the client devices, and the aggregation of these quantizers performed at the central server using a proportional weighted average scheme. Clients or *edge devices* usually have heterogeneous hardware specifications, training data sizes, and battery levels, which necessitates the edge devices to choose an appropriate quantization algorithm.

There are several applications where learning such a global quantizer model is necessary. For instance, it can be used for data compression at the edge devices to store the historical data. Another application uses the learned quantizer levels for performing inference tasks at the edge devices by using the statistical distribution learned along with the quantizer. In applications where privacy is not a concern, the global quantizer can also encode the data samples to the server using a joint codebook shared across all edge devices.

#### 4.1.1 Related works

Quantization is a classic topic in data compression and statistics; it has been well studied in literature [28, 18]. Lloyd’s algorithm, also known as the  $k$ -means [24], is a widely used clustering technique in machine learning and pattern recognition. Efficient computational methods for improving the speed and reducing the sample complexity of quantizers have been dealt with previously, in the works by [82, 34, 12, 83, 15]. Distributed quantization methods have been proposed for parameter estimation in the context of wireless sensor networks [84]. As far as we know, there are no previous works in literature that addresses the distributed design of quantizers under the federated learning model.

Recent research efforts in federated learning discuss several system-level challenges related to the client-server architecture. These include model heterogeneity, network asymmetry or bandwidth constraints, and data privacy [1]. Others include the design of communication efficient quantization schemes suited for distributed mean computation [85], fairness aware federated optimization to mitigate training procedure bias [86], as well as designing federated learning systems to accommodate the scale and synchronization [87]. Xu and Wang [88] presents a review of various federated learning algorithms, emphasizing solutions to statistical, system, and privacy challenges.

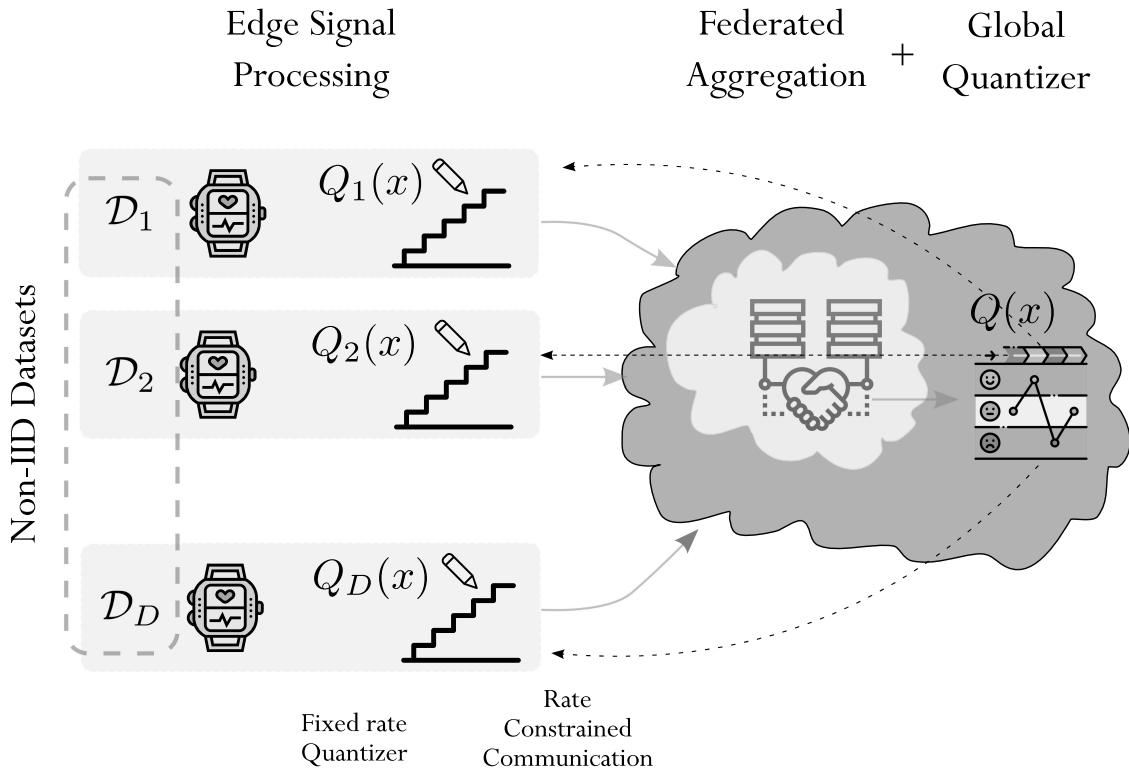


Figure 4.1: The federated learning architecture for the design of a global quantizer  $Q(x)$  using the individual data driven quantizers,  $\{Q_1(x), \dots, Q_D(x)\}$ , learned from the datasets  $\{\mathcal{D}_1, \dots, \mathcal{D}_D\}$ , which are non-IID. This model assumes fixed rate quantizer design through a rate constrained uplink communication between the edge devices and the central server.

## 4.2 Federated quantizer model

Consider a federated learning model where there are  $D$  client devices and a central server, as shown in Fig. 4.1. Each device has training data residing locally, represented by the datasets  $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_D$ . We will assume that these datasets are non-identical but independent with possibly different means and dynamic ranges. Moreover, we assume a fixed rate (scalar) quantizer at the client devices, with  $K$  levels. Each device can choose a suitable quantizer design algorithm, based on their communication, computation and energy (battery) constraints. Let  $Q_d(x) : x \rightarrow \{q_{d,1}, \dots, q_{d,K}\}$  for  $d \in \{1, 2, \dots, D\}$ , represent the quantizer function at the devices. The client devices will only send their learned quantizer level vector  $\vec{q}_d = [q_{d,1}, \dots, q_{d,K}]$ , along with certain additional parameters, which will aid in the federated aggregation. Finally, the central server combines these quantizers using one of the linear aggregation schemes discussed later.

The mean-squared error is used as the distortion metric to quantify the performance of the quantizer. Since the quantizer is data-driven, this can be measured in terms of both train and test data. We define the empirical metric,

$$\text{MSE}_{\text{emp}} := \frac{1}{N} \sum_{x \in \mathcal{D}} (Q(x) - x)^2,$$

where  $N$  is the size of the dataset  $\mathcal{D}$ , and  $Q(\cdot)$  is any generic quantizer function. However, for mathematical analysis we can use the mean-squared error as,

$$\text{MSE} := \mathbb{E} [(Q(X) - X)^2],$$

where the expectation is taken with respect to the (unknown) source distribution  $f(x)$ .

**Remark 4.1.** *In this work, being the first exposition into distributed quantizer design using federated learning, we consider only the scalar (i.e., one-dimensional) case. However, we believe that these results can be extended to the vector quantizers, which is backed by some of our experiments.*

## 4.2.1 Motivating examples for learning a global quantizer

### 1. Quantized stochastic gradient descent

In federated learning, the participating devices will aid in training a global machine learning model (e.g., Deep Neural Networks) by exchanging the gradient information. The client devices need to update the weight matrix of the global model using stochastic gradient descent (SGD). Due to the bandwidth limitation of the client-server link, a finite precision quantized SGD update will be desired. As the devices themselves have access to only a subset of (heterogeneous) data, the SGDs computed will be non-IID in nature. A global quantizer designed using the proposed methods will enable information exchange between client and server using a common codebook.

### 2. Calibration of low accuracy sensors

Distributed IoT devices, for instance, environmental monitors and healthcare wearables, often have low cost and low accuracy sensors. For performing federated learning, these devices need to be calibrated to improve the inference accuracy. Since quantizer design is a crucial step in the inference process, there is a need to learn a calibrated global quantizer

based on the distributed non-homogeneous datasets. The global quantizer can be further personalized at the IoT nodes based on the calibration model learned locally.

**Remark 4.2.** *Our discussions are centered around design of a global quantizer using federated learning. The part on personalization of quantizers will be part of a future work.*

Next, we summarize the main contributions in this chapter.

## 4.2.2 Quantizer design algorithms considered in this chapter

**Quantizer Design at Clients:** We consider four candidate algorithms that can be implemented at the edge devices. These are equispaced (ES), Learning Vector Quantizer (LVQ), Learning-ALM (LALM), and Lloyd’s or  $k$ -means algorithm. We summarize these quantization algorithms under four different performance metrics - viz. hyperparameter requirement, computational complexity, sample complexity and space complexity, in the Table 4.1.

	ES	LVQ	LALM	$k$ -means
Hyper-parameters	✗	✓	✗	✗
Computational complexity	0	$O(NK)$	$O(K(N + I))$	$O(NKI)$
Sample Complexity	$O(1)$	–	–	$O\left(\frac{1}{\sqrt{N}}\right)$
Space Complexity	$K$	$K + 1$	$3K + 2$	$N + K$

Table 4.1: A summary of performance of various scalar quantization methods possible at the client devices, and their properties.

**Federated Aggregation:** We analyze different federated aggregation schemes that are possible at the central server. Broadly we classify these schemes as, (i) Proportional weighted aggregation, (ii) Number based weighted aggregation, (iii) Probability score based aggregation, and (iv)  $k$ -means based aggregation. The symbols for these schemes, communication bits required, and the related analytical contributions are illustrated in

Table 4.2. All these aggregation schemes are based on convex linear combinations which are parametrized by the signal source probability density of each client device.

Scheme	Symbol used	Communication (bytes)	Results
Proportional weighted	$\pi_{\text{prop}}$	$K + 1$	Theorem 4.3
Number based	$\pi_{\text{num}}$	$2K$	Corollary 4.5
Probability-score based	$\pi_{\text{ps}}$	$2K$ or $2K + 2$	Algorithm 4
$k$ -means based	$\pi_{\text{kM}}$	$2K$	Section 4.4.4

Table 4.2: A summary of various federated aggregation schemes possible at the central server, and their communication cost.

### 4.2.3 Tradeoffs in federated learning

In this work, we characterize the three-fold tradeoffs between the mean-squared error of the designed quantizers by federated learning, the number of communication bits sent across the clients and the server, and the number of training samples (sample complexity). These tradeoffs are studied using experiments on a synthetic dataset, a fitness based dataset and the standard MNIST dataset.

## 4.3 Review of quantization schemes

We consider four quantization schemes, which are discussed in the increasing order of their computational complexity.

### 4.3.1 Equispaced (or uniform) quantizer

An equispaced quantizer, is an elementary quantizer design, which initializes the quantization levels at equal intervals, irrespective of the data samples it has observed. If the data samples were chosen from the  $[0, 1]$  interval, the  $K$ -level equispaced scheme, sets the

quantization levels as  $[\frac{1}{K+1}, \frac{2}{K+1}, \dots, \frac{K}{K+1}]$ . Since this design ignores the data statistics, it is suboptimal for all source distributions except for the uniform distribution. However, for high quantization rates the equispaced quantizer is order optimal [18], with a mean-squared error decay proportional to  $2^{-2b}$  [89], where  $b$  is the number of representation bits for a quantization level. From the perspective of federated quantizer design, this scheme is suitable for edge devices with minimal or no compute hardware. Since there are no computations, the sample and space complexities are independent of the number of data samples, and often a preferred way to save computational cost.

### 4.3.2 Learning vector quantizer

The Learning Vector Quantizer or LVQ is a stochastic gradient based algorithm proposed by [34]. It is suited for an online learning setting, where data arrives as a stream or in small batches. In its simplest form, the update step requires only the current data sample, from which the gradient is computed with respect to the nearest neighbor quantization level. The update equations are,

$$j = \arg \min_i |q_i - x_n|,$$

$$q_j \leftarrow q_j - \gamma_n (q_j - x_n).$$

The learning rate (hyperparameter)  $\gamma_n$  has to be chosen according to the stochastic gradient descent conditions [67],

$$\sum_n \gamma_n = \infty \text{ and } \sum_n \gamma_n^2 < \infty.$$

Most commonly  $\gamma_n$  is chosen as  $\frac{1}{\lceil n/s \rceil}$ , where  $s$  is a positive integer representing the subsampling rate. The LVQ convergence rate depends on the choice of  $\gamma_n$ , hence it requires a separate validation dataset to learn this hyperparameter. It is not suitable for applications with low training data.

Because the LVQ computes distances to all the  $K$  quantization levels for every data sample, its computational complexity is  $O(NK)$ . Convergence and sample complexity of LVQ, are predominantly studied with respect to classifier design [90]. However, the analytical characterization of the sample complexity of LVQ when used for quantization (or clustering) are not addressed in literature. In terms of the space complexity, LVQ is efficient, as it is a constant – which is decided by the batch size.

### 4.3.3 Learning Approximate Lloyd-Max quantizer

This is a data-driven quantizer design algorithm based on the approximate Lloyd-Max (ALM) quantizer [15]. The ALM quantizer discussed in Chapter 2, is a low complexity (density) model based quantizer, used to bypass the integral computations in the Lloyd-Max algorithm. It is known to have fast convergence property, and is also nearly-optimal. For comparison purpose, we use the core design steps from the ALM, to extend it to a training based quantizer, termed as the learning ALM (LALM). The proposed algorithm, determines an estimate of the slope and intercept parameters of the approximate piecewise linear density function, which will be later used to determine a plug-in estimator for the ALM based updates. The steps of the LALM quantizer design are summarized in Algorithm 3. In the simulation experiments, we observe that the LALM algorithm requires much lesser number of training samples than the classical  $k$ -means quantizer, which is due to the near-optimality of LALM and the sample complexity reduction in estimating an approximate density instead of the true density.

---

#### Algorithm 3 LALM Quantization Algorithm

---

**Input:** data  $X_i$ , size  $N$ ; num of levels,  $K$

**Initialize:** quantizer  $\vec{q} = [\frac{1}{K+1}, \dots, \frac{K}{K+1}]$ ; reference  $q_0 = 0$  and  $q_{K+1} = 1$

**for**  $k \in \{1, 2, \dots, K\}$  **do**

– Density estimates –  $\hat{f}_{k-1}$  and  $\hat{f}_{k+1}$  are obtained, using piecewise linear model for interval  $[q_{k-1}, q_{k+1}]$

– Slope & intercept estimates are computed–

$$\hat{m}_k \leftarrow \frac{\hat{f}_{k+1} - \hat{f}_{k-1}}{q_{k+1} - q_{k-1}}, \quad \hat{c}_k \leftarrow \frac{q_{k+1}\hat{f}_{k-1} - q_{k-1}\hat{f}_{k+1}}{q_{k+1} - q_{k-1}}$$

– Plug-in estimator of  $q_k$  (obtained from ALM) is applied,

$$\hat{q}_k \leftarrow \frac{1}{\hat{m}_k} \left( \text{HM}(\hat{f}_{k-1}, \hat{f}_{k+1}) - \hat{c}_k \right),$$

where  $\text{HM}(\cdot, \cdot)$  is the Heronian mean.

– Set  $q_k \leftarrow \hat{q}_k$

**end for**

---

The LALM algorithm converts the data points into a (sparse) piecewise linear representation, thereby reducing the computational complexity to  $O(NK + KI)$ , where  $I$  is the number of iteration runs. The terms in the above-mentioned complexity expression, arise from the number of computations required for the density estimation step and the plug-in estimation step respectively. Because the LALM algorithm, transforms the data samples to  $2K + 2$  piecewise linear slope and intercept parameters, the space complexity is considerably reduced.

#### 4.3.4 Lloyd or $k$ -means quantizer

The Lloyd's or  $k$ -means quantizer is the earliest known optimal quantizer design algorithm in literature [24, 25, 33]. This method follows a two-step update procedure using data statistics, akin to the distribution-model based Lloyd-Max algorithm. Since  $k$ -means involves distance computation at each step, its computational complexity is high and is of the order of  $NKI$  (where  $I$  is the number of iterations). To overcome this, many variants have been proposed, prominent among these being Elkan's  $k$ -means [12]. Also, the  $k$ -means algorithm performance is known to be initialization dependent, and to make the initialization efficient the  $k$ -means++ scheme is used in practice [91].

Convergence properties such as uniform consistency and sample complexity of  $k$ -means have been extensively studied [36]. A finite sample upper bound on the  $k$ -means (test) mean-squared error is known to be  $O\left(\frac{1}{\sqrt{N}}\right)$  [92, 49]. Also we note that, the space complexity of standard  $k$ -means is high, since there is a dependency on  $N$ , the number of training samples.

## 4.4 Quantizer designs for federated learning

In this section, we propose different aggregation schemes for learning a global quantizer model. The aggregation scheme is applied independent of the quantizer design algorithm used at the clients. First, we show our analytical results for two devices and two quantization levels, considering IID signal source across edge devices. Later we generalize the model for non-identical but independent signal sources with several client devices and several levels (see [Appendix C.1](#)).

### 4.4.1 Proportional weighted aggregation

In the proportional weighted aggregation, the quantizer levels communicated by the client devices are fused using a weighted average. The weight is chosen proportional to the size of data on which it was trained. Assuming two devices with training sizes  $N_1$  and  $N_2$  respectively, the weight parameters are  $\alpha = \frac{N_1}{N_1+N_2}$  and  $\bar{\alpha} = 1 - \alpha = \frac{N_2}{N_1+N_2}$ . For applying this aggregation the client device needs to communicate the training size, which will lead to a communication cost of  $K + 1$  floating point (32 bit) numbers. Next, we show that the proportional weighted scheme,  $\pi_{\text{prop}}$  guarantees a (test) mean-squared error which is less than the (proportional weighted) average of the mean-squared errors of the quantizers at the client devices.

#### FL with IID datasets across client devices

Let two client devices – Device-1 and Device-2, be trained on IID datasets  $\mathcal{D}_1$  and  $\mathcal{D}_2$  to obtain the quantizers  $[q_{1,1}, q_{1,2}]$ , and  $[q_{2,1}, q_{2,2}]$  respectively. Also, let the test mean squared errors of these devices be denoted as  $\text{MSE}_1$  and  $\text{MSE}_2$ .

**Theorem 4.3.** *Considering IID datasets, the proportional weighted federated scheme has a test mean squared error,  $\text{MSE}_F(\pi_{\text{prop}})$ , which is less than the weighted average MSE of the individual devices. That is,*

$$\text{MSE}_F(\pi_{\text{prop}}) \leq \frac{N_1}{N_1 + N_2} \text{MSE}_1 + \frac{N_2}{N_1 + N_2} \text{MSE}_2, \quad (4.1)$$

where  $N_1 = \text{card}(\mathcal{D}_1)$  and  $N_2 = \text{card}(\mathcal{D}_2)$ .

*Proof.* Since the datasets  $\mathcal{D}_1$  and  $\mathcal{D}_2$  are IID, the test mean squared errors are computed with respect to a common source density, denoted as  $f(x)$ . Without loss of generality, let  $f(x)$  be in the bounded support  $[0, 1]$ . The proportionally weighted federated aggregation scheme will have the following quantization levels,

$$q_{F,1} := \alpha q_{1,1} + \bar{\alpha} q_{2,1},$$

$$q_{F,2} := \alpha q_{1,2} + \bar{\alpha} q_{2,2},$$

where  $\alpha = \frac{N_1}{N_1+N_2}$  and  $\bar{\alpha} = 1 - \alpha$ . Then, the test mean squared error at the devices,

$$\begin{aligned} \text{MSE}_d &= \mathbb{E} [(Q_d(X) - X)^2], \quad \text{for } d \in \{1, 2\} \\ &= \int_0^{t_d} (q_{d,1} - x)^2 f(x) dx + \int_{t_d}^1 (q_{d,2} - x)^2 f(x) dx, \end{aligned}$$

where  $t_d = \frac{q_{d,1} + q_{d,2}}{2}$  represents the (nearest neighbor) decision boundary. Now the test mean squared error of the federated quantizer is:

$$\begin{aligned} \text{MSE}_F(\pi_{\text{prop}}) &= \mathbb{E} [(Q_F(X) - X)^2] \\ &= \int_0^{t_F} (q_{F,1} - x)^2 f(x) dx + \int_{t_F}^1 (q_{F,2} - x)^2 f(x) dx \end{aligned}$$

where  $t_F = \frac{q_{F,1} + q_{F,2}}{2} = \alpha t_1 + (1 - \alpha)t_2$ . The desired result to be proved here, will follow from the fact that test MSE function is convex. To see this fact, we consider a generic test MSE function of a two level quantizer given by,

$$\text{MSE}(q_1, q_2, t) := \int_0^t (q_1 - x)^2 f(x) dx + \int_t^1 (q_2 - x)^2 f(x) dx,$$

where  $q_1 < q_2$  and  $t \in (0, 1)$ . Since  $(q_1 - x)^2$  and  $(q_2 - x)^2$  are convex functions in  $q_1$  and  $q_2$ , the function  $\text{MSE}(\cdot, \cdot, t)$  is convex with respect to the both  $q_1$  and  $q_2$ . Next, we check the condition under which the MSE function is convex w.r.t the variable  $t$ . The first and second derivative of  $\text{MSE}(q_1, q_2, t)$ . These are,

$$\frac{\partial \text{MSE}(q_1, q_2, t)}{\partial t} = (q_1 - t)^2 f(t) - (q_2 - t)^2 f(t) \quad (4.2)$$

$$\frac{\partial^2 \text{MSE}(q_1, q_2, t)}{\partial t^2} = 2(q_2 - q_1) \left\{ f(t) - \left( \frac{q_1 + q_2}{2} - t \right) f'(t) \right\} \quad (4.3)$$

The above relations show that corresponding to  $t = \frac{q_1 + q_2}{2}$ , the first derivative is zero and the second derivative is positive. This implies that the function  $\text{MSE}(q_1, q_2, \frac{q_1 + q_2}{2})$  is convex in both  $q_1$  and  $q_2$ . Thus the federated quantizer test error,

$$\begin{aligned} \text{MSE}_F(\pi_{\text{prop}}) &= \text{MSE}([\alpha q_{1,1} + \bar{\alpha} q_{2,1}], [\alpha q_{1,2} + \bar{\alpha} q_{2,2}], t_F) \\ &\leq \alpha \text{MSE}(q_{1,1}, q_{1,2}, t_1) + \bar{\alpha} \text{MSE}(q_{2,1}, q_{2,2}, t_2) \\ &= \alpha \text{MSE}_1 + \bar{\alpha} \text{MSE}_2 \end{aligned} \quad (4.4)$$

Thus the required inequality is proved.  $\square$

### FL with non-identical but independent datasets across client devices

Now we consider the two client devices – Device-1 and Device-2, trained on non-identical but independent datasets  $\mathcal{D}'_1$  and  $\mathcal{D}'_2$  to obtain the quantizers  $[q'_{1,1}, q'_{1,2}]$ , and  $[q'_{2,1}, q'_{2,2}]$  respectively. Due to the statistical independence between the datasets, the pooled dataset  $\mathcal{D}'_1 \cup \mathcal{D}'_2$  has a mixture distribution,

$$g(x) = \alpha' f_1(x) + (1 - \alpha') f_2(x), \quad (4.5)$$

parameterized by  $\alpha'$  and  $f_i(x)$ ;  $i \in \{1, 2\}$  are the respective PDFs. In this case, let the test mean squared errors of these devices be denoted as  $MSE'_1$  and  $MSE'_2$ .

**Theorem 4.4.** *Considering non-identical but independent datasets, the proportional weighted federated scheme has a test mean squared error,  $MSE_F(\pi_{prop})$ , which is less than the weighted average MSE of the individual devices. That is,*

$$MSE_F(\pi_{prop}) \leq \alpha' MSE'_1 + (1 - \alpha') MSE'_2. \quad (4.6)$$

*Proof.* By denoting  $\alpha := \frac{N_1}{N_1 + N_2}$ , the federated quantizer using  $\pi_{prop}$  has quantization levels,  $q_{F,1} := \alpha q'_{1,1} + \bar{\alpha} q'_{2,1}$  and  $q_{F,2} := \alpha q'_{1,2} + \bar{\alpha} q'_{2,2}$ , where  $\bar{\alpha} = 1 - \alpha$ . Thus, the test mean squared error of the resultant quantizer  $\{q_{F,1}, q_{F,2}\}$  can be computed using the definition,  $MSE_F(\pi_{prop}) = \mathbb{E}[(Q_F(X) - X)^2]$ . That is,

$$MSE_F(\pi_{prop}) = \int_0^{t_F} (q_{F,1} - x)^2 g(x) dx + \int_{t_F}^1 (q_{F,2} - x)^2 g(x) dx \quad (4.7)$$

where  $t_F := \frac{q_{F,1} + q_{F,2}}{2}$ . Further, upon expanding the terms,

$$\begin{aligned} MSE_F(\pi_{prop}) &= \int_0^{t_F} (\alpha q'_{1,1} + \bar{\alpha} q'_{2,1} - x)^2 (\alpha' f_1(x) + \bar{\alpha}' f_2(x)) dx \\ &\quad + \int_{t_F}^1 (\alpha q'_{1,2} + \bar{\alpha} q'_{2,2} - x)^2 (\alpha' f_1(x) + \bar{\alpha}' f_2(x)) dx, \end{aligned} \quad (4.8)$$

where  $\bar{\alpha}' = 1 - \alpha'$ . Relying on the convexity of the MSE function, as proved in (4.2)-(4.3), we can rewrite the above equation as,

$$\begin{aligned} MSE_F(\pi_{prop}) &\leq \alpha \int_0^{t_1} (q'_{1,1} - x)^2 (\alpha' f_1(x) + \bar{\alpha}' f_2(x)) dx \\ &\quad + \alpha \int_{t_1}^1 (q'_{1,2} - x)^2 (\alpha' f_1(x) + \bar{\alpha}' f_2(x)) dx \\ &\quad + \bar{\alpha} \int_0^{t_2} (q'_{2,1} - x)^2 (\alpha' f_1(x) + \bar{\alpha}' f_2(x)) dx \\ &\quad + \bar{\alpha} \int_0^{t_1} (q'_{1,1} - x)^2 (\alpha' f_1(x) + \bar{\alpha}' f_2(x)) dx. \end{aligned} \quad (4.9)$$

In the right hand side of the above inequality  $t_d$  for  $d \in \{1, 2\}$  represents the quantizer boundary defined by,  $t_d := \frac{q_{d,1} + q_{d,2}}{2}$ . Now, by regrouping the terms we can express the inequality in (4.9) as,

$$\text{MSE}_F(\pi_{\text{prop}}) \leq \alpha' \alpha \text{MSE}_{1,1} + \bar{\alpha}' \alpha \text{MSE}_{1,2} + \alpha' \bar{\alpha} \text{MSE}_{2,1} + \bar{\alpha}' \bar{\alpha} \text{MSE}_{2,2}, \quad (4.10)$$

where  $\text{MSE}_{i,j} := \int_0^{t_j} (q'_{i,1} - x)^2 f_j(x) dx + \int_{t_j}^1 (q'_{i,2} - x)^2 f_j(x) dx$ . Further, on simplification,

$$\begin{aligned} \text{MSE}_F(\pi_{\text{prop}}) &\leq \alpha' [\alpha \text{MSE}_{1,1} + \bar{\alpha} \text{MSE}_{2,1}] + \bar{\alpha}' [\alpha \text{MSE}_{1,2} + \bar{\alpha} \text{MSE}_{2,2}], \\ &\leq \alpha' \max \{\text{MSE}_{1,1}, \text{MSE}_{2,1}\} + \bar{\alpha}' \max \{\text{MSE}_{1,2}, \text{MSE}_{2,2}\}. \end{aligned} \quad (4.11)$$

Finally, we can minimize the upperbound on the MSE by joint minimization with respect to the quantization levels  $\{q'_{1,1}, q'_{1,2}, q'_{2,1}, q'_{2,2}\}$ . That is,

$$\begin{aligned} \text{MSE}_F(\pi_{\text{prop}}) &\leq \min_{q'_{1,1}, q'_{1,2}, q'_{2,1}, q'_{2,2}} [\alpha' \max \{\text{MSE}_{1,1}, \text{MSE}_{2,1}\} + \bar{\alpha}' \max \{\text{MSE}_{1,2}, \text{MSE}_{2,2}\}] \\ &\leq \alpha' \text{MSE}'_1 + \bar{\alpha}' \text{MSE}'_2. \end{aligned} \quad (4.12)$$

The inequality in (4.12), is obtained by distributing the minimization on individual terms of the summation.  $\square$

*Remark:* In typical datasets the parameter  $\alpha'$  of the mixture distribution will be the fraction of the data samples, i.e.  $\alpha' = \alpha = \frac{N_1}{N_1 + N_2}$ . The generic result for performance improvement of FL aggregation with respect to the mean squared error, considering  $D$  edge devices and  $K$  quantizer levels, is discussed in [Appendix C.2](#).

#### 4.4.2 Number based weighted aggregation

The number based federated aggregation (denoted as  $\pi_{\text{num}}$ ) is an improvement over the proportional weighted approach. This method allows the central server to select two separate parameters,  $\alpha_1$  and  $\alpha_2$ , unlike the one parameter  $\alpha$  in the previous case. For the two devices and two levels example, the federated scheme will have the quantization levels,

$$q_{F1} := \alpha_1 q_{1,1} + \bar{\alpha}_1 q_{1,2} \quad \text{and} \quad q_{F2} := \alpha_2 q_{2,1} + \bar{\alpha}_2 q_{2,2}.$$

Using the convexity of the federated MSE, shown in Theorem. 4.3, we see that the number based weighted average aggregation results in a lesser MSE than the weighted average of

the client MSEs. Because there are two parameters that can be independently chosen in this model, the scheme should perform better than the proportional weighted scheme. This is formally stated in the result below.

**Corollary 4.5.** *The number based weighted federated scheme has a test mean squared error,  $MSE_F(\pi_{num})$ , which is less than the weighted average MSE of the individual devices. That is,*

$$MSE_F(\pi_{num}) \leq \delta MSE_1 + (1 - \delta) MSE_2,$$

where  $\delta = \alpha_1 \gamma + \alpha_2 (1 - \gamma)$ , and  $\gamma = \frac{q_{1,1} - q_{2,1}}{q_{1,1} - q_{2,1} + q_{1,2} - q_{2,2}}$ .

By choosing an appropriate  $\alpha_1$  and  $\alpha_2$ , we can set  $\delta$  to  $\frac{N_1}{N_1 + N_2}$ . However, the question that we ask is – whether a better choice is possible? One natural choice to pick is the relative share of data points for each quantization level. That is,

$$\alpha_1 = \frac{n_{1,1}}{n_{1,1} + n_{2,1}} \quad \text{and} \quad \alpha_2 = \frac{n_{1,2}}{n_{1,2} + n_{2,2}},$$

where  $\{n_{d,j}; (d, j) \in \{1, 2\}^2\}$  denotes the number of data samples of Device- $d$  mapped to  $q_{d,j}$ . Thus, in this scheme the clients have to send a number vector  $\vec{n} = [n_{d,1}, n_{d,2}]$  in addition to the quantizer levels  $\vec{q} = [q_{d,1}, q_{d,2}]$ . The number based approach is observed to have good generalization over test data in the simulations we have conducted. However, this gain is achieved by communicating  $2K$  floating point coefficients, consisting of the quantization levels and the associated number of points, which is almost twice that was communicated under  $\pi_{prop}$ .

### 4.4.3 Probability score weighted aggregation

Probability score based aggregation (denotes as  $\pi_{ps}$ ) stems from the LALM quantization scheme (Section 4.3.3), where each client device estimates an approximate probability density function. For computing the probability score, we estimate the area under the piecewise linear region within the boundary of a quantization level (which turns out to be sum of trapezoidal regions). If  $p_k$  represents the probability score, and  $\hat{\mathbb{P}}(\cdot)$  the probability estimate,  $p_k := \hat{\mathbb{P}}(X \in I_k)$ , where  $I_k$  is the interval that maps to the level  $q_k$ . This scheme is akin to the number based scheme when the client devices have approximately the same

number of data samples. For two devices and two levels, the federated quantizer based on probability score aggregation has levels given by,

$$q_{F1} := \frac{p_{1,1}q_{1,1} + p_{2,1}q_{2,1}}{p_{1,1} + p_{2,1}} \text{ and } q_{F2} := \frac{p_{1,2}q_{1,2} + p_{2,2}q_{2,2}}{p_{1,2} + p_{2,2}}.$$

The notation  $p_{d,k} := \widehat{\mathbb{P}}(X \in I_{d,k})$ , where the interval  $I_{d,k}$  is associated with the level  $q_{d,k}$ .

### Probability score to achieve asynchronous updates

In the federated architecture, the client devices may report their updates at different times. Because of this, the global model has to be updated from the previous state by incorporating the most recent updates from the clients. Since the aggregation schemes use weighted average over the quantization levels, it is required to estimate the number of data points associated with each quantization level in the global model, which is otherwise unknown. Here we propose an algorithm that estimates the number data point (see Algorithm 4), using insights from the LALM quantizer design. From the number of

---

#### Algorithm 4 Number Estimation for the Global Quantizer

---

**Input:** quantization levels  $\vec{q}_d$ ; training samples  $N_d$ ; probability scores  $\{p_{d,k} : k \in \{1, \dots, K\}\}$  for  $d \in \{1, 2, \dots, D\}$

**Initialize:** global (federated) quantizer,  $\vec{q}_F = [q_{F,1}, \dots, q_{F,K}]$  based on  $\pi_{\text{num}}$  or  $\pi_{\text{ps}}$ ; set reference levels  $q_{F,0} = -\infty$  &  $q_{F,K+1} = \infty$

**for**  $k \in \{1, 2, \dots, K\}$  **do**

**for**  $d \in \{1, 2, \dots, D\}$  **do**

- Area under piecewise linear density estimate for **Device-d** in the range  $[q_{k-1}, q_{k+1}]$  (denoted as  $A_{d,k}$ ) is obtained using  $p_{d,k}$
- Data sample contribution from **Device-d** is determined using  $n_{d,k} \leftarrow A_{d,k} \times N_d$

**end for**

Calculate total contribution,  $n_{F,k} \leftarrow \sum_d n_{d,k}$

**end for**

**Return**  $\vec{n}_F = [n_{F,1}, n_{F,2}, \dots, n_{F,K}]$

---

points estimated, the central server can use any of the four federated aggregation schemes discussed in this section to determine the updated global quantizer.

#### 4.4.4 $k$ -means based aggregation

The three federated aggregation schemes mentioned before applies when there is a clustering around respective quantization levels obtained from different devices. In other words, it works if the cluster of levels  $\{q_{d,1} : d \in \{1, 2, \dots, D\}\}$  is separated sufficiently from other clusters  $\{q_{d,2}\}, \{q_{d,3}\}, \dots, \{q_{d,K}\}$  for  $d \in \{1, 2, \dots, D\}$ . However, this is not the case, when the dynamic ranges of the training data are heterogeneous. In such situations, the aggregation scheme should be able to handle inter-mixing of quantization levels from the different clients. For instance, considering two devices and two levels the generalized aggregation will result in the federated quantization levels,

$$\begin{aligned} q_{F1} &:= \alpha_1 q_{1,1} + \alpha_2 q_{1,2} + \alpha_3 q_{2,1} + \alpha_4 q_{2,2} \\ q_{F2} &:= \beta_1 q_{1,1} + \beta_2 q_{1,2} + \beta_3 q_{2,1} + \beta_4 q_{2,2}, \end{aligned}$$

where  $\alpha_i, \beta_i \geq 0$ ;  $i \in \{1, \dots, 4\}$ , and  $\sum_i \alpha_i = \sum_i \beta_i = 1$ .

A special case of such a generalized aggregation is the (central)  $k$ -means algorithm, which has the additional conditions that  $\alpha_i \beta_i = 0$  and  $\alpha_i + \beta_i > 0$ . That is, the  $k$ -means aggregation picks each quantization level and assigns it to either the cluster of  $q_{F1}$  or the cluster of  $q_{F2}$ . Since the  $k$ -means quantizer is known to be the best algorithm at the client devices, its combination with  $k$ -means aggregation is found to perform well under the federated architecture. This scheme is termed as  $k$ - $k$ -means quantizer design.

## 4.5 Simulations and experiments

To study the benefits of the distributed quantizer design using federated learning, we carried out experiments on three datasets. In accordance with the analytical results in Theorem. 4.3 and Theorem. 4.4, we show performance gains with the federated design on heterogeneous distributions, which have different means and dynamic ranges.

**Notations:** In this section we name each federated learning scheme in the format  $\langle \text{Device Quantizer Scheme} \rangle - \langle \text{Aggregation Scheme} \rangle$ . For example, if equispaced scheme is used at the devices and  $k$ -means is used at the server, it will be denoted as ES- $\pi_{kM}$ . Similarly, for a scheme with LALM at the client devices and  $\pi_{\text{num}}$  at the server, the notation LALM- $\pi_{\text{num}}$  will be used.

### 4.5.1 Simulations on synthetic data

We synthesized independent data samples from the  $\text{Beta}(a,b)$  distribution for various values of  $a$  and  $b$  (greater than 1), and then used this data to train the federated model while considering the heterogeneity of the system. We consider two specific cases, first where clients have different sample sizes (**HetSamp**), and the second where clients have different probability distributions (**HetDist**).

Under the **HetSamp** model, we considered 100 devices, with data distribution  $\text{Beta}(4,2)$  and having data sizes of either 50 or 100 chosen with equal probability. The MSE performance of the four different federated number based schemes were analyzed (see Figure. 4.2), by 100 Monte-Carlo evaluations. It is observed that, the MSE of the equispaced quantizer is at a constant gap separation from the MSE of a baseline quantizer, which has access to all the training data from all clients. The remaining three quantizer design schemes are seen to be achieve the baseline performance in some regions, while deviating at other regions. For instance, the LVQ is optimal for low values of  $K$ , and diverges as  $K$  increases. This is because the LVQ learning rate,  $\gamma_n$  was chosen as  $\frac{1}{\lceil n/200 \rceil}$ , which does not modify with  $K$ . The deviation in the other two plots are attributed to the model inaccuracies arising due to less number of data samples in the considered quantization interval.

For the **HetDist** model, we considered 10 devices, each having 1000 data samples, which are drawn from either  $\text{Beta}(4,2)$  or  $\text{Beta}(2,2)$  distributions with equal probability. The equispaced, LVQ and  $k$ -means based federated schemes are seen to be touching the baseline performance. On the other hand LALM deviates for large values of  $K$ , due to estimation errors resulting out of less number of data samples.

To study the effect of number of devices in the federated learning of the quantizers, we consider the experiment where each client has  $n = 10$  data samples, drawn from the  $\text{Beta}(4,2)$  distribution. First we consider the case with only one device, i.e.  $D = 1$ , and analyze the MSE performance using classical LVQ and  $k$ -means quantizers. Next we repeat the experiment for  $D = 10$  and  $D = 100$ , while maintaining the number of data samples per device as the same. Through this, we bring out the fact that federated learning improves as we scale the number of devices in a homogeneous setting (i.e. IID

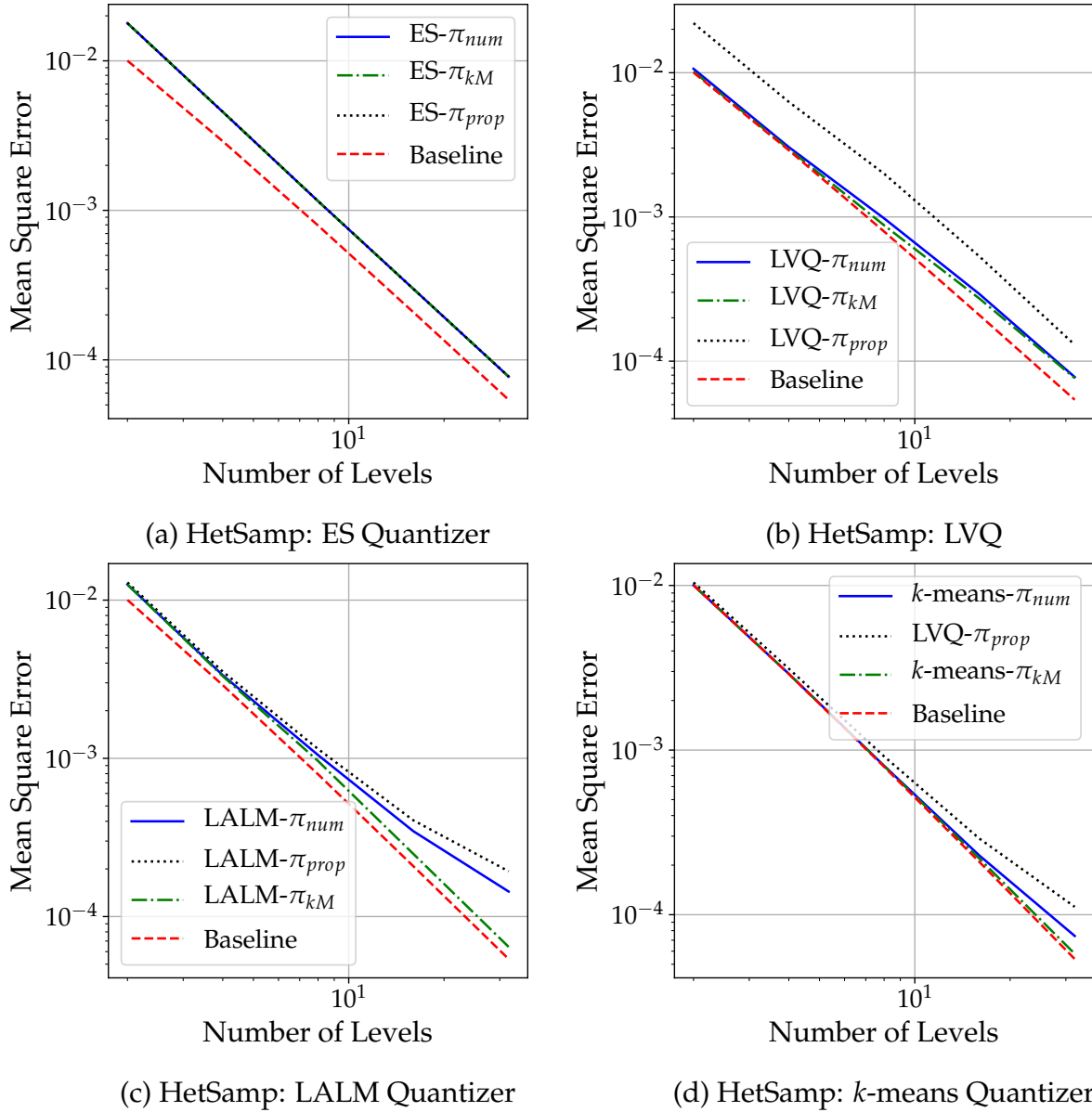


Figure 4.2: A comparison of different federated quantizers design schemes with 100 devices, data samples from Beta(4,2) distribution, and sample sizes either 50 or 100 with equal probability.

data and same number of samples across devices). The results are tabulated in Table 4.3, for two different sample sizes per device,  $n = 10$  and  $n = 50$ .

#### 4.5.2 Simulations on fitness dataset

We analyzed the performance of the federated quantizers using the heart-rate data obtained from an available Fitbit dataset [52], which has the data recorded from 14 devices

Data per device	Quantizer	Federated aggregation $\pi_{\text{kM}}$		
		$D = 1$	$D = 10$	$D = 100$
$n = 10$	LVQ	0.00185	0.00127	0.00094
	$k$ -means	0.00109	0.00084	0.00081
$n = 50$	LVQ	0.00161	0.00109	0.00087
	$k$ -means	0.00097	0.00082	0.00080

Table 4.3: This table illustrates the improvement of the MSE performance of the federated schemes when number of devices increase. Two sample sizes have been considered, and the  $k$ -means based aggregation is used at the central server.

(users) for a duration of two months. The dataset has 1.15 million data points in the first month and 2.48 million in the second month. Data samples are available for every 5 seconds when the device was in used, and each data sample is an integer type. The minimum and maximum values measured are 36 and 180, and the mean and standard deviation are 79.76 and 18.73 respectively. The details of the sample distribution across devices are summarized in Table. 4.4. The experiment was carried on a Linux machine, with the hardware specs – Processor : Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz, RAM : 32 GB. Cores : 10; and Python (ver 3.6) programming using NumPy (ver 1.15.4) and Pandas (ver 0.19.2) libraries.

**Communicated bits and MSE tradeoff :** The tradeoff between the number of bits communicated for learning the federated quantizer and the mean-squared error of this quantizer is characterized in Figure. 4.3. In this experiment, all devices have used the same quantizer algorithm and the same number of levels. To achieve a target MSE distortion, the devices need to communicate as many bits as depicted in the representative plot (dashed lines). For instance, to achieve a required MSE of approximately 3.5, each device has to send 576 bits if using the ES quantizer at the devices and the  $k$ -means aggregation at the server. In addition, this tradeoff curve also lets the devices to chose an

Data Period	Number of samples at clients			
	Min	Max	Average	Std dev
Month 1	439	283794	82477	66261
Month 2	2490	285461	177404	83109

Table 4.4: A statistical description of the number of data samples across 14 users in the fitness dataset.

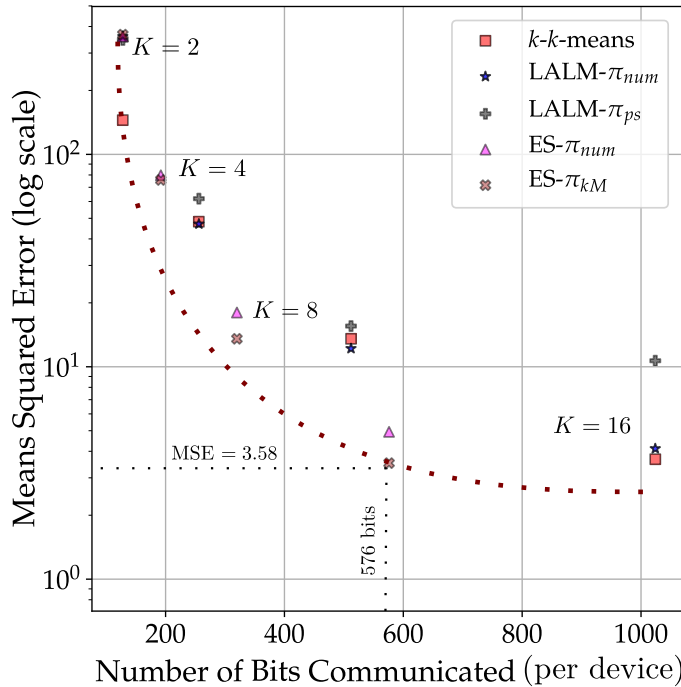


Figure 4.3: The figure illustrates the communication cost vs MSE tradeoff in quantizer design using federated learning. The plot will aid the client devices in picking an appropriate scheme considering the communication and energy constraints.

appropriate quantizer design scheme considering the communication and energy (battery) constraint. From Figure. 4.3 we infer that, even a naive quantization scheme such as the equispaced, works at the optimal boundary of the tradeoff curve, when the federated quantizer uses  $k$ -means aggregation.

**Training size and MSE tradeoff:** In classical quantizer design, we expect the quantizer MSE to decrease with increase in training samples. However, in the federated

Algorithms	Bytes Communicated	Training- 50% Days		Training- 60% Days		Training- 70% Days		Training- 80% Days	
		Train	Test	Train	Test	Train	Test	Train	Test
ES-num	40	16.43	18.01	16.16	16.87	16.88	18.33	21.63	21.32
ES- $k$ -Means	40	12.65	12.47	12.02	13.50	12.98	13.37	14.63	14.33
$k$ - $k$ -means	64	12.29	13.54	12.00	13.52	12.01	14.05	12.54	13.74
LALM-num	64	11.41	12.18	11.23	12.50	11.29	12.65	12.07	12.14
LALM-pscore	72	14.21	15.54	11.33	12.93	13.95	16.62	13.69	15.42
LALM- $k$ -means	64	12.18	12.83	11.62	12.76	11.63	12.98	12.18	12.54
<b>Baseline</b>	$\sim$ 4.6 Million	<b>11.26</b>	<b>12.15</b>	<b>11.07</b>	<b>12.41</b>	<b>11.08</b>	<b>12.61</b>	<b>11.81</b>	<b>11.97</b>

Table 4.5: We compare the mean-squared error of different federated quantization schemes measured with respect to various train and test sizes, filtered using number of training days. Number of bytes communicated under each scheme is also compared.

learning design, this does not hold, since the data is non-IID and data samples have heterogeneous characteristics such as mean and dynamic range. This fact is validated in Table 4.5, using means-squared error performance on different federated schemes. In this experiment, the training data size was varied, by filtering the data corresponding to 50%, 60%, 70% and 80% of the training days (which is 60 days in this dataset). Federated schemes such as  $k$ - $k$ -means, LALM-num and LALM- $k$ -means show consistent behavior, in the initial three training cases. However, an increasing trend in MSE is observed for the equispaced quantizer schemes for the training on 80% days, since a training bias is introduced by a set of devices having excessive training data. We expect that this training procedure bias, can be avoided by considering a suitable regularizer that accounts for user fairness [86], which will be considered in a future work.

### 4.5.3 Simulations on MNIST dataset

**Dataset & Hardware Specification:** MNIST is a standard dataset used in machine learning [93], which consists of 60000 images of size  $28 \times 28$  for training set and 10000 images for testing. Simulations were done using “Python3 Google Compute Engine backend” with RAM: 0.77GB/12.72GB and Disk:28.34GB/107.77GB

For conducting experiments on vector quantizer (clustering) design using federated learning model on MNIST dataset, we have considered 10 devices each having 6000 training images, and applied the  $k$ - $k$ -means quantization scheme. At the client devices  $k$ -means clustering was performed (using `MiniBatchKMeans` routine available in `sklearn` library, with 100 iterations and a randomly picked initialization), and the resulting cluster centers ( $28 \times 28$  images) were communicated along with the cluster labels learned from the locally available training data. At the server,  $k$ -means based federated aggregation was performed with similar iteration and initialization parameters used at the devices. The performance of  $k$ - $k$ -means scheme is analyzed using label accuracy and mean-squared error on the test dataset, by considering quantized client-server communication with  $b$  bits per image pixels (corresponding to cluster centers). In Figure. 4.4, we compare the results for binary (i.e.  $b = 1$ ) equispaced quantized communication scheme with the full precision (i.e.  $b = 8$ ) scheme. We observe that, even though the test MSE increases considerably for binary communication, the test accuracy only drops within a margin of 5%. In addition,

we note that the fluctuations in the accuracy plot, arise possibly due to the stochastic nature of `MinibatchKMeans` routine.

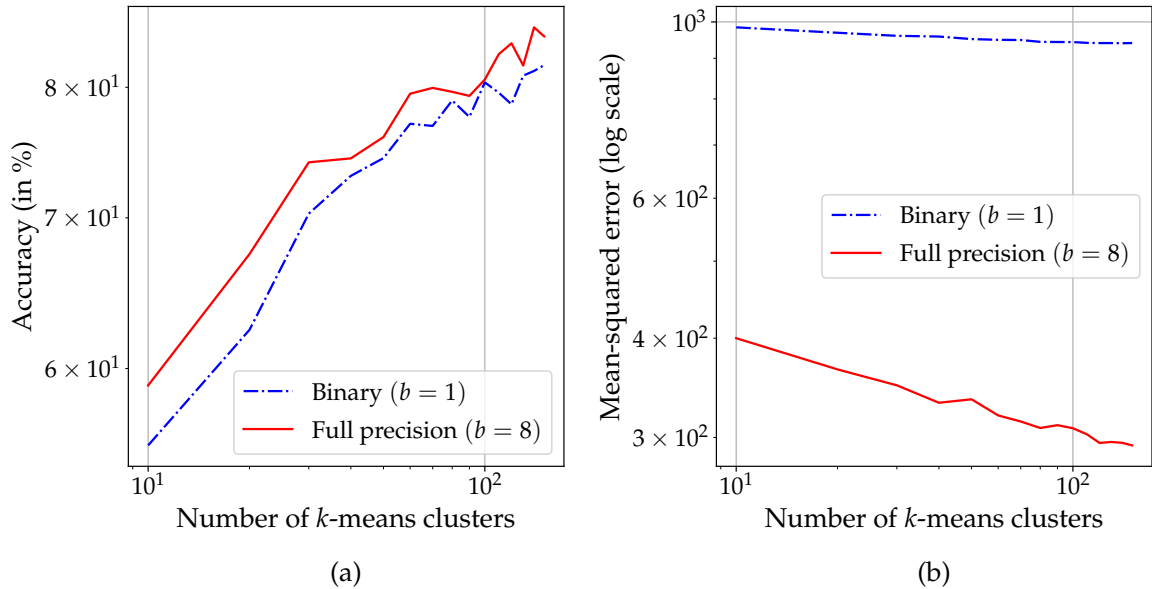


Figure 4.4:  $k$ - $k$ -means clustering performance on MNIST data when  $b$  bits per pixel quantized communication is used. Plot (a) shows the classification accuracy and plot (b) shows the test MSE.

Next we verify the analytical results shown in Theorem 4.3 and Theorem 4.4 using the MNIST dataset. Corresponding to 10 distributed client devices, each having 6000 training images, we perform the federated learning based  $k$ - $k$ -means quantizer design. As the training data is shuffled the dataset available at each device is non-identical. In Fig. 4.5 (a), we observe that the federated scheme improves the classification accuracy by approximately 10% for every choice of cluster size,  $k$ . The efficacy of  $k$ - $k$  means is also visible in the MSE performance shown in Fig. 4.5 (b), where the decay of MSE for federated learning scheme is faster compared to the  $k$ -means implemented at individual devices. In these plots, we note that the number of communicated bytes is proportional to the number of  $k$ -means clusters as each cluster is composed of 784 pixels and each pixel is 1-byte. In addition, the  $k$ - $k$ -means classification task preserves the implicit privacy at each edge device as only the quantized representation is communicated.

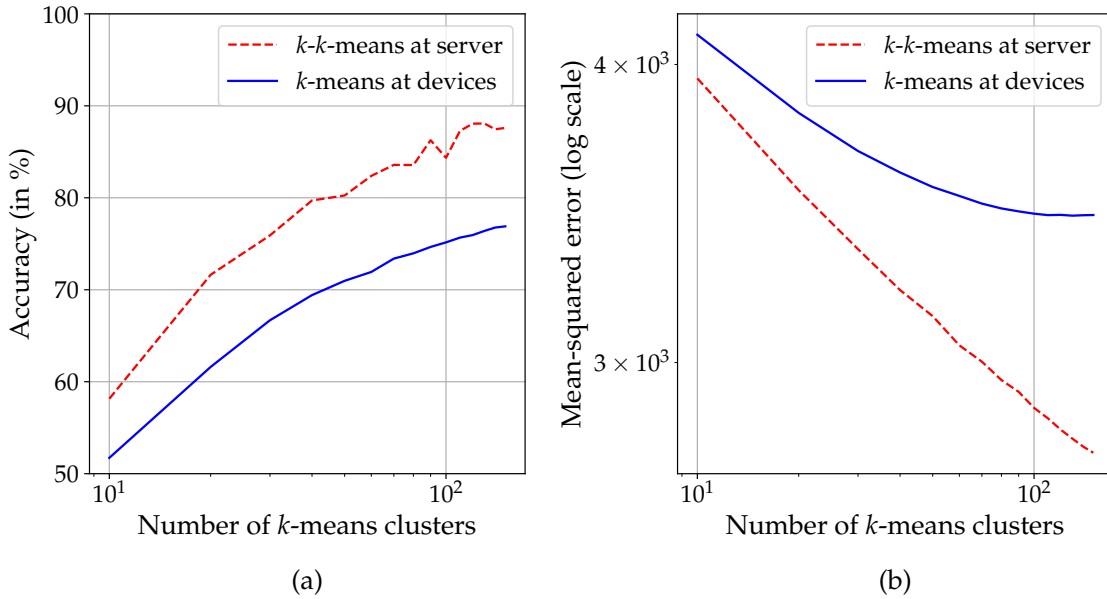


Figure 4.5:  $k$ - $k$ -means clustering performance as compared to the  $k$ -means clustering implemented at individual edge devices. Simulations use the MNIST image classification dataset. In (a), the classification accuracy of the federated learning is shown; and (b) shows the MSE performance.

## 4.6 Summary

In this chapter, we introduced a novel framework for the design of quantizers using the federated learning architecture, where client devices only communicate their learned quantizer instead of their training data. Using candidate quantization algorithms already known in literature, we proposed quantizer designs that allow linear aggregation of quantization levels, and analyzed their performance with respect to non-identical but independent data using the mean-squared error distortion. Our experimental results were shown on both synthetic and real-world datasets, which incorporates the model heterogeneity inherent in federated learning.

## Chapter 5

# Algorithms for Overpredictive Signal Analytics in Federated Learning

In this chapter, we build on the FL architecture developed in Chapter. 4 by considering the overprediction constraint motivated in the context of envelope quantizer design (Chapter. 2). Using FL, clients (IoT devices) having many signal samples can aid a data center (a third-party server) learn the global signal model by pooling these distributed samples. The clients may have privacy concerns, and the pooling of distributed samples will require accounting for the communication cost involved. As a result, a processed approximation of these samples may be desirable in the FL paradigm. This decentralized learning approach is termed *distributed signal analytics* in this work. Overpredictive signal approximations maybe desired to perform such distributed signal analytics, which is primarily motivated by network demand (capacity) estimation and planning applications.

In the FL framework, we propose algorithms that calculate an overpredictive signal approximation at the client devices using an efficient convex optimization framework. The tradeoff between the number of bits communicated by clients to the server and the signal approximation error is quantified. An experimental analysis of our signal approximations is presented on an available residential energy consumption dataset, based on signal analytics using the estimated cumulative distribution function of the signal sources.

## 5.1 Background

In specific networked system applications, such as electricity network demand planning, flood/drought forecasting systems or Unmanned Aerial Vehicle (UAV) path planning, or autonomous driving, it is essential to have an overpredictive estimate of signals. For instance, consider a smart city application shown in Fig. 5.1, where each household has a smart energy meter to monitor and record the instantaneous electric power. At the end of the day, each smart meter will need to report an approximate summary of consumer energy usage to a centrally located server, accessible to the electricity network planner. The planner desires to have an overpredictive estimate of the energy demand time-series so that the generation can be planned to meet the consumer demand consistently.

In such a smart city application, we attempt to answer the question, *how to perform overpredictive signal analytics when the devices are distributed?*. The significant challenges in doing such distributed signal analytics in FL are identified to be: (1) asymmetric communication resources (i.e., the uplink rate is less than the downlink rate), (2) heterogeneous devices, (3) sensitive private data, and (4) scale of operation (a large number of devices) [1]. This work will primarily address communication rate constraints while implicitly ensuring user privacy and scalability. In this first exposition, we restrict the focus to the class of homogeneous user devices. FL allows a common global model to be learned from distributed devices by using efficient compressed signal representations. The existing literature in FL, however, has a focus on learning a parametric model using a central parametric server, such as a neural network model through the successive exchange of gradients [94, 39, 1, 87, 85, 42]. In this work, the approach is different as we utilize the FL architecture for learning signal analytics [95], instead of a parametric model. The goal is to achieve efficient statistical representation for performing analytics involving signal overprediction.

A key feature in FL is to utilize edge signal processing to perform distributed signal analytics. Let us revisit the example of energy demand time-series aggregation performed at the smart city electricity server, as illustrated in Fig. 5.1. This central server performs network planning to meet the electricity demands of the consumers in the city. At each of the  $D$  consumers (also known as edge/client devices), the installed smart meter would

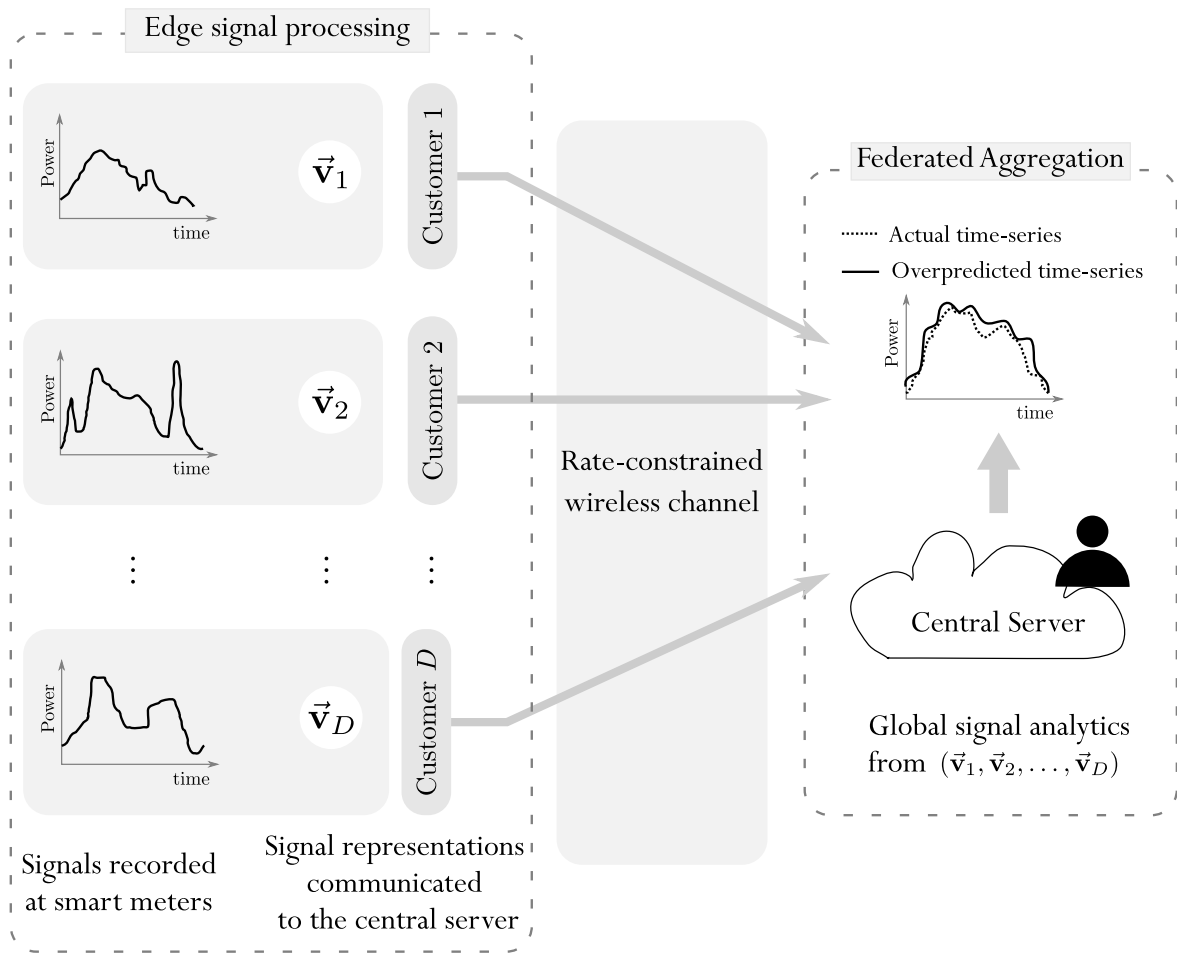


Figure 5.1: The figure illustrates federated learning for overpredictive signal analytics in a smart city with  $D$  electricity consumers. The power consumption time-series recorded at the consumer’s smart meters are converted to compressed signal representations,  $\{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_D\}$  using edge signal processing. These signal representations are communicated to the central server for computing signal analytics that overpredict the actual demand time-series.

record the instantaneous power consumed and communicate a compressed representation, *viz.*  $\{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_D\}$ , of the demand signal to the server through a rate-constrained wireless channel. The network planner aggregates the client signal representations to estimate an approximate demand time-series signal, which over-predicts the actual aggregate demand signal. A possible implementation scheme would be to compute approximate signal representations locally at the edge devices such that these representations satisfy the individual signal overprediction constraints. The edge devices then communicate these compressed signal representations to the server, where signal analytics of interest

at computed. The signal analytics desired at the server may include the peak electric power demand or the aggregate signal Cumulative Distribution Function (CDF). Such overpredicted demand signal analytics computed from the approximate signal representations  $\{\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2, \dots, \vec{\mathbf{v}}_D\}$ , will reduce the client-server communication cost. Since the actual data does not move to the server, implicit data privacy is achieved. The computed signal analytics at the server is useful to maintain the stability and security of the electricity distribution network.

The main contributions of this work are summarized below.

- We propose an overprediction-based signal approximation algorithm using the Fourier basis for signal representation in the federated learning setup.
- We propose a federated aggregation procedure to learn the global signal analytics at the central server, using the empirical CDF of the aggregate signal.
- We provide mathematical upperbounds on the pointwise difference of the actual signal CDF and the Glivenko-Cantelli CDF estimate of the overpredicted signal, considering the effect of signal sampling.
- We analyze the tradeoff between the communication cost and the signal approximation error and validate it using experiments on an off-the-shelf residential energy consumption dataset [64].

### 5.1.1 Prior literature

Using the federated learning (FL) approach, communication-efficient distributed learning was introduced by McMahan et al. [1]. The generic FL architecture proposed in the paper considers massively distributed, heterogeneous devices with limited communication. The authors illustrate the utility of FL models for training deep neural networks with on-device data using stochastic gradient descent (SGD) based *FedAvg* algorithm. An improved *FedProx* algorithm was later proposed to handle system heterogeneity in FL, which in addition has convergence guarantees [96]. In another related work, Suresh et al. [85] discusses a communication-efficient distributed mean estimation using FL, stochastic quantization, and structured rotation. Bonawitz et al. [87], has described the system-level implementation of FL algorithms in large-scale networks. Advances in FL algorithms, including open

directions in distributed learning with privacy-sensitive data, have been summarized in the review paper by Kairouz and McMahan [8]. From a signal processing perspective, the applications, and challenges of FL have been discussed by Li et al. [7]. Signal processing methods such as compressed gradients and sketched updates have found utility in training and deploying FL in low-power TinyML devices [97, 98]. Further, FL models to address the fairness of resource allocation have been studied from the perspective of parametrized cost functions and empirical probability distributions [99, 86]. The references above mostly deal with single task FL models. Considering real-world IoT systems, a multi-task learning scheme was proposed [100] that allows for a certain degree of model personalization.

Several applications of signal overprediction are discussed in the literature. These include watershed management (hydrology), Unmanned Aerial Vehicle (UAV) path planning, CPU power prediction, and database management for TV whitespace protection contours. In hydrology, flood and drought predictions are made possible through statistical learning approaches that use custom loss functions to include overprediction constraints [101]. Support Vector Machine (SVM) models utilizing asymmetric loss functions are generally used for CPU power cycle prediction [102]. Recent research shows that neural networks do not capture accurate depth information in images, which can have significant safety implications in autonomous driving [10]. Overpredicting ADC design is a suggested solution to overcome this safety challenge. Maheshwari and Kumar have proposed a novel quantizer design for the TV whitespace (geo-database management) application, which determines an overpredicting envelope [66]. Recent research analyzes a Federated learning (FL) approach considering personalized client attributes for energy demand prediction using a clustered aggregation scheme [103]. In another related work, Konstantinos et al. studies the time-series dimensionality reduction approach with symbolic aggregate approximation (SAX) and Lloyd's algorithm [104], using a combination of quantization and event-based sampling to achieve signal compression. Recently, Saputra et al. have examined the utility of FL algorithms for location-specific demand prediction in an electric vehicle (EV) charging architecture [105].

## 5.2 System model and background concepts

In this section we describe the signal model for overpredictive signal analytics, and provide the background concepts useful for the related mathematical analysis.

### 5.2.1 Distributed signal model

Consider a distributed client-server model consisting of  $D$  clients and a central server, as described by the federated learning architecture shown in Figure 5.2. Each client device records a continuous-time signal represented as  $f_1(t), f_2(t), \dots, f_D(t)$ . Without loss of generality we will assume that the recorded signals have a bounded support in the interval  $[0, 1]$ . For the ensuing analysis, we assume that these signals are  $p$ -times differentiable, where  $p \geq 1$ . The signals at the client devices are to be communicated to the central server to learn a global model. However, due to the rate constrained client-server communication channel, the devices only communicate an approximate signal, which will be denoted as  $\hat{f}_1(t), \hat{f}_2(t), \dots, \hat{f}_D(t)$ .

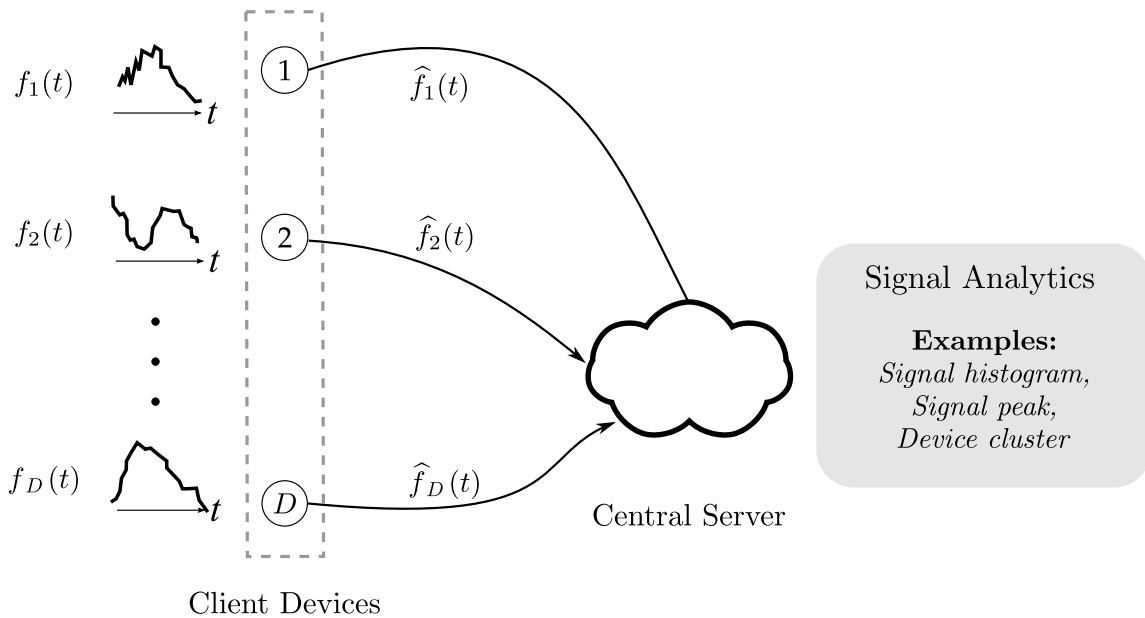


Figure 5.2: The federated learning system model for distributed signal analytics. In this model,  $D$  client devices communicate an approximation of the recorded signals, and the central server pools these signals to obtain useful signal analytics such as the histogram, aggregate signal or a clustering of the devices.

### 5.2.2 Fourier basis representation

In this first exposition of the study of distributed signal approximation in a federated learning model, will use the Fourier basis representation for the signals at the client devices. That is, each recorded signal at the clients can be represented by a Fourier series,

$$f_i(t) = \sum_{k=-\infty}^{\infty} a_i[k] \exp(j2\pi kt), \quad t \in [0, 1],$$

where  $i \in \{1, 2, \dots, D\}$ . It will be assumed that  $f_i(0) = f_i(1)$ , which is due to the periodicity requirement of the Fourier series representation. Since the signals at the clients are assumed to  $p$ -times differentiable, we observe a polynomial decay in the Fourier series coefficients of the signals, which is stated in the fact below.

**Fact 5.1** (Sec 2.3,[106]). *A signal  $f(t), t \in [0, 1]$ , with  $f(0) = f(1)$ , is  $p$ -times differentiable if its Fourier coefficient  $a[k]$  satisfies the condition,*

$$|a[k]| \leq \frac{C}{|k|^{p+1+\varepsilon}} \quad \text{for some } C, p, \varepsilon > 0.$$

### 5.2.3 Mathematical formulation and related definitions

For the signal approximation, we consider the  $L$  length bandlimited approximation of the signals,  $f_i(t)$  for  $i \in \{1, 2, \dots, D\}$ , given by the Fourier series,

$$\widehat{f}_i(t) = \sum_{k=-L}^L b_i[k] \exp(j2\pi kt) \quad \text{for } t \in [0, 1], \quad (5.1)$$

where the coefficients  $b_i[k]$  for  $k \in \{-L, \dots, L\}$ , will be a function of  $a_i[k], k \in \mathbb{Z}$ . In particular, the approximation coefficients  $b_i[k]$  are chosen such that it minimizes the approximation error measured with respect to distance metric  $d(\cdot)$ , such as  $\mathcal{L}_1$ ,  $\mathcal{L}_2$  or  $\mathcal{L}_\infty$  norm, subject to an overprediction constraint. Mathematically, this is stated as,

$$\arg \min_{\widehat{f}_i(t)} d(f_i(t), \widehat{f}_i(t)) \quad \text{subject to } \widehat{f}_i(t) \geq f_i(t). \quad (5.2)$$

Since the Fourier basis is orthogonal, the problem now translates into an equivalent optimization in terms of the coefficients  $a_i[k]$ 's and  $b_i[k]$ 's. For instance, when the distance metric is the  $\mathcal{L}_2$  norm, the equivalent optimization is

$$\arg \min_{b_i[k]} \sum_{k=-L}^L |a_i[k] - b_i[k]|^2 \quad \text{subject to } \widehat{f}_i(t) \geq f_i(t). \quad (5.3)$$

The constraint in the considered optimization problem always ensures an over-prediction of the true signal, hence the signal approximation obtained by this method will be called the *envelope approximation*. For simplicity, we will denote the envelope approximation of a signal  $f(t)$  by  $\hat{f}_{\text{env}}(t)$ .

### 5.2.4 Distance measures of interest

For the overpredictive signal approximation, we will construct an envelope for the signal recorded at each client device. The envelope approximation  $\hat{f}_{\text{env}}(t)$  of the signal  $f(t)$  is obtained by solving the optimization problem described in (5.2), with respect to a distance metric  $d(\cdot)$  defined over the signal space. In this chapter, the  $\mathcal{L}_1$  and  $\mathcal{L}_2$  distance measures will be analyzed, and the  $\mathcal{L}_\infty$  formulation will be used to derive performance bounds. Using the Fourier representations of  $\hat{f}_{\text{env}}(t)$  (akin to (5.1), with the subscript indices dropped) and  $f(t)$ , and the envelope property  $\hat{f}_{\text{env}}(t) \geq f(t)$  it can be seen that,

$$\|\hat{f}_{\text{env}} - f\|_1 = b[0] - a[0], \quad (5.4)$$

$$\|\hat{f}_{\text{env}} - f\|_2^2 = \sum_{|k| \leq L} |b[k] - a[k]|^2 + \sum_{|k| > L} |a[k]|^2 \quad (5.5)$$

$$\|\hat{f}_{\text{env}} - f\|_\infty \leq \sum_{|k| \leq L} |b[k] - a[k]| + \sum_{|k| > L} |a[k]| \quad (5.6)$$

For ease of notation, the signal approximation error corresponding to the  $\mathcal{L}_1$ ,  $\mathcal{L}_2$  and  $\mathcal{L}_\infty$  distance measures will be denoted by  $\text{SA}_1$ ,  $\text{SA}_2$  and  $\text{SA}_\infty$  respectively. The upperbound in (5.6) will be used to show the performance bounds of envelope approximations in Section 5.4.1.

## 5.3 Algorithms for overpredictive analytics

In this section we revisit the distributed signal approximation application using federated learning illustrated in Fig. 5.2, where there are  $D$  devices and a central server. We discuss the algorithmic scheme for performing signal analytics in a federated learning setting, where each device reports an envelope approximation of the signal observed. The optimization program implemented at the client devices and the signal analytics performed at central server are discussed below.

### 5.3.1 Envelope approximation at the clients

The optimization problem of interest at the client devices, is to determine the best possible signal approximation, which forms an envelope of the true signal. This can be stated as,

$$\min \text{SA}_q := \int_0^1 \left| \widehat{f}(t) - f(t) \right|^q dt \text{ subject to } \widehat{f}(t) \geq f(t) \quad (5.7)$$

for  $q = 1, 2$ , where the above minimizations are over  $\widehat{f}_1(t), \dots, \widehat{f}_D(t)$ . These device level analytics problems can be efficiently solved, by using the equivalent forms of the function norms in the Fourier domain, discribed in (5.4) and (5.5).

For  $q = 1$ , the above envelope approximation formulation will become a *linear program*, since the cost function in the Fourier representation is the difference of the zero frequency components of the envelope and true signal. When  $q = 2$  the problem is equivalent to a *quadratic program with linear constraints*, with the objective described by the squared error of the Fourier coefficients within bandwidth  $L$ . Both  $q = 1$  and  $q = 2$ , has linear inequality constraints due to the envelope overprediction criteria. Since linear programs are relatively easier to implement, the client devices with limited hardware can choose to solve  $\mathcal{L}_1$  optimization problem over the  $\mathcal{L}_2$  problem.

#### Envelope optimization with discrete-time signals

In practice, the recorded client device signals are available at  $n$  discrete time samples. Since the true signal values are unknown at all times except the sampled points, the set of inequality constraints in discrete-time case is a finite subset of the envelope constraints in the continuous time case. Hence, the discrete-time optimization yields a lower cost than the continuous-time optimization. A related analysis of the discrete-time envelope optimization in the context of TV whitespace application, along with Mean Squared Error (MSE) performance is discussed in [107]. More detailed analysis of the discrete time envelope optimization is presented in Sec 5.4.

### 5.3.2 Aggregation algorithms at the server

The signal approximations obtained from the clients will be combined at the server to learn the signal analytics of interest. In general a signal analytic is a function over the envelope approximations  $\widehat{f}_1(t), \widehat{f}_2(t), \dots, \widehat{f}_N(t)$ . Examples of such analytics will include

the aggregate function,  $\widehat{s}(t) := \sum_i \widehat{f}_i(t)$ , or a statistical quantity such as the empirical CDF,

$$\widehat{F}_N(x) := \frac{1}{N} \sum_{n=1}^N \mathbb{1}_{(-\infty, x]}(\widehat{f}(t_n)), \quad (5.8)$$

where  $\mathbb{1}_{(-\infty, x]}(Y)$  represents the 0–1 indicator function for the probability event  $\{Y \leq x\}$ , and  $t_n$  for  $n \in \{1, 2, \dots, N\}$  represents the time samples in the interval  $[0, 1]$ . It is known that several statistical properties can be inferred using this classical Glivenko-Cantelli estimate of the CDF [55]. Since the Glivenko-Cantelli estimate satisfies the uniform convergence property, at the server there exists an implicit minimization of an empirical loss function involving the true and estimated CDFs, i.e. the objective function,

$$\|\widehat{F}_N(x) - F(x)\|_\infty, \quad (5.9)$$

is minimized by when  $\widehat{F}_N(x)$  is the Glivenko-Cantelli estimate of  $F(x)$ . This minimization task conforms with the definition of federated optimization [1]. Application of CDF estimation based signal analytics will be discussed in the section 5.5.

### 5.3.3 Algorithm sketch for over-predictive signal analytics

It is assumed that each client device in the federated learning model works in a distributed manner. To ensure  $\widehat{f}_i(t) \geq f_i(t)$  for  $i \in \{1, 2, \dots, D\}$ , we propose that each device can perform envelope approximation of its observed signal. The following steps are proposed for obtaining a bandwidth- $L$  approximation  $\widehat{f}_i(t)$  of  $f_i(t)$ :

1. Each device  $i$  records its individual signal  $f_i(t)$ , computes its envelope  $\widehat{f}_{i,\text{env}}(t)$ , using the envelope optimization (5.7), and communicates the  $(2L + 1)$  Fourier coefficients to the server.
2. Using Fourier coefficients from each client device, the server calculates a global model (or statistic) of interest, denoted as  $G_{\text{server}}(\widehat{f}_{1,\text{env}}, \widehat{f}_{2,\text{env}}, \dots, \widehat{f}_{D,\text{env}})$ .

Signal envelope calculation in Step 1 above is outlined next. For  $\mathcal{L}_1$  distance (see (5.4)), it will be calculated as,

$$\begin{aligned} & \text{minimize } b[0] - a[0] \\ & \text{subject to } \vec{b}^T \Phi(t) \geq f(t), \end{aligned} \quad (5.10)$$

where  $\Phi(t) = [\exp(-2\pi Lt), \dots, \exp(2\pi Lt)]^T$  and  $\vec{b} = (b[-L], \dots, b[L])^T$  are the Fourier series coefficients of the envelope approximation. The above linear program with linear constraints is solvable efficiently [107]. For  $\mathcal{L}_2$ , the cost function  $b[0] - a[0]$  is replaced by the quadratic cost in (5.5).

As  $L$  is increased, the envelopes  $\widehat{f}_{i,\text{env}}(t)$  become more proximal to their target  $f_i(t)$ . It is expected that the approximation errors,  $\text{SA}_1$  and  $\text{SA}_2$  will decrease as  $L$  increases. However, analyzing the dependence of  $\text{SA}_q$ ,  $q = 1, 2$  versus  $L$  is difficult. Accordingly, a naïve envelope approximation will be used to analyze the fundamental bounds on their tradeoff.

## 5.4 Performance bounds on optimal envelope approximation

In this section, we analyze the performance of the  $L$  bandwidth envelope approximation. In particular, we provide an upper bound for the envelope approximation error, using the idea of naïve envelope approximation scheme, described below .

### 5.4.1 A naïve envelope approximation scheme

First consider a single client device,  $i = 1$  in isolation. Let  $f_{1,\text{proj}}(t)$  be the orthogonal projection of  $f_1(t)$  on the span of  $\exp(j2\pi kt)$  for  $|k| \leq L$ . Then  $f_{1,\text{proj}}(t) = \sum_{|k| \leq L} a_1[k] \exp(j2\pi kt)$ . The naïve envelope approximation scheme is as follows [66]:

$$f_{1,\text{env}}(t) = f_{1,\text{proj}}(t) + C_0, \quad (5.11)$$

where  $C_0 = \|f_1 - f_{1,\text{proj}}\|_\infty$ . Using the triangle inequality,

$$C_0 \leq \sum_{|k| > L} |a_1[k]| \leq \sum_{|k| > L} \frac{C}{|k|^p}, \quad p > 1. \quad (5.12)$$

For  $p > 1$  [108, Sec. 2.2], we can show that,  $C_0 = O\left(\frac{1}{L^{p-1}}\right)$ .

### 5.4.2 Envelope approximation analysis

In this section we answer the question – *for what class of signals is the naïve envelope approximation scheme good?* The result stated below shows that there exist a certain class of signals for which the naïve approximation is order optimal to the optimal envelope scheme, while using  $\mathcal{L}_1$  or  $\mathcal{L}_2$  norm minimization.

In the result stated below we use the notation  $SA_q$  to denote the optimal envelope approximation error (see (5.4)), and the notation  $SA'_q$  to be the approximation error corresponding to the naïve envelope approximation in (5.11), where  $q \in \{1, 2\}$  according to the  $\mathcal{L}_1$  or  $\mathcal{L}_2$  cost metric.

**Theorem 5.2.** *The approximation errors of the optimal envelope signal and the naïve envelope signal, viz.  $SA_q$  and  $SA'_q$  respectively for  $q \in \{1, 2\}$ , are shown to be order optimal for a specific class of  $p$ -times differentiable signals as stated below.*

(i) *The approximation error of the optimal  $\mathcal{L}_2$  envelope signal,  $SA_2$ , is order optimal to the approximation error of the naïve envelope signal,  $SA'_2$  for the class of  $p$ -times differentiable signals where Fourier coefficients,  $|a_1[k]| = \frac{1}{|k|^p}$  for  $|k| > L$ . That is,*

$$1 \leq \frac{SA'_2}{SA_2} \leq \left(1 + \frac{1}{L}\right)^{2p-1}.$$

(ii) *The approximation error of the optimal  $\mathcal{L}_1$  envelope signal,  $SA_1$  is the same as the approximation error of the naïve envelope signal,  $SA'_1$  for the class of  $p$ -times differentiable signals with Fourier coefficients,  $a_1[k] \geq 0$  (non-negative) and  $a_1[k] = a_1[-k]$  (symmetric) for all  $k \in \mathbb{Z}$ . That is,*

$$SA_1 = SA'_1 = 2 \sum_{k>L} a_1[k].$$

The detailed proof is shown in [Appendix D.1](#).

*Remark:* The above result is shown for one (that is  $D = 1$ ) client device. For  $D$  clients with a sum signal analytic at the server (that is  $\widehat{s}(t) := \sum_{i=1}^D \widehat{f}_i(t)$ ),  $SA_1$  as well as  $SA'_1$  scale linearly with  $D$ . In contrast,  $SA_2$  and  $SA'_2$  will scale quadratically with  $D$ . Thus in both  $\mathcal{L}_1$  and  $\mathcal{L}_2$  norm based error, the ratio between the approximation errors will remain the same, as in the  $D = 1$  case discussed in the proof.

### 5.4.3 CDF of the envelope signal approximation

The CDF is a useful signal analytic to determine various statistical functions, especially in a distributed learning setting such as federated learning. Functions of the CDF estimates are beneficial in estimating customer usage statistics such as mean, median or peak demand in the electricity smart metering illustrated in Fig. 5.1. By invoking Theorem 5.2,

we derive performance bounds on the estimated CDF obtained from the envelope approximation algorithms (see Sec. 5.3). Let  $F_X(x)$  denote the CDF of the actual signal  $X(t)$  and let  $F_{X_{\text{env}}}(x)$  be the CDF of the envelope signal approximation,  $\widehat{X}_{\text{env}}(t)$ , with  $(2L+1)$  Fourier coefficients. In the result below, we characterize an upper bound on the pointwise difference between the actual and the estimated CDFs, as stated below.

**Theorem 5.3.** *The pointwise difference between the CDF of the actual signal,  $F_X(x)$  and the CDF of the envelope signal approximation,  $F_{\widehat{X}_{\text{env}}}(x)$ , with  $(2L+1)$  Fourier coefficient is bounded by,*

$$F_X(x) - F_{\widehat{X}_{\text{env}}}(x) \leq \frac{C}{L^{\frac{2p-1}{3}}}, \quad (5.13)$$

where  $C = \left(4^{\frac{1}{3}} + 2 \times 4^{-\frac{2}{3}}\right) \frac{f_{X,\max}^{2/3}}{(2p-1)^{1/3}}$  and  $f_{X,\max} := \max_x \left[ \frac{dF_X(x)}{dx} \right]$ .

The proof of this result is available in [Appendix D.2](#)

#### 5.4.4 Effect of subsampling on CDF estimate

In practice, signals are obtained as discrete samples over a finite interval. Thus, it is essential to understand how envelope approximation can be performed over signal samples and evaluate the tradeoff between the envelope approximation error and the sampling rate. In this connection, we will consider the aspect of signal sampling while limiting the discussion to the  $\mathcal{L}_2$  distortion. However, a similar analysis also extends to the  $\mathcal{L}_1$  distortion metric. The envelope signal approximation problem posed in (5.7) is reformulated to reflect the finite signal sample assumption as described below. If  $\vec{b}_{\text{opt}}$  represents the minimizer of  $\mathcal{L}_2$  distance metric applied in (5.7), then we define,

$$\begin{aligned} \vec{b}_{\text{app},n} &:= \arg \min_{\vec{b}} \sum_{|k| \leq L} |b[k] - a[k]|^2 + \sum_{|k| > L} |a[k]|^2 \\ &\text{subject to } \widehat{f}_{\text{env}}(t_n) \geq f(t_n), \quad \forall t_n \in \left\{0, \frac{1}{n}, \frac{2}{n}, \dots, 1\right\}, \end{aligned} \quad (5.14)$$

where  $n$  is a positive integer denoting the number of signal samples. Also, we define the approximately optimal envelope signal,  $f_{\text{app},n}(t) := \vec{b}_{\text{app},n} \Phi(t)$ . The above formulation in (5.14) has only finite inequality constraints as against the infinite inequality constraints in (5.7). Due to the relaxation of the envelope constraints attributed to signal sampling,

the  $\mathcal{L}_2$  cost of the approximately optimum envelope is lesser than the optimum envelope. This difference in the  $\mathcal{L}_2$  cost of the envelope error, for the infinite envelope constraint and the sampled envelope constraint relaxation as characterized in [107] is written as,

$$\sum_{|k| \leq L} \left[ |b_{\text{opt}}[k] - a[k]|^2 - |b_{\text{app},n}[k] - a[k]|^2 \right] \leq 2(b_{\text{app},n}[0] - a[0]) \left( \frac{c + c'}{n} \right) + o(1/n). \quad (5.15)$$

The constants  $c$  and  $c'$  in the above upperbound expression corresponds to the maximum slopes of the actual signal and the approximately optimal envelope signal respectively. That is,  $|f'(t)| \leq c$  and  $|f'_{\text{app},n}(t)| := c'$ . The upperbound on  $c'$  is shown to be  $2\pi L \|f\|_\infty$ , in [107].

Let  $F_X(x)$  be the CDF of the actual signal,  $X(t) := \sum_{k \in \mathbb{Z}} A[k] \exp(j2\pi kt)$  for  $t \in [0, 1]$ . Further, let  $F_{X_{\text{app},n}}(x)$  be CDF of the envelope signal approximation with the subsampled constraints,  $\widehat{X}_{\text{app},n}(t) := \sum_{|k| \leq L} B_{\text{app},n}[k] \exp(j2\pi kt)$  for  $t \in [0, 1]$ , where the set of coefficients  $\{B_{\text{app},n,i}[k] : -L \leq k \leq L\}$  is as defined in (5.14). An upperbound between the pointwise difference of the two CDFs is shown below,

**Theorem 5.4.** *The pointwise difference between the CDF of the actual signal,  $F_X(x)$  and the CDF of the subsampled envelope signal approximation,  $F_{X_{\text{app},n}}(x)$ , with  $(2L + 1)$  Fourier coefficient is bounded by,*

$$F_X(x) - F_{X_{\text{app},n}}(x) \leq C \times \left\{ \frac{4}{2p-1} \frac{1}{L^{2p-1}} + 8\mu_{\text{app},n} \frac{c+c'}{n} + o(1/n) \right\}^{1/3}, \quad (5.16)$$

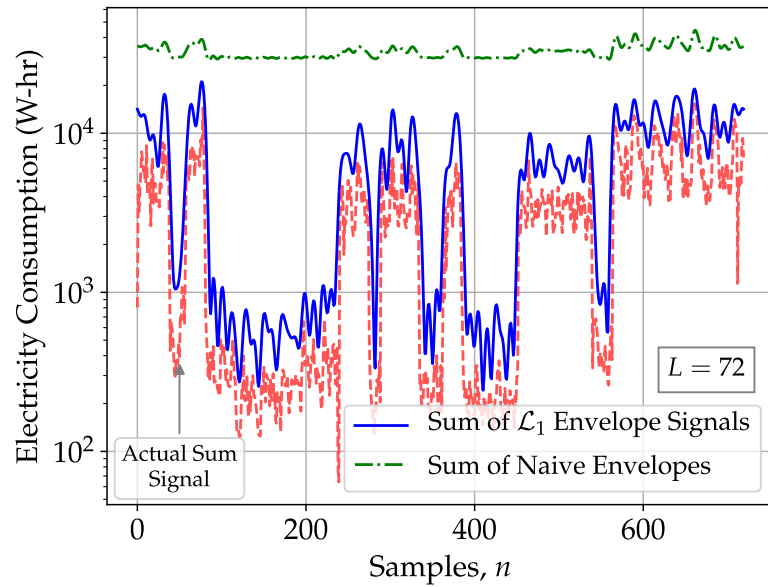
where  $C = \left(2^{\frac{1}{3}} + 2^{-\frac{2}{3}}\right) f_{X,\text{max}}^{2/3}$  and  $f_{X,\text{max}} := \max_x \left[ \frac{dF_X(x)}{dx} \right]$ . The constant,  $\mu_{\text{app},n} = \mathbb{E} [B_{\text{app},n}[0] - A[0]]$ , where  $B_{\text{app},n}[0]$  and  $A[0]$  are the zero frequency components corresponding to the signals  $\widehat{X}_{\text{app},n}(t)$  and  $X(t)$  respectively.

The proof of this result is available in [Appendix D.3](#).

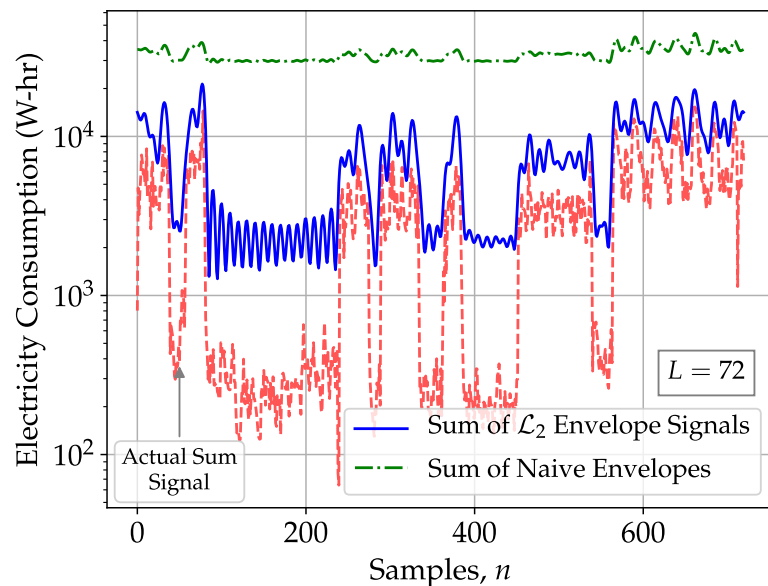
## 5.5 Experimental results and discussion

### 5.5.1 Simulations on the electricity consumption dataset

To analyze the effect of envelope approximation in a federated learning model, we have conducted experiments on an available electrical energy consumption dataset [64], consisting of hourly electricity consumption of 39 users from a residential building. The



(a)



(b)

Figure 5.3: The sum of signals obtained by distributed envelope approximation schemes are compared with the ground truth signal, when the number of coefficients communicated are  $L = 72$ . Plot (a) and (b) consider  $\mathcal{L}_1$  and  $\mathcal{L}_2$  cost functions respectively. It is observed that the optimal envelope approximation signal estimates the peak energy demand regions.

experimental results discussed here, are conducted on those users having atleast 30 days of data with synchronized timestamps – which turned out to be 37 users. In the original dataset, energy (in W-hr units) measurements corresponding to three phases were avail-

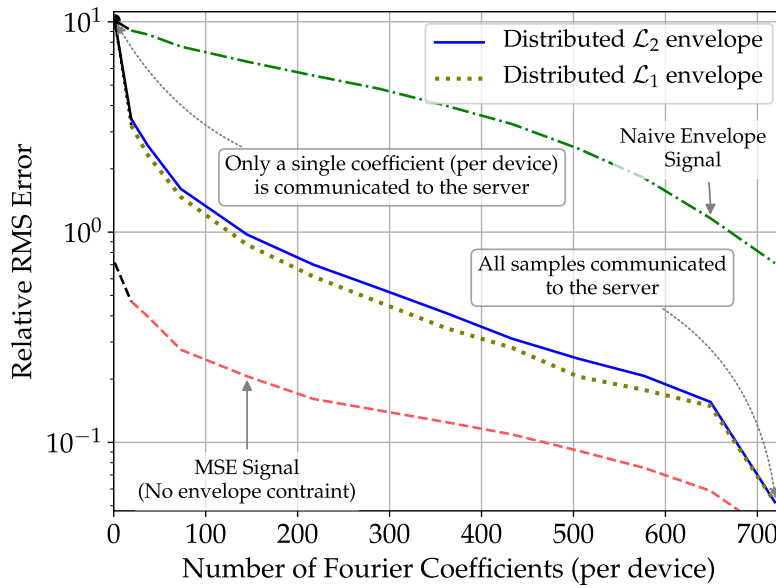


Figure 5.4: This plot depicts the tradeoff between normalized root mean-squared (RMS) error and the number of Fourier coefficients transmitted per device. The distributed envelope approximation scheme based on  $\mathcal{L}_1$  and  $\mathcal{L}_2$ -norms converge to the zero error as the number of communicated coefficients increase. The plot also shows the RMS error for the naïve envelope signal and the MSE optimal signal (which is devoid of any envelope constraints).

able. However, for the doing the experimental study we have considered only a single phase (represented as W3 in the dataset).

**Hardware/Software Specifications:** The simulations on the dataset were performed in a PC with the Processor model – Intel(R) Core(TM) i3-2310M CPU @ 2.10GHz, 2100 Mhz, 2 Core(s), RAM 6 GB; and implemented in MATLAB 2015b (Windows platform) using the standard curve fitting toolbox and CVX package (ver 2.1).

### 5.5.2 Communication cost and approximation error tradeoff

The tradeoff between the number of communicated Fourier coefficients and the envelope approximation error, is studied by considering a sum signal analytic (or the sum of user energy consumptions) to be learned at the central server. Each Fourier coefficient communicated to the server is a real valued floating point number which is typically 4 Bytes (or 32 bits). In this setting, each client device (i.e. the 37 users) sends the Fourier coefficients

of the envelope approximation, representing 30 days of hourly energy consumption, or 720 signal samples (refer Fig. 5.3 for the time-series plot of the sum signal analytic). In the tradeoff plot shown in Fig. 5.4, we notice that there is a graceful degradation of the relative root mean-squared (RMS) error of the optimal envelope approximation scheme for both  $\mathcal{L}_1$  and  $\mathcal{L}_2$  cost functions, as the number of Fourier coefficients transmitted are reduced. It is observed that the distributed signal approximation schemes approach an error of zero when 720 Fourier coefficients are transmitted. However, the naive approximation, which adds only a constant to the projection based Fourier representation, fails to capture the envelope trend for the considered dataset. For reference, we have marked two extremum points – first corresponding to the least communication scheme (that is  $L = 1$ ), and the second where all coefficients are communicated (that is  $L = 360$ ). The  $\mathcal{L}_1$  cost function results in a relatively lower RMS error compare to  $\mathcal{L}_2$ , which is due to the better time-series fit of  $\mathcal{L}_1$  as shown in Fig. 5.3. Further, we observe that the relative approximation error of classical MSE minimizer (that is, without the envelope constraint) is lower than distributed envelope by one order of magnitude.

### 5.5.3 Cumulative distribution function based signal analytics

In this experimental study, we allow the central server to learn the CDF of the signal samples from the approximate signals reported by the individual clients. From the learned CDF, the central server can infer statistical properties, which are symmetric functions of the data, to measure the electricity usage patterns at the client devices. In Fig. 5.5, we illustrate the convergence of the CDF estimate obtained from the envelope approximation samples to the baseline CDF, obtained by a model which access to all the raw signal samples from all clients. At low approximation levels, for instance  $L = 36$ , the estimated CDF of the envelope scheme is much away from the true CDF. As we increase the number of communicated Fourier coefficients, the gap between the CDFs is observed to reduce. Another important observation is that the overprediction based CDFs (envelope as well as the naive approximation schemes) always appears to the right of the true CDF. This is because of the envelope constraint at the devices.

Using these CDF plots, the quantile estimates can also be inferred. In Table 5.1, we compare the quantiles of the true CDF with the quantiles of the CDF estimates. It

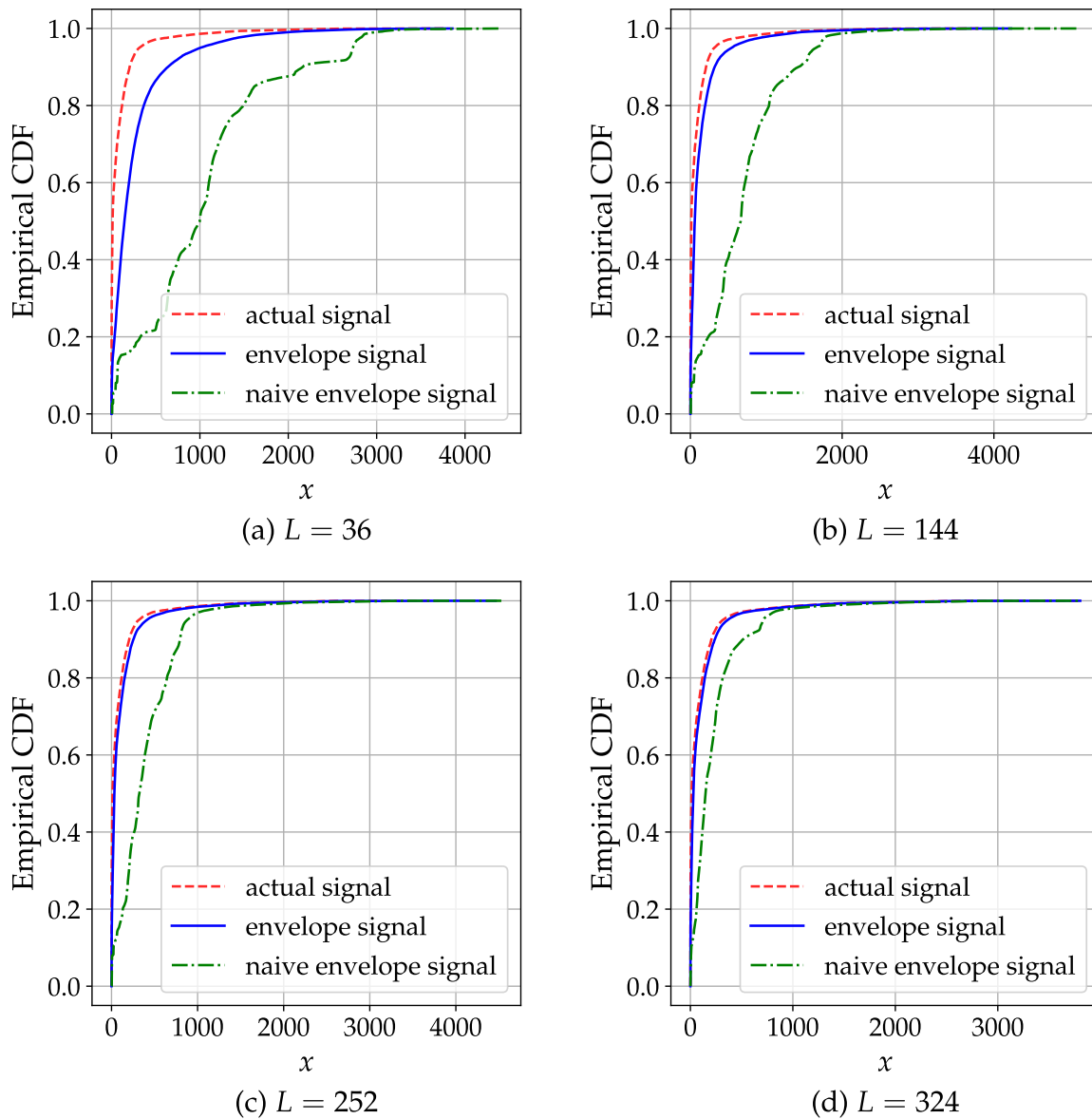


Figure 5.5: Plot illustrating the convergence of the empirical CDFs of the envelope signal to the actual CDF (obtained by pooling the raw signal samples at server) for the  $\mathcal{L}_2$  cost function. The optimum as well as the naïve envelope signal CDFs always appears towards the right of the true CDF, due to the envelope constraint.

is noted that the higher order quantiles deviate much as compared to the lower ones, which is attributed to the overprediction constraint at the devices. The accuracy of the quantile estimates in Table 5.1 can be improved by an appropriate choice of the basis representation. This shall be addressed in a future work.

Table 5.1: A comparison of the quantiles of the true signal with the envelope approximation signal

Quantile	Actual Signal	Cost Function	Envelope Signal			Naïve Envelope Signal		
			$L = 36$	$L = 180$	$L = 324$	$L = 36$	$L = 180$	$L = 324$
<b>10%</b>	0	$\mathcal{L}_1$	0.21	$10^{-6}$	$10^{-7}$	66.34	39.19	7.79
			$\mathcal{L}_2$	3.39	0.80	0.11		
<b>50%</b>	6.09	$\mathcal{L}_1$	19.25	10.12	6.78	754.85	410.91	116.05
			$\mathcal{L}_2$	65.36	27.55	13.36		
<b>90%</b>	204.53	$\mathcal{L}_1$	429.59	272.41	225.71	2162.7	1229.2	514.11
			$\mathcal{L}_2$	431.21	268.6	218.42		

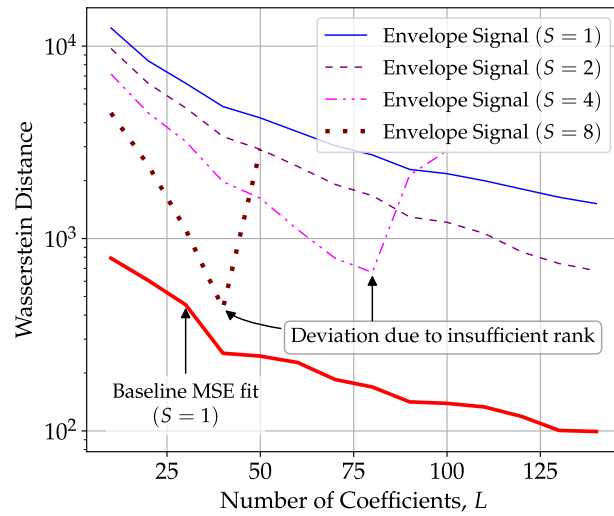
### 5.5.4 Effect of subsampling

In practice, as the signal acquisition in the edge device will not be a recording a continuous signal, we will investigate the effect of signal sampling on the envelope approximation algorithms. Since the dataset considered here is already discrete time-series with 720 samples, we will see the effect of subsampling on the various performance metrics. In particular, for this experimental study we consider three performance error metric, *viz.* (i) Wasserstein distance between the estimated envelope CDF and the actual CDF, (ii) Number of envelope constraint violations with respect to the actual 720 samples, and (iii) Peak envelope error after subsampling. We have used the 1-D Wasserstein distance between the CDF of the envelope signal and the CDF of the actual signal, defined as,

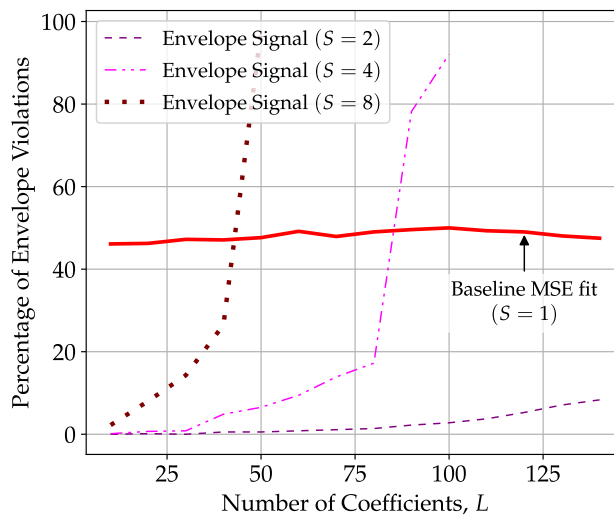
$$\mathcal{W}(X, \hat{X}_{\text{env}}) := \int_0^1 \left| F_X^{-1}(z) - F_{\hat{X}_{\text{env}}}^{-1}(z) \right| dz,$$

where  $F_{\hat{X}_{\text{env}}}(x)$  is the envelope signal CDF and  $F_X(x)$  is the true signal CDF. The number of envelope constraint violations and the peak envelope error are computed based on the pointwise difference between the sum of envelope approximated time-series and the sum of actual time-series. In Fig .5.6, we show the various performance metrics considering the  $\mathcal{L}_1$  cost function for different subsampling rate, i.e.  $S \in \{1, 2, 4, 8\}$ . At a subsampling rate of  $S$ , the number of envelope constraints is  $720/S$ . From Fig. 5.6 (a), it is observed that subsampling by considering alternate signal samples, i.e.  $S = 2$ , reduces the Wasserstein distance between the CDFs of the envelope signal and the true signal, when compared with the envelope without subsampling, i.e.  $S = 1$ . This is attributed to the envelope constraint relaxation upon subsampling, which results in a lesser minima for the same objective function. Since the CDF of the MSE approximation signal does account for the envelope constraints the Wasserstein distance approach zero with increasing approximation coefficients, as shown in Fig. 5.6 (a).

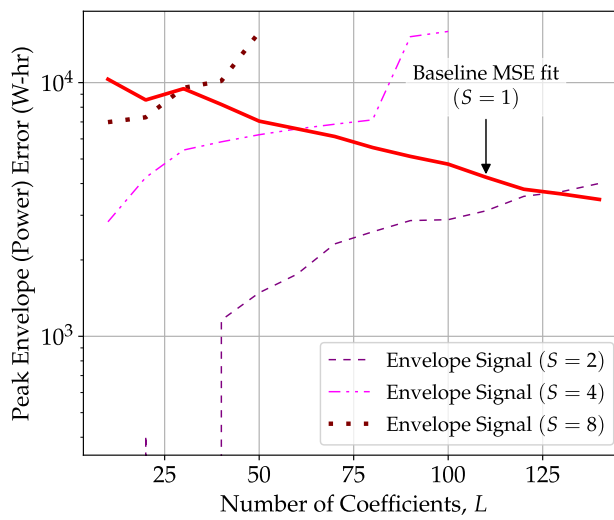
However, subsampling introduces violations of the envelope constraint as shown in Fig. 5.6 (b)-(c). Both the percentage of envelope constraint violations and the peak error due to envelope constraint violation increase with the number of approximation coefficients  $L$ . This is because of the effect of signal aliasing and signal overfitting introduced due to subsampling. Further it is noted that the signal reconstruction from subsampled time-series fails when the number of Fourier coefficients is more than the total number of signal



(a)



(b)



(c)

Figure 5.6: The figures depict the effect of subsampling on the  $\mathcal{L}_1$  cost based envelope approximation based on three performance metrics: (a) Wasserstein distance of estimated envelope CDF and true CDF, (b) percentage of envelope violations at different subsampling rates, (c) peak envelope error due to subsampling.

Table 5.2: Comparison of different error metric for the  $\mathcal{L}_1$  and the  $\mathcal{L}_2$  cost functions, assuming  $S = 2$  subsampling

Error Metric	Cost Function	Envelope Approximation							
		$L = 10$	$L = 20$	$L = 30$	$L = 40$	$L = 50$	$L = 60$	$L = 70$	$L = 80$
Wasserstein Distance	$\mathcal{L}_1$	9725.63	6396.24	4775.37	3390.32	2906.97	2370.42	1913.22	1670.10
	$\mathcal{L}_2$	11378.52	8051.19	6050.24	4403.93	3867.93	3220.06	2725.07	2400.25
Number of Envelope Violations	$\mathcal{L}_1$	0	1	0	4	4	6	8	10
	$\mathcal{L}_2$	0	0	0	2	2	5	6	7
Peak Envelope Error (in W-Hr)	$\mathcal{L}_1$	0	408.26	0	1158.47	1484.85	1756.70	2313.55	2581.35
	$\mathcal{L}_2$	0	0	0	683.21	793.70	1271.43	2044.13	2363.26

samples, i.e.  $2L + 1 \geq 720/S$ . In this case envelope approximation algorithm fails to solve the optimization program owing to rank deficiency. This effect is shown in Fig. 5.6 (a), where the Wasserstein distance returned by the solver starts to increase at  $L = 45$  for  $S = 8$  and  $L = 90$  for  $S = 4$ . For  $S = 2$ , the rank violation will begin at  $L = 180$  although not shown in the plot.

The rank deficiency also results in an abrupt increase in envelope errors as depicted in Fig. 5.6 (b)-(c). The tradeoff between the sampling rate, Wasserstein's distance and the envelope violations, suggests that envelope approximation algorithm should be limited to  $2L + 1 < 720/S$ , where the aliasing effects due to subsampling are small. A similar trend is seen to hold when the  $\mathcal{L}_1$  cost function is replaced by the  $\mathcal{L}_2$  cost function. We make a comparison between the envelope approximation algorithms of  $\mathcal{L}_1$  and  $\mathcal{L}_2$  cost functions based on the three performance metrics for  $S = 2$  in Table 5.2. Due to the relatively poor signal fitting of the  $\mathcal{L}_2$  compared to  $\mathcal{L}_1$  as shown in Fig. 5.3, the Wasserstein distance is more for  $\mathcal{L}_2$  cost. However the number of envelope errors and the peak envelope error is reduced because of the higher envelope constraint guard region at lower signal amplitudes for the  $\mathcal{L}_2$  approximation. Thus, for the low envelope approximation regime, i.e.  $L = 10$  to  $L = 50$ , it is suggestive to employ the  $\mathcal{L}_2$  approximation algorithm when subsampling, as the envelope constraint violations are better controlled.

## 5.6 Summary

In this chapter, we proposed signal approximation algorithms to perform overprediction of distributed signals in a client-server federated learning model. Using a convex optimization framework, we determine overpredicting signal representations locally at each edge device, further communicating these to the cloud server to compute signal analytics. Such signal analytics, like the aggregate signal or the Cumulative Distribution Function (CDF) of the time-series signal, derived from the individual signal representations using federated aggregation, were analyzed to determine the tradeoff characteristics between the mean-squared approximation error and the number of approximation coefficients (communicated bytes). Particularly for the CDF approximation, we used the Glivenko-Cantelli estimate to estimate upperbounds on the pointwise difference between the actual CDF and the approximate CDF. Experiments on a residential energy consumption dataset

validate the analytical performance bounds of the envelope signal analytics considered. In the future, we envisage extending the proposed algorithms for time-series prediction considering the signal overprediction constraints.

# Chapter 6

## Conclusions and Future Research Directions

### 6.1 Concluding remarks

IoT devices have to handle massive amounts of data with inherent resource constraints. As a consequence, it is becoming increasingly difficult to store or communicate the acquired data to a central server for signal (or data) processing. Therefore, to address these problems, we have focused on developing algorithms for edge processing that minimize communication and extract maximum signal information for inference tasks. In the first part of this thesis (Chapter 2-3), we have addressed the question “*How can we design nearly-optimal quantizers in low power edged devices with minimal hardware resources?*”. To answer this question, we introduced piecewise linear approximation on the signal source PDF to speedup the well-known Lloyd-Max quantizer.

The near-optimal quantizer design, which we term as *Approximate Lloyd-Max (ALM)* algorithm, bypasses the integration step to compute the centroid. One of the chief advantages of ALM is the reduction in the update computations, which has been studied in Chapter 2. To analyze the speedup obtained through the approximation, we employ the Perron-Frobenius theory found in Markov chains’ literature, and demonstrate the fast convergence of ALM. In particular, we establish an exponential convergence rate for ALM near the limit of the optimal Lloyd-Max quantizer. Using the framework developed for

ALM, a data-driven quantizer design is proposed, which we term as *Learning ALM* or LALM. The proposed LALM quantizer learns the quantizer levels that eventually converges in probability to the  $k$ -means quantizer. Through simulations on an Android-based edge device, we have demonstrated the energy efficiency, speedup, and near-optimality of LALM compared to well-known quantization schemes such as  $k$ -means and scalar LVQ.

Chapter 3 has explored quantizer design for edge devices with specific system constraints. In particular, the overprediction constraint has been studied, and the main question that we have answered here are: “*Can we design an overpredictive quantizer using online updates?*” and “*Is it possible to extend the ALM algorithm to accommodate the overprediction system constraint?*”. We have answered both questions in the affirmative. To this end, we have developed the *Stochastic Approximation based Envelope (SANE)* quantizer and the *Approximate Envelope Quantizer (AEQ)* respectively. SANE is similar in spirit to the LVQ, which is a well-known quantizer design based on gradient descent. However, in SANE, we have incorporated a two time-scale online quantizer level update procedure that simultaneously estimates the signal source probability density. The convergence properties of SANE are derived using the ODE approach, which enable a choice over learning rates for achieving fast convergence. For a TV whitespace dataset application, we learn overpredictive quantizers to approximate spatial contours. In particular, we have demonstrated the efficacy of SANE relative to other state-of-the-art methods in terms of higher approximation accuracy and faster convergence. The second part of Chapter 3 has discussed AEQ, which is an extension of piecewise linear approximation-based ALM, that accommodates the overprediction constraint. A data-driven extension termed *Learning AEQ* or *LAEQ* has been proposed, and a comparative study of LAEQ with SANE is performed.

Next, in the second part of this thesis, we have focused on distributed signal processing for federated learning applications. Specifically, we have addressed the questions: “*How to scale the quantizer design at edge devices to a distributed problem such as federated learning?*”, and “*How overprediction system constraint can be addressed through the choice of signal representation in federated learning?*”

In Chapter 4, we have presented distributed quantizer design applications in feder-

ated learning (FL). The algorithms for FL involve signal processing at the edge devices and signal aggregation at the central server. We have compared the performance of various quantizer design schemes at the edge devices in terms of computational complexity, space complexity, and sample complexity. As FL architecture supports heterogeneous devices, such a comparison has been of value, as it helps edge devices to choose between quantizer designs, looking into the energy and hardware constraints. By using appropriate aggregation schemes which linearly combine the quantization levels at the central server, we infer the performance improvement of the global quantizer designed using the federated approach. In particular, we have analytically demonstrated that the mean squared error of the quantizer after federated aggregation is less than the convex weighted combination of individual device level quantizer. The result holds for non-identical and independent datasets across edge devices. In addition, we have performed simulations on 3 separate datasets; *viz.* a synthetically generated Beta distributed dataset, an available fitness tracker dataset and the standard MNIST handwritten digits dataset. In each of these datasets we have characterized the communication cost vs. accuracy (or MSE) tradeoff, and also have verified the MSE reduction property of federated aggregation shown via mathematical analysis.

Finally, in Chapter 5, we have solved the signal representation problem in FL to facilitate system specific constraints such as overprediction. We have proposed algorithms that can be implemented in resource constrained edge devices, by using optimization tools accounting for signal sampling. Our discussions have been centered around minimizing the  $\mathcal{L}_1$  and  $\mathcal{L}_2$  cost of the overpredicted signal, also termed as the *envelope signal*. For the class of  $p$ -times differentiable signals we have characterized the tradeoff between the mean squared error and the number of signal representation coefficients. Apart from the optimization at the edge devices, we have proposed a federated optimization at the central server for computing signal analytics, which are essentially functions computed over the cumulative distribution function (CDF). By using the Glivenko-Cantelli theorem, we reconstruct the CDF of the time-series signals from the signal representation transmitted over a rate constrained channel. Through our analysis we have characterized upperbounds on the pointwise reconstruction error between the estimated and the actual CDF, considering the effects of signal subsampling. We have experimentally demonstrated the efficacy

of the proposed algorithms and the federated optimization schemes on an off the shelf residential electrical energy consumption dataset with several independent consumers.

## 6.2 Directions for future research

The research in this thesis has lead a number of new challenging and open research questions. In what follows, we conclude this thesis by providing some directions for future research.

### 6.2.1 Approximate vector quantization

In Chapter 2 and 3, we have restricted the discussion to scalar quantizers. Our goal has been to minimize the computations while performing the quantizer design. If the signals are vectors, the quantizer design problem will need to account for the higher dimensional partitions, or *voronoi regions*. For instance, in the case of the signal  $\vec{x} \in \mathbb{R}^2$ , the voronoi regions would be polygons on the  $\mathbb{R}^2$  plane. The equivalent of piecewise linear approximations in this case will be planar approximations limited to specific regions. Speeding up vector quantizers even in  $\mathbb{R}^2$  is a challenge since voronoi regions have different shapes and estimation of the approximate PDF will be affected by the curse of dimensionality.

Some initial experiments conducted in this direction suggests that rectangular approximation of voronoi partitions is a viable option, as it cuts the space and computational complexity significantly. However, analyzing the degradation of performance metrics such as MSE will remain challenging in 2-D as there are approximation errors due to piecewise planar fitting and errors leading from modifying the voronoi partitions. Extending the existing 1-D ALM (or even AEQ) with some preprocessing step might be another plausible solution. By identifying independent components in the given 2-D data, a product quantizer (i.e. two independent 1-D quantizers along each independent components) is a computationally feasible alternative. Using efficient grid-search algorithms it may be possible to identify the optimum quantizer levels among the designed product quantizer.

We performed a comparative study of different vector quantization algorithms, *viz.* Product-LALM, equi-spaced, k-means ++ and LVQ. For the study we considered IID Beta(4,2) distribution with number of levels  $K = 512$  and dimension  $d = 32$ . Among the choices of quantizers, we have seen that LVQ gives the maximum speedup while k-means

++ provides the highest accuracy (i.e., lowest MSE). In this context, we observe that it is relevant to characterize the tradeoff between speedup and accuracy. The role in the design of `Product-LALM` will be to provide near-optimal performance both in terms of speedup and accuracy.

### 6.2.2 Sample complexity of LALM and LAEQ

In this thesis, we have not rigorously computed the sample complexity of data driven quantizers such as LALM and LAEQ. Instead we have only provided a weak upperbound for the same by relying on results available in literature [49]. The exact sample complexity analysis might require tools from statistical learning theory such as the Vapnik-Chervonenkis (VC) dimension, which is challenging to determine due to the effects of approximation. Further, these analysis often require specific signal class assumptions, which might hinder the generalizability of such results.

### 6.2.3 Extension of FL methods for generic non-IID signal sources

One of the key challenges in the federated learning architecture is the system heterogeneity and the non-IID nature of the datasets present at the client devices. In Chapter 4, we have considered a subclass of non-IID signal sources, that is non-identical but independently distributed data. This assumption has made the analysis tractable, as the joint PDF of the pooled dataset turns out to be a linearly weighted mixture distribution. However, in practice the datasets at the client devices might be correlated, thus forcing us to seek tools to handle signal sources with correlation.

The sensor calibration literature provides a plausible solution to handle this correlation in distributed datasets. In this setup, we can think of each dataset to be a random process which is a filtered version of an original IID random process, which is unknown. The quest here to determine inverse map of this filtering function, that will return a common dataset with IID entries. The inverse map computation for linear models are already available in literature [109]. However, many IoT applications, for instance heart rate monitoring using fitness trackers, have non-linear models. To account for the non-linearity, we have to rely on computationally efficient inverse operations with graceful performance degradation. Another alternative for the inverse problem discussed above could be using

optimal transport theory, which helps to map the signal PDF of individual devices to a common PDF by optimizing the certain distance metrics (e.g. Wasserstein distance)

#### **6.2.4 Privacy versus utility tradeoff in FL**

In Chapter 5, the main focus was on minimizing the communication cost in federated learning. Although, the proposed schemes ensure implicit ensure privacy to the edge device user, we have not rigorously defined the notion of privacy. For applications such as smart meters for electricity consumption monitoring, the major challenge is to preserve privacy sensitive data such as the occupancy information of the consumers. In a future work, we intend to draw insights into the tradeoff between privacy and utility of the data using suitable mathematical formulation.

# Appendices



# Appendix A

## Supplement to Chapter 2

### A.1 Properties 1-6 for illustration in Sec.2.4.2

The proof sketch for the properties of ALM update matrix is shown.

**Property 1:**  $P$  is row stochastic. (For proof see [23], Sec. 4.3)

**Property 2:** Eigenvalues of  $P$  satisfy  $|\lambda| \leq 1$ .

Consider a vector  $\vec{v}$ , and the inequality  $\|P\vec{v}\|_1 \leq \|P\|_1\|v\|_1$ . Since  $\|P\|_1 = 1$ , all eigenvalues  $|\lambda| \leq 1$ .

**Property 3:**  $\lambda = 1$  is an eigenvalue and  $\mathbf{1} = [1, 1, \dots, 1]^T$  is a corresponding eigenvector. Since  $P$  is row-stochastic, row sum evaluates to 1.

**Property 4:** All eigenvectors of  $P$  are either symmetric or antisymmetric.

$P$  is an  $n \times n$  matrix, which is row symmetric in a cyclic sense, i.e.  $\vec{r}_i = \text{Flip}(\vec{r}_{n-i+1})$ . This follows from row symmetry of individual matrices  $P_1$  and  $P_2$  in the relation  $P = P_2P_1$ . For a row symmetric matrix, with  $n$  an even positive integer, the eigenvectors satisfy,

$$P\vec{v} = \begin{bmatrix} P_u \\ P_l \end{bmatrix} \vec{v} = \lambda\vec{v},$$

where  $P_u$  and  $P_l$  are the upper and lower  $\frac{n}{2}$  rows of the  $P$  matrix. Since  $P$  is row symmetric in the cyclic sense, we can show that there exist  $\vec{w} = \text{Flip}(\vec{v})$  such that  $P\vec{w} = \lambda\vec{w}$ . Thus a symmetric flip of an eigenvector is also a eigenvector of the  $P$  matrix.

When  $n$  is an odd positive integer the same argument follows with the following

partition of  $P = \begin{bmatrix} P_u \\ \vec{r}_{\frac{n+1}{2}} \\ P_l \end{bmatrix}$ , where  $P_u$  and  $P_l$  represents  $\frac{n-1}{2} \times n$  matrices.

**Property 5:** The geometric multiplicity of  $\lambda = 1$  is 2; i.e. there are 2 eigenvectors corresponding to this eigenvalue.

We show that  $P - I$  has kernel dimension (or nullity) of 2. We observe that the diagonal element  $[P - I]_{ii} \neq 0$  for  $i = 2, 3, \dots, n - 1$ . Using row reduction, one can show that all rows are independent except for the first and the last row. This shows that the geometric multiplicity is 2.

**Property 6:** If  $\vec{v}_1 \neq \mathbf{1}$  is an eigenvector of  $\lambda = 1$ , then  $\vec{v}_2 = \mathbf{1} - \vec{v}_1$  is an independent eigenvector of  $\lambda = 1$ .

This follows from 3) and 5). Since the geometric multiplicity of  $P$  for  $\lambda = 1$  is two, there are two independent eigenvectors.

## A.2 Existence of real root for ALM updates

We show the existence of a real root in the interval  $[s_{k-1}, s_{k+1}]$  for the polynomial described in Table 2.1. Using the intermediate value theorem, we demonstrate a sign change at the end points of the polynomial  $r(u)$ .

Case 1:  $2 \leq k \leq K - 1$ . (Here we note that  $r_3 = 0$ )

$$\begin{aligned} r(s_{k-1}) &= r_0 + r_1 s_{k-1} + r_2 s_{k-1}^2 \\ &= \frac{-(s_{k+1} - s_{k-1})^2}{8} \left[ \frac{2}{3} f_X(s_{k-1}) + \frac{1}{3} f_X(s_{k+1}) \right] < 0 \end{aligned}$$

At the right boundary, that is,  $u = s_{k+1}$ ,

$$\begin{aligned} r(s_{k+1}) &= r_0 + r_1 s_{k+1} + r_2 s_{k+1}^2 \\ &= \frac{(s_{k+1} - s_{k-1})^2}{8} \left[ \frac{2}{3} f_X(s_{k+1}) + \frac{1}{3} f_X(s_{k-1}) \right] > 0 \end{aligned}$$

Case 2:  $k = 1$  or  $K$ . We show the proof for  $K = 1$  and explain the modifications necessary for  $k = K$ . The polynomial  $r(u)$  evaluated at  $u = s_{\text{start}}$  and  $u = s_2$  simplifies to :

$$\begin{aligned} r(s_{\text{start}}) &= -\frac{1}{2}(s_2 - s_{\text{start}})^2 \left[ \frac{1}{3} f_X(s_2) + \frac{2}{3} f_X(s_{\text{start}}) \right] < 0 \\ r(s_2) &= \frac{1}{2}(s_2 - s_{\text{start}})^2 \left[ \frac{2}{3} f_X(s_2) + \frac{1}{3} f_X(s_{\text{start}}) \right] > 0. \end{aligned}$$

For the case  $k = K$ , we note that all signs are reversed with respect to the  $k = 1$  case. The modified polynomial still results in sign change at the end points, that is  $r(s_K)r(s_{\text{stop}}) < 0$ .

### A.3 Properties of the limiting matrix $P^*$

The limiting matrix,  $P^*$  has two non-zero columns,  $\vec{c}_{\text{start}}^*$  and  $\vec{c}_{\text{stop}}^*$ , that correspond to the fixed points. Here we show the proof of the property that, the elements of  $\vec{c}_{\text{start}}^* := [c_{\text{start},1}^*, c_{\text{start},2}^*, \dots, c_{\text{start},K+2}^*]$  has a decreasing trend, i.e.  $1 := c_{\text{start},1}^* < c_{\text{start},2}^* < \dots < c_{\text{start},K+2}^* := 0$ . To show this, we use the convex combination property of the ALM update matrix  $P^{(i)}$ . First we illustrate this for the  $K = 3$  case, and later extend for the general case. Recall the structure of  $P^{(i)}$  matrix for  $K = 3$ , given in (2.16). At first, observe that the rows of the operator  $P^{(i)}$  influence the odd elements of the column vectors, which it multiplies. For  $i = 1$ , we note that the elements of first column of  $P^{(1)}$  are  $\vec{c}_{\text{start}}^{(1)} = [1, \theta_1^{(1)}, \theta_2^{(1)}\theta_1^{(1)}, 0, 0]^T$ , which forms a decreasing sequence. Upon applying the update matrix  $P^2$  on this column, we get the new column vector,  $\vec{c}_{\text{start}}^{(2)} = P^{(2)}\vec{c}_{\text{start}}^{(1)} = [1, \theta_1^{(2)} + \bar{\theta}_1^{(2)}\theta_2^{(1)}\theta_1^{(1)}, \dots, 0]^T$ . Each element of  $\vec{c}_{\text{start}}^{(2)}$  is a strict convex combination of elements located in the odd positions of  $\vec{c}_{\text{start}}^{(1)}$ . Since the column  $\vec{c}_{\text{start}}^{(1)}$  has decreasing elements, the structure of the rows of  $P^{(2)}$  preserves the decreasing nature. For instance, the second element  $c_{\text{start},2}^{(2)} = \theta_1^{(2)} + \bar{\theta}_1^{(2)}\theta_2^{(1)}\theta_1^{(1)}$  is a linear combination of 1 and  $\theta_2^{(1)}\theta_1^{(1)}$  – which is strictly less than 1. Further  $c_{\text{start},3}^{(2)}$  is a linear combination having value in the interval  $(c_{\text{start},4}^{(2)}, c_{\text{start},2}^{(2)})$ . In this manner we argue that at  $i = 2$ , the first column of  $P^{(2)}P^{(1)}$  comprise of a strictly decreasing sequence. In the subsequent steps, for  $i > 2$ , the decreasing trend will be preserved since  $P^{(i)}$  has a fixed matrix structure.

For the generic result, we first observe that  $P^{(i)}$  has a structure which consists of rows having either two or three non-zero elements. Building from the  $K = 3$  illustration, the generic case will add rows having three non-zero elements while expanding the dimension. Again since  $P^{(i)}$  is row-stochastic, the nature of the strict convex combination along the rows will preserve the decreasing trend in the column vector,  $\vec{c}_{\text{start}}^{(i-1)}$ . This will show that, as  $i \rightarrow \infty$ , the fixed point  $\vec{c}_{\text{start}}^*$  will have elements in the decreasing order.

## A.4 Proof of the ALM Convergence Theorem 2.1

For ease of exposition, we partition the proof of the convergence theorem into three parts, described below

- I. In the first part, we substantiate the claim that every update coefficient,  $\theta_j^{(i)}$  of  $P^{(i)}$  satisfies the criteria,

$$0 < \theta_j^{(i)} < 1; \quad \text{for all } i \geq 1 \text{ and } j = 1, 2, \dots, K. \quad (\text{A.1})$$

- II. In the second part, we show that the odd index columns of the product sequence  $\left\{ \prod_{i=1}^L P^{(i)} : L \geq 1 \right\}$  converges to the zero column vector as  $L \rightarrow \infty$ . i.e.;

$$\lim_{L \rightarrow \infty} \vec{c}_k^{(L)} = \vec{0}; \quad \text{for } k = 3, 5, \dots, 2m + 1 \quad (\text{A.2})$$

where  $2m + 1$  is the largest odd index less than  $K + 1$ .

- III. In the final part, we establish the existence of a limiting matrix  $P^*$ , which determines the fixed points of the ALM algorithm. This proof will utilize the facts from the previous parts.

### *Proof. Proof of Part I:*

The proof will be dealt in two separate cases. In the *first case*, consider the piecewise slopes,  $m_j = 0$  for  $j = 1, 2, \dots, K + 1$  (this would correspond to a uniform source). From the illustrative example presented in Sec. 2.4.2, we get  $s_j^{(i+1)} = \frac{1}{2}s_{j-1}^{(i)} + \frac{1}{2}s_{j+1}^{(i)}$  (except at the boundary levels  $s_1^{(i)}$  and  $s_K^{(i)}$ ). This corresponds to  $\theta_j^{(i)} = \frac{1}{2}$ . In the *second case*, consider  $m_j \neq 0$  for  $j = 1, 2, \dots, K + 1$ . Also, note the slope condition  $|m_j| \leq m$  (see (2.1)). From the ALM algorithm, we are aware that  $\theta_j^{(i)} \in [0, 1]$ . Consider, the boundary values  $\theta_j^{(i)} = 0$  and  $\theta_j^{(i)} = 1$ . On backtracking, these would correspond to solutions  $s_j^{(i+1)} = s_{j+1}^{(i)}$  and  $s_j^{(i+1)} = s_{j-1}^{(i)}$  respectively. For this to happen, the ALM optimality criteria (refer (2.5)) insists the condition,  $s_{j+1}^{(i)} = s_{j-1}^{(i)}$ , to be satisfied. This, being a trivial case, contradicts the assumption of bounded slope, i.e.,  $|m_j| \leq m$ . Hence we justify the claim that  $\theta_j \in (0, 1)$ .

### **Proof of Part II:**

In this part we show that, all the odd indexed columns of  $\prod_{i=1}^L P^{(i)}$  converge to the zero

vector, as  $L \rightarrow \infty$ . We utilize the mathematical *induction principle*. First, we note that the ALM operator for the initial iteration is  $P^{(1)}$ . Further, at iteration  $i = 2$ , the product operator becomes  $P^{(2)}P^{(1)}$ . The elements of this product can be expressed as the inner product (dot product) between rows and columns of  $P^{(2)}$  and  $P^{(1)}$  respectively. For brevity, let us denote the  $r$ -th row of  $P^{(2)}$  by  $\vec{v}_r^{(2)}$ , and the  $s$ -th column of  $P^{(1)}$  by  $\vec{u}_s^{(1)}$ . If  $w_{r,s}^{(2)}$  stands for the  $(r, s)$  entry of the  $P^{(2)}P^{(1)}$ , then  $w_{r,s}^{(2)} = \langle \vec{v}_r^{(2)}, \vec{u}_s^{(1)} \rangle$ . For later use, the following facts are stated.

- (F1)** Except for the first and last column, the maximum element of each column vector,  $\vec{u}_s^{(1)}$  is *strictly* less than unity. In other words,  $\max_{1 \leq t \leq K+1} [\vec{u}_s^{(1)}]_t < 1$  for every  $s \neq 1$  and  $s \neq K + 2$ . (This follows from Part I of the current theorem)
- (F2)** Each row of  $P^{(2)}$  constitutes a convex combination. That is,  $\mathbf{1}^T \vec{v}_r^{(2)} = 1$  where  $\mathbf{1} = [1, 1, \dots, 1]^T$ . (This is true since  $P^{(2)}$  is a row stochastic matrix)

Using the facts above, we assert that, excluding the boundary rows ( $r = 1$  and  $r = K + 2$ ) and columns ( $s = 1$  and  $s = K + 2$ ), every element of  $[P^{(2)}P^{(1)}]_{r,s} = w_{r,s}^{(2)}$  satisfy,

$$0 \leq w_{r,s}^{(2)} < \max_{1 \leq t \leq K+1} [\vec{u}_s^{(1)}]_t. \quad (\text{A.3})$$

This implies that, all the elements of the column vectors,  $\{\vec{c}_k^{(2)} : k = 3, 5, \dots, 2m + 1\}$ , are *strictly less* than the maximum elements in respective columns of  $P^{(1)}$  (Recall the property of  $P^{(i)}$  matrix in (2.16), which compels the even index columns of  $P^{(2)}P^{(1)}$  to be zero vectors). More precisely, we can represent this transform operator as a contraction, viz.  $w_{r,s}^{(2)} = \alpha_1(r, s) \max_t [\vec{u}_s^{(1)}]_t$ , where  $0 < \alpha_1(r, s) < 1$ .

For  $L = 3$ , the method is identical to what is depicted above. The fact that  $P^{(2)}P^{(1)}$  is row stochastic, enables us to extend the same arguments to the product  $P^{(3)}[P^{(2)}P^{(1)}]$ . For the induction argument, assume that the contraction property holds for  $L = n - 1$ . Now, we argue that the same would hold for  $L = n$ . Let  $w_{r,s}^{(n)} := [\prod_{i=1}^n P^{(i)}]_{r,s}$ . Then, by invoking the induction assumption (as well as the one-step product illustrated earlier), we have

$$w_{r,s}^{(n)} = \left( \prod_{i=1}^{n-1} \alpha_i(r, s) \right) \max_{1 \leq s \leq K+1} w_{r,s}^{(1)},$$

where each contraction coefficient  $0 < \alpha_i(r, s) < 1$ . The above observation shows that,  $\{w_{r,s}^{(n)} : 0 \leq n < \infty\}$  is a monotonically decreasing sequence, with the infimum

$$\begin{aligned} \inf\{w_{r,s}^{(n)}\} &= \lim_{n \rightarrow \infty} \left[ \prod_{i=1}^{n-1} \alpha_i(r, s) \right] \max_{1 \leq s \leq K+1} w_{r,s}^{(1)}, \\ &= \lim_{n \rightarrow \infty} (1 - \varepsilon)^{n-1} \max_{1 \leq s \leq K+1} w_{r,s}^{(1)}, \\ &= 0, \end{aligned}$$

where  $\varepsilon > 0$  because of the slope condition  $\left| \frac{df_X(x)}{dx} \right| \leq m$  (see (2.1); and Part I of proof), and is independent of  $n$ . The above relation holds for all  $w_{r,s}^{(n)}$  such that  $r \in \{2, 3, \dots, K+1\}$  and  $s \in \{3, 5, \dots, 2m+1\}$ . Applying the monotone convergence theorem [110], we see that the odd index columns,  $\{\vec{c}_3, \vec{c}_5, \dots, \vec{c}_{2m+1}\}$ , each decay to the zero vector.

### Proof of Part III:

This part is devoted to assert the existence of the limiting matrix  $P^*$ , and then determine the fixed points corresponding to the ALM quantizer. In the previous part, the convergence of the odd indexed columns of the product matrix was shown. In fact, all columns of  $\prod_{i=1}^L P^{(i)}$ , except the first and last, converge to the zero vector, as the even columns are zero by default. This implies that  $P^* = [\vec{c}_{\text{start}}^*, \mathbf{0}, \dots, \mathbf{0}, \vec{c}_{\text{stop}}^*]$ . The convergence of the remaining two columns can be understood by considering the following properties.

- (P1) The first and last columns, viz.  $\vec{c}_{\text{start}}^*$  and  $\vec{c}_{\text{stop}}^*$ , are non-zero vectors. (This is by the row stochastic property)
- (P2) The elements of  $\vec{c}_{\text{start}}^* := [c_{\text{start},1}^*, c_{\text{start},2}^*, \dots, c_{\text{start},K+2}^*]$  is decreasing in order. (This is due to combination of fact (F1) in Part II, and the structure of the  $P^{(i)}$  matrix – for detailed proof see the Appendix. A.3)
- (P3) The columns,  $\vec{c}_{\text{start}}^*$  and  $\vec{c}_{\text{stop}}^*$  are fixed points and are non-oscillating. (This is since all eigenvalues of  $P^{(i)}$  matrix are non-negative and atmost unity)

Because of the row-stochasticity, the all ones vector,  $\mathbf{1}$  is an eigenvector of  $P^*$  corresponding to  $\lambda = 1$ . From this we see  $\vec{c}_{\text{start}}^* + \vec{c}_{\text{stop}}^* = \mathbf{1}$ . In other words, the vector pair  $(\vec{c}_{\text{start}}^*, \vec{c}_{\text{stop}}^*)$  is *order reversed* (i.e.  $\vec{c}_{\text{start}}^*$  is  $\vec{c}_{\text{stop}}^*$  flipped), and also constitutes the fixed points. Since

$\text{rank}(P^*)$  is two, each of these are independent eigenvectors having  $\lambda = 1$ . If we impose an ordering constraint,  $s_{\text{start}} < s_1 < \dots < s_K < s_{\text{stop}}$ , the vector  $\vec{c}_{\text{stop}}^*$  corresponds to the unique global minimizer of ALM.

*Convergence rate of ALM:* The rate at which the ALM algorithm converge is determined using analysis similar to the Perron-Frobenius theory [23]. However, we have to account for the difference that the stochastic matrix multiplied on the right are different for each iteration. Using eigen decomposition, we can express matrix  $P^{(i)} = V_{(i)}\Lambda_{(i)}V_{(i)}^{-1}$  for  $i \in \mathbb{Z}_+$ . Hence, after  $L$  iterations the product matrix,

$$\prod_{i=1}^L P^{(i)} = \tilde{V}_{(L)}\tilde{\Lambda}_{(L)}\tilde{V}_{(L)}^{-1},$$

where  $\tilde{V}_{(L)} = V_{(L)}V_{(L-1)}\dots V_{(1)}$  and the diagonal matrix  $\tilde{\Lambda}_{(L)} = \Lambda_{(1)}\Lambda_{(2)}\dots\Lambda_{(L)}$ . On assuming  $\Lambda_{(i)}$  has eigenvalues arranged in descending order, using the properties of  $P^{(i)}$  matrix enumerated in Sec. 2.4.2, we have  $[\Lambda_{(i)}]_{jj} = 1$  for  $j = 1, 2$  and  $[\Lambda_{(i)}]_{jj} < 1$  for  $j > 2$ . Since  $\tilde{\Lambda}_{(L)}$  is the product of  $L$  diagonal matrices, its entries are elementwise products. Thus,  $[\tilde{\Lambda}_{(L)}]_{jj} = 1$  for  $j = 1, 2$  and  $[\tilde{\Lambda}_{(L)}]_{jj} < 1$  for  $j > 2$ . With growing  $L$ , all eigenvalues except  $\lambda = 1$  will lead the respective eigenvector components to decay to zero matrix, thus resulting in the convergence the product matrix  $\prod_{i=1}^L P^{(i)}$ . This suggests that the rate of convergence depends on the largest non-unity eigenvalue, which corresponds to the diagonal element  $[\tilde{\Lambda}_{(L)}]_{33}$ . The rate of convergence in  $n$  iterations will be decided by the second largest eigenvalue, or Perron eigenvalue corresponding to each  $P^{(i)}$  matrix. A lower bound on the convergence rate can be expressed in terms of the minimum of these Perron eigenvalues [50, 111, 112], that is  $(\min_{1 \leq i \leq n} [\Lambda_{(i)}]_{33})^n$ . Further, from properties **(P1)**–**(P3)** stated above, as  $L \rightarrow \infty$  the columns of  $\tilde{V}_{(L)}$  converge to the eigenvectors of  $P^*$ , and its fixed points are the first two columns (eigenvectors) of  $\tilde{V}_{(L)}$ . We also discover that the initialization  $\vec{s}^{(0)}$  has no bearing on the fixed points of  $P^*$ .  $\square$

## A.5 Proof of Near-LM Optimality Theorem 2.4

*Proof.* This proof will use mathematical inductive on the iteration index. Recall that,  $q_2^{(i)}$  denotes the LM quantization level at  $i^{\text{th}}$  iteration, similarly  $s_2^{(i)}$  denotes the ALM level. Let,  $b_2^{(i)} := \frac{q_1^{(i)} + q_2^{(i)}}{2}$ ,  $d_2^{(i)} := \frac{s_1^{(i)} + s_2^{(i)}}{2}$  represent the left boundaries and,  $b_3^{(i)} := \frac{q_2^{(i)} + q_3^{(i)}}{2}$ ,  $d_3^{(i)} := \frac{s_2^{(i)} + s_3^{(i)}}{2}$  denote the right boundaries of LM and ALM respectively. Let the quantization

levels be initialized, so that  $q_k^{(0)} = s_k^{(0)}$ , for  $k = 1, 2, \dots, K$ . Refer to (2.20) for the definition of  $\varepsilon_K$ , which is often used in this proof.

**Analysis for iteration 1:** Consider the LM and ALM optimality conditions for  $i = 1$ , in (A.4) and (A.5) respectively. i.e.

$$0 = 2 \int_{b_2^{(0)}}^{b_3^{(0)}} (q_2^{(1)} - x) f_X(x) dx, \quad (\text{A.4})$$

$$0 = 2 \int_{d_2^{(0)}}^{d_3^{(0)}} (s_2^{(1)} - x) f_{\text{app}}(x) dx. \quad (\text{A.5})$$

$$\text{Select, } D^{(1)} := 2 \int_{d_2^{(0)}}^{d_3^{(0)}} (q_2^{(1)} - x) f_{\text{app}}(x) dx. \quad (\text{A.6})$$

Since,  $d_2^{(0)} = b_2^{(0)}$  and  $d_3^{(0)} = b_3^{(0)}$ , we obtain two variants of the expression in (A.6). First variant is obtained using the difference of (A.6) and (A.4), that is,

$$\begin{aligned} |D^{(1)}| &= 2 \left| \int_{d_2^{(0)}}^{d_3^{(0)}} (q_2^{(1)} - x) (f_{\text{app}}(x) - f_X(x)) dx \right|, \\ &\stackrel{\text{(a)}}{=} (q_3^{(0)} - q_1^{(0)}) \left( q_2^{(1)} - \frac{q_1^{(0)} + 2q_2^{(0)} + q_3^{(0)}}{4} \right) \mathcal{O}(\varepsilon_K), \end{aligned} \quad (\text{A.7})$$

where (a) is a result of Taylor series expansion in (2.19) followed by integration. Second variant uses the difference (A.6) – (A.5),

$$\begin{aligned} |D^{(1)}| &= 2 \left| \int_{d_2^{(0)}}^{d_3^{(0)}} (q_2^{(1)} - s_2^{(1)}) f_{\text{app}}(x) dx \right| \\ &\stackrel{\text{(b)}}{=} 2 |q_2^{(1)} - s_2^{(1)}| (d_3^{(0)} - d_2^{(0)}) f_{\text{avg}}(s_2^{(0)}), \end{aligned} \quad (\text{A.8})$$

where  $f_{\text{avg}}(s_2^{(0)}) := f_{\text{app}}\left(\frac{d_2^{(0)} + d_3^{(0)}}{2}\right)$ . Step (b) holds since  $q_2^{(1)}$  and  $s_2^{(1)}$  are constants, and  $f_{\text{app}}(\cdot)$  is an affine function. Using  $(d_3^{(0)} - d_2^{(0)}) = \frac{1}{2}(q_3^{(0)} - q_1^{(0)})$ , and  $f_{\text{avg}}(\cdot) > 0$ , we equate (A.7) and (A.8) to obtain,

$$\begin{aligned} |q_2^{(1)} - s_2^{(1)}| &= \frac{1}{f_{\text{avg}}(s_2^{(0)})} \left( q_2^{(1)} - \frac{q_1^{(0)} + 2q_2^{(0)} + q_3^{(0)}}{4} \right) \mathcal{O}(\varepsilon_K) \\ &\stackrel{\text{(c)}}{=} \frac{1}{f_{\text{avg}}(s_2^{(0)})} \mathcal{O}(\varepsilon_K^{1.5}). \end{aligned} \quad (\text{A.9})$$

Step (c) uses the fact,  $\frac{q_1^{(0)} + 2q_2^{(0)} + q_3^{(0)}}{4} \in (q_1^{(0)}, q_3^{(0)})$ , which leads to the difference  $\left( q_2^{(1)} - \frac{q_1^{(0)} + 2q_2^{(0)} + q_3^{(0)}}{4} \right)$  being upper bounded by  $\max \left\{ \left| q_2^{(1)} - q_1^{(0)} \right|, \left| q_2^{(1)} - q_3^{(0)} \right| \right\} = \mathcal{O}(\varepsilon_K^{0.5})$ .

**Analysis for iteration 2:** Consider the iteration  $i = 2$  along similar steps. We define the mismatch as  $|D^{(2)}| := 2 \left| \int_{d_2^{(1)}}^{d_3^{(1)}} (q_2^{(2)} - x) f_{\text{app}}(x) dx - \int_{b_2^{(1)}}^{b_3^{(1)}} (q_2^{(2)} - x) f_X(x) dx \right|$ . As earlier, we have two variants of  $D^{(2)}$  from the optimality condition of ALM and LM (see (A.4) and (A.5)). Using difference  $|D^{(2)}|$  similar to (A.7),

$$\begin{aligned}
|D^{(2)}| &= 2 \left| \int_{d_2^{(1)}}^{d_3^{(1)}} (q_2^{(2)} - x) f_{\text{app}}(x) dx - \int_{b_2^{(1)}}^{b_3^{(1)}} (q_2^{(2)} - x) f_X(x) dx \right|, \\
&\stackrel{(d)}{\leq} 2 \underbrace{\left| \int_{d_2^{(1)}}^{b_2^{(1)}} (q_2^{(2)} - x) f_{\text{app}}(x) dx + \int_{b_3^{(1)}}^{d_3^{(1)}} (q_2^{(2)} - x) f_{\text{app}}(x) dx \right|}_{D_1^{(2)}} \\
&\quad + 2 \underbrace{\left| \int_{b_2^{(1)}}^{b_3^{(1)}} (q_2^{(2)} - x) (f_{\text{app}}(x) - f_X(x)) dx \right|}_{D_2^{(2)}}. \tag{A.10}
\end{aligned}$$

Here, (d) is due to splitting of the integration limit (or interval)  $[d_2^{(1)}, d_3^{(1)}]$ , into  $[d_2^{(1)}, b_2^{(1)}]$ ,  $[b_2^{(1)}, b_3^{(1)}]$  and  $[b_3^{(1)}, d_3^{(1)}]$  and then applying triangle inequality. By Taylor approximation of  $f_X(x)$ , (2.19), the bound on  $D_2^{(2)}$  is  $\mathcal{O}(\varepsilon_K^2)$ .

For finding a bound on  $D_1^{(2)}$ , we will bound the distance between ALM and LM boundaries. Thus, applying (A.9), we get

$$\begin{aligned}
|d_2^{(1)} - b_2^{(1)}| &= \left| \frac{s_2^{(1)} + s_1^{(1)}}{2} - \frac{q_2^{(1)} + q_1^{(1)}}{2} \right| \\
&= \left[ \frac{1}{2f_{\text{avg}}(s_1^{(0)})} + \frac{1}{2f_{\text{avg}}(s_2^{(0)})} \right] \mathcal{O}(\varepsilon_K^{1.5}). \tag{A.11}
\end{aligned}$$

If  $f_{2,H}^{(1)}$  represent the *harmonic mean* of  $f_{\text{avg}}(s_1^{(0)})$  and  $f_{\text{avg}}(s_2^{(0)})$ , then  $|d_2^{(1)} - b_2^{(1)}| = \frac{1}{2f_{2,H}^{(1)}} \mathcal{O}(\varepsilon_K^{1.5})$ . By replacing the index  $k = 2$  by  $k = 3$ , we get,  $|d_3^{(1)} - b_3^{(1)}| = \frac{1}{2f_{3,H}^{(1)}} \mathcal{O}(\varepsilon_K^{1.5})$ .

Using these distances we bound the first term  $D_1^{(2)}$  in (A.10), as

$$\begin{aligned}
D_1^{(2)} &= \left| \int_{d_2^{(1)}}^{b_2^{(1)}} (q_2^{(2)} - x) f_{\text{app}}(x) dx \right. \\
&\quad \left. - \int_{d_3^{(1)}}^{b_3^{(1)}} (q_2^{(2)} - x) f_{\text{app}}(x) dx \right|, \\
&\stackrel{\text{(e)}}{=} \left| \frac{f_{\text{app}}(b_2^{(1)})}{f_{2,H}^{(1)}} - \frac{f_{\text{app}}(b_3^{(1)})}{f_{3,H}^{(1)}} \right| \mathcal{O}(\varepsilon_K^2), \\
D_1^{(2)} &\stackrel{\text{(f)}}{=} \underbrace{|f_{\text{app}}(b_2^{(1)})| \left( \frac{1}{f_{2,H}^{(1)}} - \frac{1}{f_{3,H}^{(1)}} \right)}_{\Delta_1^{(2)}} - \underbrace{\frac{m_2(b_3^{(1)} - b_2^{(1)})}{f_{3,H}^{(1)}}}_{\Delta_2^{(2)}} \mathcal{O}(\varepsilon_K^2). \tag{A.12}
\end{aligned}$$

Above step, (e) follows from the trapezoid approximation, and (f) is obtained by adding and subtracting  $\frac{f_{\text{app}}(b_2^{(1)})}{f_{3,H}^{(1)}}$ . We have used the definition  $f_{\text{app}}(x) := m_2x + c_2$  for  $x \in [b_2^{(1)}, b_3^{(1)}]$ . Since  $|m_2| \leq m$  by assumption, using the bounds on the boundary levels, i.e.  $|d_k^{(1)} - b_k^{(1)}| = \frac{1}{f_{k,H}^{(1)}}$ , along with definitions of the harmonic mean,

$$|\Delta_1^{(2)}| = \frac{m}{f_{\text{avg}}(s_1^{(0)})} \mathcal{O}(\varepsilon_K^{0.5}) + \frac{m^2}{f_{\text{avg}}(s_1^{(0)}) f_{\text{avg}}(s_3^{(0)})} \mathcal{O}(\varepsilon_K).$$

Similarly,  $|\Delta_2^{(2)}| = \frac{m}{f_{3,H}^{(1)}} \mathcal{O}(\varepsilon_K^{0.5})$ , which results in the bound,

$$\begin{aligned}
D_1^{(2)} &\leq (|\Delta_1^{(2)}| + |\Delta_2^{(2)}|) \mathcal{O}(\varepsilon_K) \\
&= \frac{3m}{f_{2,\min}} \mathcal{O}(\varepsilon_K^{2.5}) + \left( \frac{m}{f_{2,\min}} \right)^2 \mathcal{O}(\varepsilon_K^3), \tag{A.13}
\end{aligned}$$

where  $f_{2,\min} = \min_{i \geq 0} \left\{ \min_{k=\{1,2,3\}} f_{\text{avg}}(s_k^{(i-1)}) \right\}$ . From the upper-bounds of both  $D_1^{(2)}$  and  $D_2^{(2)}$ , the mismatch term  $D^{(2)} := 2D_1^{(2)} + 2D_2^{(2)}$  is  $\mathcal{O}(\varepsilon_K^2)$ , since the second degree term is dominant. Next, using the variant of  $D^{(2)}$ , similar to (A.8), we get  $D^{(2)} = 2|q_2^{(2)} - s_2^{(2)}| (d_3^{(1)} - d_2^{(1)}) f_{\text{avg}}(s_2^{(1)})$ . Since  $|d_3^{(1)} - d_2^{(1)}| = \mathcal{O}(\varepsilon_K^{0.5})$ ,

$$|q_2^{(2)} - s_2^{(2)}| = \frac{D^{(2)}}{2(d_3^{(1)} - d_2^{(1)}) f_{\text{avg}}(s_2^{(1)})} = \frac{1}{f_{\text{avg}}(s_2^{(1)})} \mathcal{O}(\varepsilon_K^{1.5}),$$

which is similar to the upper-bound, (A.9), obtained after first iteration. Next, we repeat this procedure for  $i > 2$ .

**Analysis for iteration  $> 2$ :** Since we are able to bound the mismatch,  $D^{(2)}$ , the above procedure extends to the subsequent values of the iteration index as well. That is,  $D^{(i)} =$

$\mathcal{O}(\varepsilon_K^2)$ , and the distance between the LM and ALM levels is bounded by  $\frac{1}{f_{\text{avg}}(s_2^{(i-1)})} \mathcal{O}(\varepsilon_K^{1.5})$ . This implies that, there is higher accuracy of ALM quantization levels near the maximum of the source probability density. When the probability density value is close to zero, this upper bound on the level difference becomes loose, hence ALM levels deviates from the LM levels.  $\square$

### A.5.1 Proof extension to show order optimality of ALM

Let  $\mathcal{D}_{\text{ALM}} := \sum_{k=1}^K \int_{d_k}^{d_{k+1}} (s_k^* - x)^2 f_X(x) dx$  and  $\mathcal{D}_{\text{LM}} := \sum_{k=1}^K \int_{b_k}^{b_{k+1}} (q_k^* - x)^2 f_X(x) dx$ . Then,

$$\begin{aligned}
\mathcal{D}_{\text{ALM}} &= \sum_{k=1}^K \int_{d_k}^{d_{k+1}} (s_k^* - x)^2 f_X(x) dx \\
&\stackrel{\text{(g)}}{\leq} \sum_{k=1}^K \left[ \int_{d_k}^{d_{k+1}} (s_k^* - q_k^*)^2 f_X(x) dx + \int_{d_k}^{d_{k+1}} (q_k^* - x)^2 f_X(x) dx \right] \\
&\stackrel{\text{(h)}}{\leq} \frac{1}{f_{\min}^2} \mathcal{O}(\varepsilon_K^3) + \mathcal{D}_{\text{LM}} + \sum_{k=1}^K \left[ \int_{d_k}^{d_{k+1}} (q_k^* - x)^2 f_X(x) dx - \int_{b_k}^{b_{k+1}} (q_k^* - x)^2 f_X(x) dx \right] \\
&\stackrel{\text{(i)}}{=} \frac{1}{f_{\min}^2} \mathcal{O}(\varepsilon_K^3) + \mathcal{D}_{\text{LM}} + \sum_{k=1}^K \left[ \int_{d_k}^{b_k} (q_k^* - x)^2 f_X(x) dx + \int_{b_{k+1}}^{d_{k+1}} (q_k^* - x)^2 f_X(x) dx \right]
\end{aligned} \tag{A.14}$$

where  $f_{\min} = \min_{1 \leq l \leq K} f_{l,\min}$ . In the above equation, (g) is due to the triangle inequality, and the first term in (h) is due to (A.9), applied on  $|q_k - s_k|$  for  $1 \leq k \leq K$ . The remaining terms in (h) is obtained by adding and subtracting  $\mathcal{D}_{\text{LM}} := \sum_{k=1}^K \int_{d_k}^{d_{k+1}} (q_k^* - x)^2 f_X(x) dx$ . Finally, (i) is obtained by rearranging the limits of the integral from  $[d_k, d_{k+1}]$  and  $[b_k, b_{k+1}]$  to  $[d_k, b_k]$  and  $[b_{k+1}, d_{k+1}]$  respectively.

Now, by considering the upperbound on  $|q_k - s_k|$  (see (A.9)) and using the equivalent of (A.11)  $|d_k - b_k|$  is upperbounded by  $\frac{1}{f_{\min}} \mathcal{O}(\varepsilon_K^{1.5})$ . Since  $d_k$  and  $b_k$  are bounded, the term  $(q_k^* - x)^2 f_X(x) \leq \varepsilon_K f_{\max}$ , by invoking the definition of  $\varepsilon_K$  (see (2.20)) and  $f_{\max} := \max_{x \in [0,1]} f_X(x)$ . Thus, the two integrals in equation (A.14) can be bounded as

$$\int_{d_k}^{b_k} (q_k^* - x)^2 f_X(x) dx \leq \varepsilon_K f_{\max} \times \frac{\mathcal{O}(\varepsilon_K^{1.5})}{f_{\min}}. \tag{A.15}$$

Next, by substituting the above bound we get,

$$\mathcal{D}_{\text{ALM}} - \mathcal{D}_{\text{LM}} \leq \frac{1}{f_{\min}^2} \mathcal{O}(\varepsilon_K^3) + \frac{2f_{\max}}{f_{\min}} \times \mathcal{O}(K \varepsilon_K^{2.5}), \tag{A.16}$$

where the extra factor  $K$  comes from the summation in (A.14). Finally, since the LM quantizer has an optimality bound given by  $\mathcal{O}(\varepsilon_K)$  [28], ALM quantizer is order optimal to LM quantizer with the constant of proportionality  $\left[ \frac{1}{f_{\min}^2} \mathcal{O}(\varepsilon_K^2) + \frac{2f_{\max}}{f_{\min}} \mathcal{O}(K\varepsilon_K^{1.5}) + 1 \right]$ , which approach 1 as  $K \rightarrow \infty$  at exponential rate.

# Appendix B

## Supplement to Chapter 3

### B.1 Implementation of wavelet density estimation

In our implementation, we have used the Daubechies wavelet for the estimation of unknown probability density function. This choice of the wavelet function was considered as it generalizes well for the class of density functions we consider. A major challenge in the density estimation is in fixing the resolution of the wavelet approximation. For this, we have used an energy based criterion (Mallet, 1998) which is described below.

**Resolution of the wavelet approximation.** The wavelet density estimation can operate at different resolutions to obtain the desired accuracy level. Corresponding to a resolution of  $J$ , we have  $2^J$  approximation coefficients and the same number of wavelet coefficients. Let  $\phi(t)$  represent the scaling function and  $\psi(t)$  represent the wavelet. For limiting the number of coefficient, we have discarded those coefficients which are less than a threshold. Because wavelet coefficients are mostly below the threshold, the approximated density function can be represented as,

$$\hat{f}(x) \approx \sum_{k=1}^{2^J} c_{Jk} \phi(2^J t - k). \quad (\text{B.1})$$

Since the basis functions  $\phi(2^J t - k)$  form an orthonormal set, the energy of the wavelet

approximation is,

$$\begin{aligned} \sum_{k=1}^{2^J} |c_{Jk}|^2 &= \left\langle \sum_{k=1}^{2^J} c_{Jk} \phi(2^J t - k), \sum_{k=1}^{2^J} c_{Jk} \phi(2^J t - k) \right\rangle \\ &\approx \|\widehat{f}(t)\|^2 = \int_0^1 \widehat{f}(t)^2 dt = \mathbb{E} [\widehat{f}(X)]. \end{aligned} \quad (\text{B.2})$$

Using the law of large numbers, we can approximate  $\mathbb{E}[\widehat{f}(X)]$  as  $\frac{1}{n} \sum_{i=1}^n \widehat{f}(X(i))$ . This empirical sample average can be evaluated using the observed signal samples and the approximate density expression in (B.1). Further, we find the smallest value of  $J$  that ensure the equality in expression (B.2) is (approximately) met. This method hence determines the minimum wavelet resolution that meets the energy conservation in the wavelet and signal domains.

The learning procedure to determine the resolution,  $J$  is done using a training set. For stationary signals the procedure is done once at the beginning of the quantizer design.

**Sliding Window Estimation.** To allow online learning of the probability density, we have considered batch-wise estimation using sliding windows. Each window has a length of  $M$ , with an overlap of  $M/2$  previous signal samples. Wavelet density is estimated on the samples from each sliding window, and these estimates are plugged-in for the density evaluations used in the stochastic approximation. For ensuring smooth transition of estimates, we take the weighted average of the wavelet density estimates of the adjacent windows. In other words, this can be stated as,

$$\widehat{f}(x) = \alpha \widehat{f}^{(i)}(x) + (1 - \alpha) \widehat{f}^{(i-1)}(x), \quad (\text{B.3})$$

where  $\widehat{f}^{(i)}(x)$  is the density estimate in the  $i$ -th window and  $\alpha \in [0, 1]$ . In this work, we have used  $\alpha = \frac{1}{2}$ .

## B.2 SANE quantizer design based on mean squared error minimization

Recall the gradient expression (partial derivative) of the MSE distortion with respect to the level  $q_k$  (refer (3.9) in Sec. 3.4.2). The stochastic approximation formulation, will need to estimate the conditional mean,  $\mu_{[q_{k-1}, q_k]}$ , in addition to the quantization bin

probability,  $\mathbb{P}\{X \in (q_{k-1}, q_k]\}$  and the probability density  $f(q_k)$ . Let,  $\widehat{\mu}_k(n)$  denote the empirical estimate of the conditional mean  $\mu_{[q_{k-1}, q_k]}$ .

Following the algorithm akin to MAE minimization, the quantizer level update steps for the MSE distortion are represented as,

$$\widehat{\mu}_k(n+1) = \frac{n \times z_k(n) \widehat{\mu}_k(n) + X(n)}{n \times z_k(n) + 1}, \quad (\text{B.4})$$

$$q_k(n+1) = q_k(n) - a(n) \left\{ 2z_k(n) [q_k(n) - \widehat{\mu}_k(n)] - (q_{k+1}(n) - q_k(n))^2 \widehat{f}(q_k(n)) \right\}, \quad (\text{B.5})$$

$$z_k(n+1) = (1 - b(n))z_k(n) + b(n) \mathbb{1}_{X(n)} \{(q_{k-1}(n), q_k(n+1)]\}, \quad (\text{B.6})$$

where  $\widehat{f}(x)$  is determined using the wavelet density estimation. The mean update in (B.4) is the extra term which is absent in the MAE formulation. It is necessary in MSE formulation since the derivative of the MSE distortion (see (3.9)) involves the mean of the data points in the interval  $[q_{k-1}, q_k]$ . The choice of learning rates  $a(n)$  and  $b(n)$  in (B.5)-(B.6), is to be made using the convergence criteria for stochastic approximation, explained in Sec. 3.4.1. An alternate method for the estimation of the conditional mean,  $\mu_{[q_{k-1}, q_k]}$  is by using the density estimate  $\widehat{f}(x)$ . This is given by,

$$\widehat{\mu}_k(n+1) = \frac{\int_{q_{k-1}}^{q_k} x \widehat{f}(x) dx}{\int_{q_{k-1}}^{q_k} \widehat{f}(x) dx}. \quad (\text{B.7})$$

Among the two methods, the empirical estimate in equation (B.4) is preferred over the estimate in (B.7), because of the reduced computational complexity involved.

### B.3 Roots corresponding to AEQ

We show that the polynomial equation  $p(u) = p_0 + p_1 u + p_2 u^2 + p_3 u^3 = 0$ , with coefficients as listed in Table. 3.2, has atleast one real root in the interval  $[q_{k-1}, q_{k+1}]$ . Recall that  $q_{k-1}$  and  $q_{k+1}$  represents the left and right nearest neighbors of the quantization level  $q_k$ . We show the above fact using the intermediate value theorem, that is,  $p(u) = 0$  if

$p(q_{k-1})p(q_{k+1}) < 0$ . Evaluating the polynomial at the end points of the interval we get,

$$\begin{aligned}
p(q_{k-1}) &= p_0 + p_1q_{k-1} + p_2q_{k-1}^2 + p_3q_{k-1}^3 \\
&= -c_kq_{k-1}^2 - c_kq_{k+1}^2 + 2c_kq_{k-1}q_{k+1} - m_kq_{k-1}^3 - m_kq_{k-1}q_{k+1}^2 + 2m_kq_{k-1}^2q_{k+1} \\
&= -(c_k + m_kq_{k-1})(q_{k+1} - q_{k-1})^2 \\
&= -f_X(q_{k-1})(q_{k+1} - q_{k-1})^2 \\
&< 0,
\end{aligned} \tag{B.8}$$

and

$$\begin{aligned}
p(q_{k+1}) &= p_0 + p_1q_{k-1} + p_2q_{k-1}^2 + p_3q_{k-1}^3 \\
&= c_kq_{k-1}^2 + c_kq_{k+1}^2 - 2c_kq_{k-1}q_{k+1} + \frac{2}{3}m_kq_{k-1}^3 + \frac{1}{3}m_kq_{k+1}^3 - m_kq_{k-1}^2q_{k+1} \\
&= -p(q_{k-1}) + \frac{1}{3}m_k(q_{k+1}^3 - q_{k-1}^3) - m_kq_{k+1}q_{k-1}(q_{k+1} - q_{k-1}) \\
&= \left(\frac{2}{3}f_X(q_{k-1}) + \frac{1}{3}f_X(q_{k+1})\right)(q_{k+1} - q_{k-1})^2 \\
&> 0.
\end{aligned} \tag{B.9}$$

From the above two inequalities we observe that the product  $p(q_{k-1})p(q_{k+1})$  is always negative and hence there always exist a root of  $p(u) = 0$  in the interval  $[q_{k-1}, q_{k+1}]$ .

## B.4 Proof that AEQ optimality condition results in a positive derivative

In this section we show that the AEQ optimality condition defined by the polynomial  $p(u)$ , in Table. 3.2 has a positive slope. The proof for the same follows from the convexity of the cost function (3.13). The derivative of the polynomial  $p(q_k)$  with respect to  $q_k$  is given as,

$$\begin{aligned}
\frac{dp(q_k)}{dq_k} &= 2(q_{k+1} - q_{k-1}) \left( m_k \left( \frac{q_{k+1} + q_{k-1}}{2} \right) + c_k \right) \\
&\quad - 2m_k(q_{k+1} - q_k)^2 \\
&= 2(q_{k+1} - q_{k-1})f_{\text{app}} \left( \frac{q_{k+1} + q_{k-1}}{2} \right) \\
&\quad - 2m_k(q_{k+1} - q_k)^2
\end{aligned} \tag{B.10}$$

We consider the following three cases -  $m_k = 0$ ,  $m_k < 0$  and  $m_k > 0$ . In the first and second case, we see that the derivative is positive since  $f_{\text{app}} \left( \frac{q_{k+1} + q_{k-1}}{2} \right) > 0$ . When

$m_k > 0$ , we use the fact that, the optimal solution  $q_k$  is closer to  $q_{k+1}$  than  $q_{k-1}$ . In other words, we get the condition  $q_{k+1} - q_k \leq q_k - q_{k-1}$ . Using the above fact, we rewrite (B.10) as,

$$\begin{aligned} \frac{dp(q_k)}{dq_k} &\geq 2(q_{k+1} - q_{k-1}) \left[ m_k \left( \frac{2q_k - q_{k+1} + q_{k-1}}{2} \right) + c_k \right] \\ &\geq 2(q_{k+1} - q_{k-1}) f_{\text{app}}(q_{k-1}) > 0 \end{aligned} \tag{B.11}$$



# Appendix C

## Supplement to Chapter 4

### C.1 General IID datasets in federated aggregation

**Theorem C.1.** *Let  $D$  client devices – Device-1, Device-2, ..., Device- $D$  be trained on i.i.d. datasets  $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_D$  to obtain the  $K$ -level quantizers  $\vec{q}_1, \vec{q}_2, \dots, \vec{q}_D$  respectively, where  $\vec{q}_d := [q_{d,1}, \dots, q_{d,K}]$ . Also, let the test mean squared errors of these devices be denoted as  $MSE_d$  for  $d \in \{1, 2, \dots, D\}$ . Then, the proportional weighted federated scheme has a test mean squared error,  $MSE_F(\pi_{prop})$ , which is less than the weighted average MSE of the individual devices. That is,*

$$MSE_F(\pi_{prop}) \leq \frac{N_1}{N} MSE_1 + \frac{N_2}{N} MSE_2 + \dots + \frac{N_D}{N} MSE_D, \quad (\text{C.1})$$

where  $N_d = \text{card}(\mathcal{D}_d)$  for  $d \in \{1, \dots, D\}$  and  $N = \sum_{d=1}^D N_d$ .

*Proof.* Because the devices are assumed to have i.i.d. datasets  $\mathcal{D}_d; d \in \{1, \dots, D\}$ , the test mean-squared errors are computed with the common density  $f(x)$ , which is assumed a bounded support in the interval  $[0, 1]$ . The quantizer corresponding to the proportional weighted federated aggregation scheme will have the levels,

$$\begin{aligned} q_{F,1} &:= \alpha_1 q_{1,1} + \alpha_2 q_{2,1} + \dots + \alpha_D q_{D,1}, \\ q_{F,2} &:= \alpha_1 q_{1,2} + \alpha_2 q_{2,2} + \dots + \alpha_D q_{D,2}, \\ &\vdots \\ q_{F,K} &:= \alpha_1 q_{1,K} + \alpha_2 q_{2,K} + \dots + \alpha_D q_{D,K}, \end{aligned}$$

where  $\alpha_d = \frac{N_d}{N}$  for  $d \in \{1, \dots, D\}$  and  $N = \sum_{d=1}^D N_d$ . Then, the test mean square error at the devices are,

$$\begin{aligned} \text{MSE}_d &= \mathbb{E} [(Q_d(X) - X)^2], \quad \text{for } d \in \{1, 2, \dots, D\} \\ &= \int_0^{t_{d,1}} (q_{d,1} - x)^2 f(x) dx + \sum_{k=1}^{K-2} \int_{t_{d,k}}^{t_{d,k+1}} (q_{d,k+1} - x)^2 f(x) dx + \int_{t_{d,K-1}}^1 (q_{d,K} - x)^2 f(x) dx, \end{aligned}$$

where  $t_{d,j} := \frac{q_{d,j} + q_{d,j+1}}{2}$  for  $j \in \{1, 2, \dots, K-1\}$  represents the (nearest neighbor) decision boundary of Device-d, between levels  $q_{d,j}$  and  $q_{d,j+1}$ . We note that the boundary is at the mid-point of these levels. Similar to the above formulation, the test mean-squared error of the federated quantizer,  $Q_F(\cdot)$  is,

$$\begin{aligned} \text{MSE}_F(\pi_{\text{prop}}) &= \mathbb{E} [(Q_F(X) - X)^2] \\ &= \int_0^{t_{F,1}} (q_{F,1} - x)^2 f(x) dx + \sum_{k=1}^{K-2} \int_{t_{F,k}}^{t_{F,k+1}} (q_{F,k+1} - x)^2 f(x) dx \\ &\quad + \int_{t_{F,K-1}}^1 (q_{F,K} - x)^2 f(x) dx. \end{aligned} \tag{C.2}$$

The boundaries corresponding to the quantizer obtained by federated aggregation,  $t_{F,j}$  can be written as a convex combination of the respective boundaries obtained at the devices, i.e.

$$t_{F,j} = \alpha_1 t_{1,j} + \alpha_2 t_{2,j} + \dots + \alpha_D t_{D,j},$$

for  $j \in \{1, 2, \dots, K-1\}$ . The required result can be shown if the mean-squared error function is convex in all variables. To show this, we consider a generic test MSE function for  $K$  levels, given by

$$\begin{aligned} \text{MSE}(q_1, q_2, \dots, q_K; t_1, t_2, \dots, t_{K-1}) &:= \int_0^{t_1} (q_1 - x)^2 f(x) dx + \sum_{k=1}^{K-2} \int_{t_k}^{t_{k+1}} (q_{k+1} - x)^2 f(x) dx \\ &\quad + \int_{t_{K-1}}^1 (q_K - x)^2 f(x) dx, \end{aligned} \tag{C.3}$$

where the ordering,  $q_1 < q_2 < \dots < q_K$  and  $t_1 < t_2 < \dots < t_{K-1}$  are assumed, and each  $t_k; k \in \{1, 2, \dots, K-1\}$  lies in the interval  $[0, 1]$ . Since the quadratic function  $(q_k - x)^2$  is convex in  $q_k; k \in \{1, \dots, K\}$ , for a fixed  $\vec{t} = [t_1, t_2, \dots, t_{K-1}]$ , we can show that  $\text{MSE}(q_1, q_2, \dots, q_K; \vec{t})$  is also convex. This is because the mean-squared error is a weighted average of the convex quadratic functions in  $q_1, q_2, \dots, q_K$ . Further, these

weights are positive (since the density function  $f(x) > 0$ ), thus resulting in a positive weighted quadratic function that is always convex. Next we verify the convexity w.r.t the vector variable  $\vec{t}$  for a fixed quantizer level vector  $\vec{q} = [q_1, \dots, q_K]$ . The gradient w.r.t variable  $\vec{t}$  is given by,

$$\nabla_{\vec{t}} \text{MSE}(\vec{q}; \vec{t}) := \begin{bmatrix} \frac{\partial}{\partial t_1} \text{MSE}(\vec{q}; \vec{t}) \\ \frac{\partial}{\partial t_2} \text{MSE}(\vec{q}; \vec{t}) \\ \vdots \\ \frac{\partial}{\partial t_{K-1}} \text{MSE}(\vec{q}; \vec{t}) \end{bmatrix} = \begin{bmatrix} (q_1 - q_2)(q_1 + q_2 - 2t_1)f(t_1) \\ (q_2 - q_3)(q_2 + q_3 - 2t_2)f(t_2) \\ \vdots \\ (q_{K-1} - q_K)(q_{K-1} + q_K - 2t_{K-1})f(t_{K-1}) \end{bmatrix}.$$

It is noted that the gradient becomes zero at  $t_k = t_k^*$  where  $t_k^* = \frac{1}{2}(q_k + q_{k+1})$  for  $k \in \{1, 2, \dots, K-1\}$ . Next, we compute the Hessian matrix,  $H$  w.r.t the variable vector  $\vec{t}$ . The diagonal entries of this matrix is given by,

$$\frac{\partial^2}{\partial t_k^2} \text{MSE}(\vec{q}; \vec{t}) = f'(t_k)(q_k - q_{k+1})(q_k + q_{k-1} - 2t_k) + 2f(t_k)(q_{k+1} - q_k),$$

which is positive at  $t_k^*$ , as the first term vanishes, and the density function  $f(x) > 0$ . Since the elements of gradient vector has only a single variable, the off-diagonal elements of the Hessian,  $H_{ij} = \frac{\partial^2}{\partial t_j \partial t_i} \text{MSE}(\vec{q}; \vec{t})$  are zeros. Thus, the Hessian matrix is a positive definite matrix, at the values  $t_k^* = \frac{1}{2}(q_k + q_{k+1})$ . From this we conclude that, corresponding to the nearest-neighbor (mid-point) boundaries of the quantization levels, the function  $\text{MSE}(\vec{q}; \vec{t}^*)$ , with  $\vec{t}^* = [t_1^*, \dots, t_{K-1}^*]$ , is convex in all the variables. Thus, the federated quantizer test error,

$$\begin{aligned} \text{MSE}_F &= \text{MSE}(\alpha_1 \vec{q}_1 + \alpha_2 \vec{q}_2 + \dots + \alpha_D \vec{q}_D; \alpha_1 \vec{t}_1^* + \alpha_2 \vec{t}_2^* + \dots + \alpha_D \vec{t}_D^*) \\ &\leq \alpha_1 \text{MSE}(\vec{q}_1; \vec{t}_1^*) + \alpha_2 \text{MSE}(\vec{q}_2; \vec{t}_2^*) + \dots + \alpha_D \text{MSE}(\vec{q}_D; \vec{t}_D^*) \\ &= \alpha_1 \text{MSE}_1 + \alpha_2 \text{MSE}_2 + \dots + \alpha_D \text{MSE}_D. \end{aligned} \tag{C.4}$$

Thus the desired inequality is obtained.  $\square$

*Remark:* The above proof shows the convexity of the mean-squared error distortion, by assuming that the density function  $f(x)$  is continuous and differentiable on its bounded support  $[0, 1]$ . A variant of the convexity analysis for generic (convex) distortion measures have been shown for the class of log-concave density functions (with possibly unbounded support) by [30], to prove that the Lloyd algorithm converges to a unique minimizer (fixed point).

## C.2 General non-identical but independent datasets in federated aggregation

**Theorem C.2.** *Let  $D$  client devices – Device-1, Device-2, ..., Device- $D$  be trained on non- identical but independent datasets  $\mathcal{D}'_1, \mathcal{D}'_2, \dots, \mathcal{D}'_D$  to obtain the  $K$ -level quantizers  $\vec{q}'_1, \vec{q}'_2, \dots, \vec{q}'_D$  respectively, where  $\vec{q}'_d := [q'_{d,1}, \dots, q'_{d,K}]$ . Also, let the test mean squared errors of these devices be denoted as  $MSE'_d$  for  $d \in \{1, 2, \dots, D\}$ . Then, the proportional weighted federated scheme has a test mean squared error,  $MSE_F(\pi_{prop})$ , which is less than the weighted average MSE of the individual devices. That is,*

$$MSE_F(\pi_{prop}) \leq \alpha'_1 MSE'_1 + \alpha'_2 MSE'_2 + \dots + \alpha'_D MSE'_D, \quad (C.5)$$

where  $\alpha'_d$  are the parameters of the mixture distribution of the pooled dataset,  $\mathcal{D}'_1 \cup \mathcal{D}'_2 \cup \dots \cup \mathcal{D}'_D$ , such that  $\sum_{d=1}^D \alpha'_d = 1$ .

*Proof.* By using the notation  $\alpha_i = \frac{N_i}{N}$  for  $i = 1, 2, \dots, D$ , where  $N = \sum_{i=1}^D N_i$ , we represent the quantizer levels of the federated aggregation scheme  $\pi_{prop}$  as,

$$q_{F,k} = \alpha_1 q'_{1,k} + \alpha_2 q'_{2,k} + \dots + \alpha_D q'_{D,k} \quad \text{for } k = 1, 2, \dots, K.$$

Therefore, the test mean squared error of the resultant quantizer,  $\vec{q}_F := \{q_{F,1}, \dots, q_{F,K}\}$ , can be computed using using the definition,  $MSE_F(\pi_{prop}) := \mathbb{E}[(Q_F(X) - X)^2]$ , where the expectation is over the mixture distribution  $g(x) = \alpha'_1 f_1(x) + \alpha'_2 f_2(x) + \dots + \alpha'_D f_D(x)$ . Here  $f_d(x)$  for  $d = 1, 2, \dots, D$ , represent the PDF of the dataset  $\mathcal{D}'_d$  at Device- $d$ . i.e.,

$$\begin{aligned} MSE_F(\pi_{prop}) &= \int_0^{t_{F,1}} (q_{F,1} - x)^2 g(x) dx + \sum_{k=1}^{K-2} \int_{t_{F,k}}^{t_{F,k+1}} (q_{F,k+1} - x)^2 g(x) dx \\ &\quad + \int_{t_{F,K-1}}^1 (q_{F,K} - x)^2 g(x) dx. \end{aligned} \quad (C.6)$$

where  $t_{F,j} = \alpha_1 t_{1,j} + \alpha_2 t_{2,j} + \dots + \alpha_D t_{D,j}$  with the definition of the the  $j^{\text{th}}$  quantizer level boundary as  $t_{d,j} := \frac{q'_{d,j} + q'_{d,j+1}}{2}$ . By invoking the convexity property of the mean squared error function,  $MSE_F(\cdot)$  as proved in (C.4), we get,

$$\begin{aligned} MSE_F(\pi_{prop}) &= \sum_{d=1}^D \alpha_d \left[ \int_0^{t_{d,1}} (q_{d,1} - x)^2 g(x) dx + \sum_{k=1}^{K-2} \int_{t_{d,k}}^{t_{d,k+1}} (q_{d,k+1} - x)^2 g(x) dx \right. \\ &\quad \left. + \int_{t_{d,K-1}}^1 (q_{d,K} - x)^2 g(x) dx \right] \end{aligned} \quad (C.7)$$

Now, upon using the notation,

$$\text{MSE}_{i,j} := \left[ \int_0^{t_{i,1}} (q_{i,1} - x)^2 f_j(x) dx + \sum_{k=1}^{K-2} \int_{t_{i,k}}^{t_{d,i+1}} (q_{d,i+1} - x)^2 f_j(x) dx + \int_{t_{i,K-1}}^1 (q_{i,K} - x)^2 f_j(x) dx \right], \quad \text{for } (i,j) \in \{1, 2, \dots, D\}^2,$$

we can rewrite equation (C.7) by expanding the mixture probability density function,  $g(x)$ , and rearranging the resultant terms. That is,

$$\text{MSE}_F(\pi_{\text{prop}}) = \sum_{d=1}^D \alpha'_d \left[ \sum_{i=1}^D \alpha_i \text{MSE}_{i,d} \right]. \quad (\text{C.8})$$

Further we can upperbound the above mean squared error, using the fact  $\sum_{i=1}^D \alpha_i \text{MSE}_{i,d} \leq \max_{i \in \{1, \dots, D\}} \text{MSE}_{i,d}$ , where  $\sum_{i=1}^D \alpha_i = 1$ . Therefore,

$$\text{MSE}_F(\pi_{\text{prop}}) \leq \alpha'_1 \max_i \text{MSE}_{i,1} + \alpha'_2 \max_i \text{MSE}_{i,2} + \dots + \alpha'_D \max_i \text{MSE}_{i,D}. \quad (\text{C.9})$$

Finally, by jointly minimizing the right hand side of the above upperbound with respect to the collective quantization level vectors  $\{\vec{q}'_1, \vec{q}'_2, \dots, \vec{q}'_D\}$ , we can tighten the upperbound as,

$$\begin{aligned} \text{MSE}_F(\pi_{\text{prop}}) &\leq \min_{\{\vec{q}'_1, \vec{q}'_2, \dots, \vec{q}'_D\}} \sum_{d=1}^D \alpha'_d \max_i \text{MSE}_{i,d}, \\ &\leq \alpha'_1 \text{MSE}'_1 + \alpha'_2 \text{MSE}'_2 + \dots + \alpha'_D \text{MSE}'_D. \end{aligned} \quad (\text{C.10})$$

□

*Remark:* In practice the parameters of the mixture probability density will be unknown and hence need to be estimated from the data. A naïve estimate for  $\alpha'_d$  will be the fraction of training data contributed by Device- $d$ , i.e.  $\alpha'_d = \alpha_d = \frac{N_d}{N}$ . Using this naïve estimate we see that the federated aggregation using  $\pi_{\text{prop}}$  is efficient in computing a global quantizer.



# Appendix D

## Supplement to Chapter 5

### D.1 Order optimality of envelope approximation

*Proof.* : We show the desired result for  $q = 2$  and  $q = 1$  separately. For notational simplicity we will consider the following analysis with respect to client device 1 with observed signal being  $f_1(t)$  and the corresponding Fourier coefficients as  $\{a_1[k] : k \in \mathbb{Z}\}$ . The envelope approximation and its Fourier coefficients are represented as  $\widehat{f}_{1,\text{env}}(t)$  and  $\{b_1[k] : -L \leq k \leq L\}$  respectively.

**Case 1:** For  $q = 2$ , the optimal envelope approximation error is computed from the envelope constraint  $\vec{b}_1^T \Phi(t) \geq f_1(t)$ . Since  $f_1(t) = \sum_{k \in \mathbb{Z}} a_1[k] \exp(j2\pi kt)$  for  $t \in [0, 1]$ , we have the bound

$$\begin{aligned} \sum_{|k| \leq L} b_1[k] \exp(j2\pi kt) &\geq \sum_{k \in \mathbb{Z}} a_1[k] \exp(j2\pi kt) \\ \Rightarrow \sum_{|k| \leq L} (b_1[k] - a_1[k]) \exp(j2\pi kt) &\geq \sum_{|k| > L} a_1[k] \exp(j2\pi kt), \end{aligned}$$

that leads to the inequality  $\sum_{|k| \leq L} |b_1[k] - a_1[k]|^2 \geq \sum_{|k| > L} |a_1[k]|^2$ , based on the Parseval's Theorem. Now, the approximation error of the optimal  $\mathcal{L}_2$ -norm envelope approximation signal,

$$\begin{aligned} \text{SA}_2 &:= \min_{\vec{b}_1} \sum_{|k| \leq L} |b_1[k] - a_1[k]|^2 + \sum_{|k| > L} |a_1[k]|^2 \\ &\geq \min_{\vec{b}_1} 2 \sum_{|k| > L} |a_1[k]|^2. \end{aligned} \tag{D.1}$$

Further, using Fact 5.1 for  $p$ -times differentiable signal class with  $a_1[k]$ ,  $k \in \mathbb{Z}$  satisfying the lower bound [108] for the series  $\sum_{|k|>L} \frac{1}{|k|^{2p}}$ ,

$$\text{SA}_2 \geq \frac{2}{2p-1} \frac{1}{(L+1)^{2p-1}}. \quad (\text{D.2})$$

Using (5.12), the approximation error of the naïve envelope can be upper bounded as [108],

$$\text{SA}'_2 \leq \frac{2}{2p-1} \frac{1}{L^{2p-1}}. \quad (\text{D.3})$$

Since  $\text{SA}_2$  refers to the approximation error of the optimal envelope,  $\text{SA}_2 \leq \text{SA}'_2$ , and we get the inequality,

$$1 \leq \frac{\text{SA}'_2}{\text{SA}_2} \leq \left[ \frac{2}{2p-1} \frac{1}{L^{2p-1}} \right] \bigg/ \left[ \frac{2}{2p-1} \frac{1}{(L+1)^{2p-1}} \right], \quad (\text{D.4})$$

or  $1 \leq \frac{\text{SA}'_2}{\text{SA}_2} \leq \left(1 + \frac{1}{L}\right)^{2p-1}$ . The right hand upperbound approaches 1 in the limit  $L \rightarrow \infty$ . This shows that the approximation errors of the optimal envelope and the naïve envelope are order optimal for the  $\mathcal{L}_2$  cost.

**Case 2:** The optimality of the naïve envelopes for the  $\text{SA}_1$  distance holds for a signal class with the following additional properties:

- (i) the Fourier coefficients  $a_1[k] \geq 0$ ,
- (ii)  $f_1(t)$  is real and even, that is  $a_1[k] = a_1[-k]$ .

From these symmetry assumptions, it follows that  $b_1[k] = b_1[-k]$ . Restricted to this signal class, the  $\text{SA}_1$  envelope approximation is re-stated as:

$$\begin{aligned} \text{SA}_1 &:= \min_{b_1[k], |k| \leq L} b_1[0] - a_1[0], \quad \text{subject to} \\ b_1[0] + 2 \sum_{1 \leq k \leq L} b_1[k] \cos(2\pi kt) &\geq \sum_{k \in \mathbb{Z}} a_1[k] e^{j2\pi kt} \end{aligned} \quad (\text{D.5})$$

The above optimization can be shown to result in  $b_{1,\text{opt}}[0] - a_1[0] = \sum_{|k|>L} a_1[k]$ , using the following argument. Consider the envelope constraint in (D.5), by rearranging the terms,

$$b_1[0] - a_1[0] \geq 2 \sum_{1 \leq k \leq L} (a_1[k] - b_1[k]) \cos(2\pi kt) + 2 \sum_{|k|>L} a_1[k] \cos(2\pi kt). \quad (\text{D.6})$$

Since we assumed  $a_1[k] > 0$  for all  $k$  and  $\{b_1[k] : 1 \leq k \leq L\}$  is the optimization variable, the right hand side term of the above inequality is maximum at  $t = 0$  and  $t = 1$ . Thus,

by choosing  $b_1[k] = a_1[k]$  for  $1 \leq k \leq L$ , we get  $\text{SA}_1 := b_{1,\text{opt}}[0] - a_1[0] \geq 2 \sum_{k>L} a[k]$ . For the considered signal class, the naïve approximation error,  $\text{SA}'_1$  is shown to satisfy  $\text{SA}'_1 = C_0 = \|f_1 - f_{1,\text{proj}}\|_\infty \leq 2 \sum_{k>L} a_1[k]$  using (5.12). Since the optimal envelope has the minimum approximation error,  $\text{SA}_1 \leq \text{SA}'_1$ . In summary,

$$2 \sum_{k>L} a_1[k] \leq \text{SA}_1 \leq \text{SA}'_1 \leq 2 \sum_{k>L} a_1[k], \quad (\text{D.7})$$

which shows that  $\text{SA}_1 = \text{SA}'_1$  for the signal class considered.

*Remark:* From (5.6), the  $L_\infty$  norm envelope distortion is expressed as an upper bound in terms of the Fourier coefficients. For the  $p$ -times differentiable signal class, this results in a distortion,  $\text{SA}'_\infty = \mathcal{O}\left(\frac{1}{L^{p-1}}\right)$ , on the naïve approximation scheme, as expressed in Table D.1. We observe that an analytical expression for  $\text{SA}_\infty$  is challenging to determine without additional information on the signal class.  $\square$

Table D.1: Bounds on the approximation errors

$\mathcal{L}_q$ norm	$\text{SA}_q$	$\text{SA}'_q$
$q = 1$	$b_1[0] - a_1[0]$	$\sum_{ k >L}  a_1[k] $
$q = 2$	$\frac{2}{2p-1} \frac{1}{(L+1)^{2p-1}}$	$\frac{2}{2p-1} \frac{1}{L^{2p-1}}$
$q = \infty$	—	$\frac{2}{p-1} \frac{1}{L^{p-1}}$

## D.2 Communication vs accuracy tradeoff in envelope CDF estimation

*Proof.* Using the result from Grimmett and Stirzaker, for any  $\epsilon > 0$ ;

$$\begin{aligned} F_{X_{\text{env}}}(x) &\geq F_X(x - \epsilon) - \mathbb{P}(X_{\text{env}} - X > \epsilon) \\ \Rightarrow F_X(x) - F_{X_{\text{env}}}(x) &\leq F_X(x) - F_X(x - \epsilon) + \mathbb{P}(X_{\text{env}} - X > \epsilon) \end{aligned} \quad (\text{D.8})$$

In the limit  $\varepsilon \rightarrow 0$ ,

$$F_X(x) - F_{X_{\text{env}}}(x) \leq \varepsilon f_X(x) + \mathbb{P}(X_{\text{env}} - X > \varepsilon) \quad (\text{D.9})$$

Since  $X_{\text{env}}$  is the  $(2L + 1)$  coefficient envelope approximation,

$$\|X - X_{\text{env}}\|_2^2 = \sum_{|k| \leq L} |B_{\text{env}}[k] - A[k]|^2 + \sum_{|k| > L} |A[k]|^2 \quad (\text{D.10})$$

From the envelope constraint,  $X_{\text{env}}(t) \geq X(t)$ , we have

$$\begin{aligned} \sum_{|k| \leq L} B_{\text{env}}[k] e^{j2\pi kt} &\geq \sum_{k \in \mathbb{Z}} A[k] e^{j2\pi kt} \\ \Rightarrow \sum_{|k| \leq L} (B_{\text{env}}[k] - A[k]) e^{j2\pi kt} &\geq \sum_{|k| > L} A[k] e^{j2\pi kt} \\ \Rightarrow \sum_{|k| \leq L} |B_{\text{env}}[k] - A[k]|^2 &\geq \sum_{|k| > L} |A[k]|^2 \end{aligned} \quad (\text{D.11})$$

Applying the above inequality in (D.10),

$$\begin{aligned} \|X_{\text{env}} - X\|^2 &\leq 2 \sum_{|k| \leq L} |B_{\text{env}}[k] - A[k]|^2 \\ &\leq 2 \sum_{|k| \leq L} |B_{\text{naïve}}[k] - A[k]|^2 \\ &\leq \frac{2}{2p-1} \frac{1}{L^{2p-1}} \quad (\text{Ref: [108]}). \end{aligned} \quad (\text{D.12})$$

We now use the Chebyshev's inequality in (D.8), i.e.

$$\begin{aligned} \mathbb{P}(X_{\text{env}} - X > \varepsilon) &\leq \frac{1}{\varepsilon^2} \mathbb{E}(\|X_{\text{env}} - X\|^2) \\ &\leq \frac{1}{\varepsilon^2} \frac{2}{2p-1} \frac{1}{L^{2p-1}} \end{aligned} \quad (\text{D.13})$$

The difference of the CDF's,

$$F_X(x) - F_{X_{\text{env}}}(x) \leq \varepsilon f_X(x) + \frac{2}{\varepsilon^2(2p-1)} \frac{1}{L^{2p-1}}. \quad (\text{D.14})$$

Assuming  $f_X(x) \leq f_{X,\text{max}}$ , we get  $F_X(x) - F_{X_{\text{env}}}(x) \leq \varepsilon f_X(x) + \frac{2}{\varepsilon^2(2p-1)} \frac{1}{L^{2p-1}}$ . On minimizing the right-hand side w.r.t  $\varepsilon$ , we get,

$$F_X(x) - F_{X_{\text{env}}}(x) \leq \left(4^{\frac{1}{3}} + 2 \times 4^{-\frac{2}{3}}\right) \frac{f_{X,\text{max}}^{2/3}}{(2p-1)^{1/3}} \frac{1}{L^{\frac{2p-1}{3}}} \quad (\text{D.15})$$

□

### D.3 Effect of subsampling on the CDF

Consider the following optimization,

$$\begin{aligned} \min_{\widehat{f}_{\text{env}}} \quad & \|f - \widehat{f}_{\text{env}}\|_2^2 \\ \text{subject to} \quad & f_{\text{env}}(t) \geq f(t), \quad t \in [0, 1]. \end{aligned} \quad (\text{D.16})$$

The same optimization can be re-written in terms of the Fourier Coefficients as,

$$\begin{aligned} \vec{B}_{\text{env}} := \arg \min_{\vec{B}} \quad & \|\vec{A} - \vec{B}\|_2^2 \quad \text{subject to} \\ & \vec{B}^T \Phi(t) \geq f(t), \quad t \in [0, 1], \end{aligned} \quad (\text{D.17})$$

where  $\Phi(t)$  denotes the bandlimited Fourier basis with  $(2L + 1)$  basis elements. Suppose, only discrete samples of the signal was available, the optimization problem to solve would be,

$$\begin{aligned} \vec{B}_{\text{app},n} := \arg \min_{\vec{B}} \quad & \|\vec{A} - \vec{B}\|_2^2 \quad \text{subject to} \\ & \vec{B} \Phi(t_n) \geq f(t_n) \quad \text{for } t_n \in \left\{0, \frac{1}{n}, \frac{2}{n}, \dots, 1\right\} \end{aligned} \quad (\text{D.18})$$

Since (D.18) has only a subset of constraints compared to (D.17),

$$\|\vec{A} - \vec{B}_{\text{app},n}\|_2^2 \leq \|\vec{A} - \vec{B}_{\text{env}}\|_2^2. \quad (\text{D.19})$$

Using the method in [107], we can construct a suboptimal envelope signal with coefficients

$$\vec{B}_{\text{subopt}} \text{ such that } B_{\text{subopt}}[k] = \begin{cases} B_{\text{app},n}[k], & k \neq 0 \\ B_{\text{app},n}[0] + \frac{c+c'}{n}, & k = 0 \end{cases}, \text{ where the constants arise}$$

from the assumptions  $|f'(t)| \leq c$  and  $f'_{\text{app},n}(t) \leq c'$ , described in (5.15). Since  $\vec{B}_{\text{subopt}}$  satisfies the envelope constraint in (D.17), we can write

$$\|\vec{A} - \vec{B}_{\text{app},n}\|_2^2 \leq \|\vec{A} - \vec{B}_{\text{env}}\|_2^2 \leq \|\vec{A} - \vec{B}_{\text{subopt}}\|_2^2. \quad (\text{D.20})$$

Similar to the earlier case, we wish to find an upperbound on CDF difference, viz. for all  $\varepsilon > 0$

$$\left| F_X(x) - F_{X_{\text{app},n}}(x) \right| \leq \varepsilon f_X(x) + \frac{1}{\varepsilon^2} \mathbb{E}(\|X - X_{\text{app},n}\|^2). \quad (\text{D.21})$$

Consider,

$$\begin{aligned}
\|X - X_{\text{app},n}\|^2 &\leq \|X - X_{\text{env}}\|^2 + \|X_{\text{env}} - X_{\text{app},n}\|^2 \quad (\text{Traingle Inequality}) \\
&= \sum_{|k|\leq L} |B_{\text{env}}[k] - A[k]|^2 + \sum_{|k|\leq L} |A[k]|^2 + \sum_{|k|\leq L} |B_{\text{env}}[k] - B_{\text{app},n}[k]|^2 \\
&\leq 3 \sum_{|k|\leq L} |B_{\text{env}}[k] - A[k]|^2 + \sum_{|k|\leq L} |B_{\text{app},n}[k] - A[k]|^2 \\
&\leq 4 \sum_{|k|\leq L} |B_{\text{subopt}}[k] - A[k]|^2 \\
&\leq 4 \sum_{|k|\leq L} |B_{\text{app},n}[k] - A[k]|^2 + 8(B_{\text{app},n}[0] - A[0]) \frac{c+c'}{n} + o(1/n) \\
&\leq \frac{4}{2p-1} \frac{1}{L^{2p-1}} + 8(B_{\text{app},n}[0] - A[0]) \frac{c+c'}{n} + o(1/n) \tag{D.22}
\end{aligned}$$

Thus,

$$\left| F_X(x) - F_{X_{\text{app},n}}(x) \right| \leq \varepsilon f_X(x) + \frac{1}{\varepsilon^2} \left\{ \frac{4}{2p-1} \frac{1}{L^{2p-1}} + 8\mu_{\text{app},n} \frac{c+c'}{n} + o(1/n) \right\}, \tag{D.23}$$

where  $\mu_{\text{app},n} = \mathbb{E}(B_{\text{app},n}[0] - A[0])$ . On assuming  $f_X(x) \leq f_{X,\max}$  and minimizing the upper bound with respect to  $\varepsilon$ , we get

$$\left| F_X(x) - F_{X_{\text{app},n}}(x) \right| \leq \left[ \left( 2^{\frac{1}{3}} + 2^{-\frac{2}{3}} \right) f_{X,\max}^{2/3} \right] \left\{ \frac{4}{2p-1} \frac{1}{L^{2p-1}} + 8\mu_{\text{app},n} \frac{c+c'}{n} + o(1/n) \right\}^{1/3}. \tag{D.24}$$

# Bibliography

- [1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.
- [2] “Internet of Things - number of connected devices worldwide 2015-2025,” <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>, accessed: 2021-09-12.
- [3] “TV whitespace protection contour near New York,” [https://spectrum.iconectiv.com/main/home/contour\\_vis.shtml](https://spectrum.iconectiv.com/main/home/contour_vis.shtml), accessed: 2015-10-04.
- [4] “Thomas Edison National Historical Park,” <https://home.nps.gov/edis/learn/education/index.htm>, accessed: 2013-10-04.
- [5] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of Things for smart cities,” *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb 2014.
- [6] “Ericsson mobility report June 2021,” <https://www.ericsson.com/en/mobility-report>, accessed: 2021-09-19.
- [7] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020. [Online]. Available: <https://doi.org/10.1109/MSP.2020.2975749>
- [8] P. Kairouz and H. B. McMahan, “Advances and open problems in federated learning,” *Foundations and Trends in Machine Learning*, vol. 14, no. 1, pp. –, 2021. [Online]. Available: <http://dx.doi.org/10.1561/22000000083>
- [9] White House Report, “Consumer data privacy in a networked world: A framework for protecting privacy and promoting innovation in the global digital economy,” *Journal of Privacy and Confidentiality*, vol. 4, 2013.

- 
- [10] T. Van Dijk and G. De Croon, “How do neural networks see depth in single images?” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2183–2191.
- [11] A. Kumar, S. Goyal, and M. Varma, “Resource-efficient machine learning in 2 KB RAM for the Internet of Things,” in *International Conference on Machine Learning*, 2017, pp. 1935–1944.
- [12] C. Elkan, “Using the triangle inequality to accelerate  $k$ -means,” in *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ser. ICML’03. AAAI Press, 2003, pp. 147–153. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3041838.3041857>
- [13] A. Choromanska and C. Monteleoni, “Online clustering with experts,” in *Artificial Intelligence and Statistics*, 2012, pp. 227–235.
- [14] T. Linder, “Learning-theoretic methods in vector quantization,” in *Principles of nonparametric learning*. Springer, 2002, pp. 163–210.
- [15] V. Anavangot and A. Kumar, “A novel Approximate Lloyd-Max quantizer and its analysis,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019.
- [16] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. New York, NY, USA: Wiley-Interscience, 2006.
- [17] P. A. Chou, T. Lookabaugh, and R. M. Gray, “Entropy-constrained vector quantization,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 1, 1989.
- [18] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.
- [19] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, “Edge intelligence: Paving the last mile of artificial intelligence with edge computing,” *Proceedings of the IEEE*, vol. 107, no. 8, 2019.

- 
- [20] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Toward an intelligent edge: Wireless communication meets machine learning," *IEEE Communications Magazine*, vol. 58, no. 1, 2020.
- [21] R. Marculescu, D. Marculescu, and U. Ogras, "Edge AI: Systems design and ML for IoT data analytics," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '20. ACM, 2020.
- [22] S. U. Pillai, T. Suel, and S. Cha, "The Perron-Frobenius theorem: some of its applications," *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 62–75, March 2005.
- [23] R. G. Gallager, *Stochastic processes*. Cambridge University Press, 2013.
- [24] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, March 1982.
- [25] J. Max, "Quantizing for minimum distortion," *IRE Transactions on Information Theory*, vol. 6, no. 1, pp. 7–12, March 1960.
- [26] D. Sharma, "Design of absolutely optimal quantizers for a wide class of distortion measures," *IEEE Transactions on Information Theory*, vol. 24, no. 6, pp. 693–702, November 1978.
- [27] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, Jan 1980.
- [28] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, Oct 1998.
- [29] J. Ziv, "On universal quantization," *IEEE Transactions on Information Theory*, vol. 31, no. 3, 1985.
- [30] J. Kieffer, "Exponential rate of convergence for Lloyd's method I," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 205–210, March 1982.

- 
- [31] M. Sabin and R. Gray, “Global convergence and empirical consistency of the generalized Lloyd algorithm,” *IEEE Transactions on Information Theory*, vol. 32, no. 2, pp. 148–155, Mar 1986.
- [32] X. Wu, “On convergence of Lloyd’s method I,” *IEEE Transactions on Information Theory*, vol. 38, no. 1, pp. 171–174, Jan 1992.
- [33] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, 1967.
- [34] T. Kohonen, “Learning vector quantization,” in *Self-organizing maps*. Springer, 1995, pp. 175–189.
- [35] A. K. Jain, R. P. W. Duin, and Jianchang Mao, “Statistical pattern recognition: A review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, Jan 2000.
- [36] D. Pollard, *Convergence of Stochastic Processes*. Springer Series in Statistics, 1984.
- [37] ———, “Quantization and the method of  $k$ -means,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 199–205, March 1982.
- [38] A. Ortega and M. Vetterli, “Adaptive scalar quantization without side information,” *IEEE Transactions on Image Processing*, 1997.
- [39] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, “Federated optimization: Distributed machine learning for on-device intelligence,” *arXiv preprint:1610.02527*, 2016.
- [40] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, “Federated learning in mobile edge networks: A comprehensive survey,” *IEEE Comm. Surveys & Tutorials*, 2020.
- [41] W.-T. Chang and R. Tandon, “Communication efficient federated learning over multiple access channels,” *arXiv preprint:2001.08737*, 2020.

- [42] M. M. Amiri and D. Gündüz, “Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air,” *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, 2020. [Online]. Available: <https://doi.org/10.1109/TSP.2020.2981904>
- [43] P. Warden and D. Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O’Reilly Media, Incorporated, 2019.
- [44] Y. Guo, “A survey on methods and theories of quantized neural networks,” *arXiv preprint 1808.04752*, 2018.
- [45] A. Chatterjee and L. R. Varshney, “Towards optimal quantization of neural networks,” in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 1162–1166.
- [46] M. H. Protter, B. Charles Jr *et al.*, *Intermediate calculus*. Springer Science & Business Media, 2012.
- [47] P. Bullen, *Handbook of Means and Their Inequalities*. Springer Series in Mathematics and Its Applications, 2003, vol. 560.
- [48] T. Linder, G. Lugosi, and K. Zeger, “Rates of convergence in the source coding theorem, in empirical quantizer design, and in universal lossy source coding,” *IEEE Transactions on Information Theory*, vol. 40, no. 6, pp. 1728–1740, 1994.
- [49] B. Hajek and M. Raginsky, “Statistical learning theory,” 2021. [Online]. Available: <http://maxim.ece.illinois.edu/teaching/SLT/>
- [50] D. Shah *et al.*, “Gossip algorithms,” *Foundations and Trends® in Networking*, vol. 3, no. 1, pp. 1–125, 2009.
- [51] M. Huang, “Stochastic approximation for consensus: A new approach via ergodic backward products,” *IEEE Transactions on Automatic Control*, vol. 57, no. 12, pp. 2994–3008, Dec 2012.
- [52] R. Furberg, J. Brinton, M. Keating, and A. Ortiz, “Crowd-sourced Fitbit datasets 03.12.2016-05.12.2016,” May 2016. [Online]. Available: <https://doi.org/10.5281/zenodo.53894>

- [53] V. Weaver. Reading RAPL energy measurements from Linux. (2020, October 20). [Online]. Available: <http://web.eece.maine.edu/~vweaver/projects/rapl/>
- [54] Welch, “A technique for high-performance data compression,” *Computer*, vol. 17, no. 6, pp. 8–19, 1984.
- [55] M. J. Wainwright, *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press, 2019, vol. 48.
- [56] A. Dvoretzky, J. Kiefer, and J. Wolfowitz, “Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator,” *Ann. Math. Statist.*, vol. 27, no. 3, pp. 642–669, 09 1956.
- [57] K. Harrison, S. M. Mishra, and A. Sahai, “How much white-space capacity is there?” in *2010 IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN)*, 2010.
- [58] A. Kumar, A. Karandikar, G. Naik, M. Khaturia, S. Saha, M. Arora, and J. Singh, “Toward enabling broadband for a billion plus population with TV white spaces,” *IEEE Communications Magazine*, vol. 54, no. 7, 2016.
- [59] G. B. Giannakis, V. Kekatos, N. Gatsis, S. Kim, H. Zhu, and B. F. Wollenberg, “Monitoring and optimization for power grids: A signal processing perspective,” *IEEE Signal Processing Magazine*, vol. 30, no. 5, 2013.
- [60] J. Naus, G. Spaargaren, B. van Vliet, and H. van der Horst, “Smart grids, information flows and emerging domestic energy practices,” *Energy Policy*, vol. 68, pp. 436–446, 2014.
- [61] S. Ghosh, G. Naik, A. Kumar, and A. Karandikar, “OpenPAWS: An open source PAWS and UHF TV White Space database implementation for India,” in *2015 Twenty First National Conference on Communications (NCC)*, 2015.
- [62] *Microsoft Whitespaces*, 2015 (accessed April 23, 2020), <http://whitespaces.microsoftspectrum.com>.
- [63] M. Alizadeh, X. Li, Z. Wang, A. Scaglione, and R. Melton, “Demand-side management in the smart grid: Information processing for the power switch,” *IEEE Signal Processing Magazine*, vol. 29, no. 5, 2012.

- [64] P. M. Mammen, H. Kumar, K. Ramamritham, and H. Rashid, “Want to reduce energy consumption, whom should we call?” in *Proceedings of the Ninth International Conference on Future Energy Systems*, 2018.
- [65] G. Naik, S. Singhal, A. Kumar, and A. Karandikar, “Quantitative assessment of TV white space in India,” in *2014 Twentieth National Conference on Communications (NCC)*, 2014, pp. 1–6.
- [66] G. Maheshwari and A. Kumar, “Optimal quantization of TV white space regions for a broadcast based geolocation database,” in *2016 24th European Signal Processing Conference (EUSIPCO)*, Aug 2016, pp. 418–422.
- [67] H. Robbins and S. Monro, “A stochastic approximation method,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, 1951. [Online]. Available: <https://doi.org/10.1214/aoms/1177729586>
- [68] V. S. Borkar, *Stochastic approximation: a dynamical systems viewpoint*. Springer, 2009, vol. 48.
- [69] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977. [Online]. Available: <http://www.jstor.org/stable/2984875>
- [70] J.-C. Fort and G. Pages, “On the a.s. convergence of the Kohonen algorithm with a general neighborhood function,” *The Annals of Applied Probability*, vol. 5, no. 4, pp. 1177–1216, 11 1995. [Online]. Available: <https://doi.org/10.1214/aoap/1177004611>
- [71] H. Kushner and G. G. Yin, *Stochastic approximation and recursive algorithms and applications*. Springer Science & Business Media, 2003, vol. 35.
- [72] P. Netrapalli, “Stochastic gradient descent and its variants in machine learning,” *Journal of the Indian Institute of Science*, vol. 99, 02 2019.
- [73] M. Kristan and A. Leonardis, “Multivariate online kernel density estimation,” in *Computer Vision Winter Workshop*, 2010, pp. 77–86.

- [74] E. S. García-Treviño and J. A. Barria, “Online wavelet-based density estimation for non-stationary streaming data,” *Computational statistics & data analysis*, vol. 56, no. 2, pp. 327–344, 2012.
- [75] S. Mallet, “A wavelet tour of signal processing,” 1999.
- [76] C. Chesneau, I. Dewan, and H. Doosti, “Wavelet linear density estimation for associated stratified size-biased sample,” *Journal of Nonparametric Statistics*, vol. 24, no. 2, pp. 429–445, 2012.
- [77] G. Dalal, B. Szörényi, G. Thoppe, and S. Mannor, “Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning,” *Proceedings of Machine Learning Research*, vol. 75, 2018.
- [78] V. S. Borkar and S. Pattathil, “Concentration bounds for two time scale stochastic approximation,” in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2018, pp. 504–511.
- [79] P. Ramírez and B. Vidakovic, “Wavelet density estimation for stratified size-biased sample,” *Journal of Statistical Planning and Inference*, vol. 140, no. 2, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378375809002298>
- [80] A. Ortega and M. Vetterli, “Adaptive quantization without side information,” in *Proceedings of 1st International Conference on Image Processing*, vol. 3. IEEE, 1994.
- [81] E. Süli and D. F. Mayers, *An Introduction to Numerical Analysis*. Cambridge University Press, 2003.
- [82] R. Laroia and N. Farvardin, “Trellis-based scalar-vector quantizer for memoryless sources,” *IEEE Transactions on Information Theory*, vol. 40, no. 3, pp. 860–870, May 1994.
- [83] V. Schellekens and L. Jacques, “Quantized compressive  $k$ -means,” *IEEE Signal Processing Letters*, vol. 25, no. 8, pp. 1211–1215, 2018.

- [84] T. C. Aysal, M. Coates, and M. Rabbat, “Distributed average consensus using probabilistic quantization,” in *2007 IEEE/SP 14th Workshop on Statistical Signal Processing*, Aug 2007, pp. 640–644.
- [85] A. T. Suresh, F. X. Yu, S. Kumar, and H. B. McMahan, “Distributed mean estimation with limited communication,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017.
- [86] M. Mohri, G. Sivek, and A. T. Suresh, “Agnostic federated learning,” in *International Conference on Machine Learning*, 2019, pp. 4615–4625.
- [87] K. A. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, “Towards federated learning at scale: System design,” in *Proceedings of Machine Learning and Systems 2019, MLSys 2019, Stanford, CA, USA, March 31 - April 2, 2019*, A. Talwalkar, V. Smith, and M. Zaharia, Eds. mlsys.org, 2019. [Online]. Available: <https://proceedings.mlsys.org/book/271.pdf>
- [88] J. Xu and F. Wang, “Federated learning for healthcare informatics,” *arXiv preprint arXiv:1911.06270*, 2019.
- [89] S. Graf and H. Luschgy, *Foundations of quantization for probability distributions*. Springer, 2007.
- [90] K. Crammer, R. Gilad-Bachrach, A. Navot, and N. Tishby, “Margin analysis of the lvq algorithm,” in *Advances in neural information processing systems*, 2003, pp. 479–486.
- [91] D. Arthur and S. Vassilvitskii, “ $k$ -means++: The advantages of careful seeding,” in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA ’07. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

- [92] P. L. Bartlett, T. Linder, and G. Lugosi, “The minimax distortion redundancy in empirical quantizer design,” *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1802–1813, Sep. 1998.
- [93] Y. LeCun and C. Cortes, “MNIST handwritten digit database,” <http://yann.lecun.com/exdb/mnist/>, 2010.
- [94] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016a. [Online]. Available: <https://arxiv.org/abs/1610.05492>
- [95] “Google AI Blog federated analytics: Collaborative data science without data collection,” <https://ai.googleblog.com/2020/05/federated-analytics-collaborative-data.html>, 2020, accessed: 2021-05-19.
- [96] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, “A performance evaluation of federated learning algorithms,” in *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning, DIDL@Middleware 2018, Rennes, France, December 10, 2018*. ACM, 2018, pp. 1–8. [Online]. Available: <https://doi.org/10.1145/3286490.3286559>
- [97] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, “QSGD: communication-efficient SGD via gradient quantization and encoding,” in *Advances in Neural Information Processing Systems*, 2017, pp. 1709–1720. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/6c340f25839e6acdc73414517203f5f0-Abstract.html>
- [98] A. Feraudo, P. Yadav, V. Safronov, D. A. Popescu, R. Mortier, S. Wang, P. Bellavista, and J. Crowcroft, “CoLearn: Enabling federated learning in MUD-compliant IoT edge networks,” in *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, 2020, pp. 25–30.
- [99] T. Li, M. Sanjabi, A. Beirami, and V. Smith, “Fair resource allocation in federated learning,” in *8th International Conference on Learning Representations, ICLR*

- 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020, p. 1. [Online]. Available: <https://openreview.net/forum?id=ByexELSYDr>
- [100] V. Smith, C. Chiang, M. Sanjabi, and A. S. Talwalkar, “Federated multi-task learning,” in *Advances in Neural Information Processing Systems*, 2017, pp. 4424–4434. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/6211080fa89981f66b1a0c9d55c61d0f-Abstract.html>
- [101] E. White, “Statistical learning for unimpaired flow prediction in ungauged basins,” Ph.D. dissertation, University of California, Davis, 2020.
- [102] M. Stockman, M. Awad, and R. Khanna, “Asymmetrical and lower bounded support vector regression for power estimation,” in *2011 International Conference on Energy Aware Computing*, 2011, pp. 1–6.
- [103] Y. L. Tun, K. Thar, C. M. Thwal, and C. S. Hong, “Federated learning based energy demand prediction with clustered aggregation,” in *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2021, pp. 164–167.
- [104] K. Bountrogiannis, G. Tzagkarakis, and P. Tsakalides, “Data-driven kernel-based probabilistic SAX for time series dimensionality reduction,” in *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 2343–2347.
- [105] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck, and S. Srikanteswara, “Energy demand prediction with federated learning for electric vehicle networks,” *arXiv:1909.00907*, 2019.
- [106] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*, 3rd ed. Academic Press, 2008.
- [107] A. Kumar, “Optimal envelope approximation in fourier basis with applications in TV white space,” *CoRR*, vol. abs/1706.00900, 2017. [Online]. Available: <http://arxiv.org/abs/1706.00900>
- [108] R. Bhatia, *Fourier Series*, 2nd ed. Hindustan Book Agency, 1993.

- 
- [109] L. Balzano and R. D. Nowak, “Blind calibration of sensor networks,” in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks, IPSN 2007, Cambridge, Massachusetts, USA, April 25-27, 2007*, T. F. Abdelzaher, L. J. Guibas, and M. Welsh, Eds. ACM, 2007, pp. 79–88. [Online]. Available: <https://doi.org/10.1145/1236360.1236372>
- [110] W. Rudin, *Real and complex analysis*. Tata McGraw-Hill Education, 2006.
- [111] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, “Gossip algorithms for distributed signal processing,” *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov 2010.
- [112] P.-Y. Chevalier, “Convergent products of stochastic matrices : Algorithms and complexity,” PhD dissertation, UCL - SST/ICTM - Institute of Information and Communication Technologies, Electronics and Applied Mathematics, 2018.

# List of Publications

## Journal Publications

- J1 **V. Anavangot** and A. Kumar, “Signal Source Distribution Approximation to Speedup Scalar Quantizer Design,” in *IEEE Transactions on Signal Processing*, doi: 10.1109/TSP.2021.3125602.
- J2 **V. Anavangot** and A. Kumar, “SANE: A Stochastic Approximation based Overpredictive Quantizer Design,” *submitted to IEEE Signal Processing Letters*
- J3 **V. Anavangot** and A. Kumar, “Overpredictive Signal Analytics in Federated Learning: Algorithms and Tradeoff Analysis”, *submitted to Elsevier Signal Processing*

## Conference Proceedings

- C1 **V. Anavangot** and A. Kumar, “A Novel Approximate Lloyd-max Quantizer and Its Analysis,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, May 2019, pp. 5107-5111, doi: 10.1109/ICASSP.2019.8682396.
- C2 **V. Anavangot** and A. Kumar, “Algorithms for Overpredictive Signal Analytics in Federated Learning,” *28th European Signal Processing Conference (EUSIPCO)*, Amsterdam, Netherlands, Jan 2021, pp. 1502-1506, doi: 10.23919/Eusipco47968.2020.9287390.

## Preprints/Under Preparation

- C3 **V. Anavangot** and A. Kumar, “Distributed Quantizer Design and Tradeoffs in Federated Learning”, in Preparation