Unsupervised Learning for Parametric Optimization in Wireless Networks

Rasoul Nikbakht Silab

TESI DOCTORAL UPF / year 2020

THESIS SUPERVISOR
Prof. Angel Lozano

Dept. of Information and Communication Technologies

upf. Universitat
Pompeu Fabra
Barcelona

“ *To my parents for all the sacrifices they have made.* ”

Rasoul

Acknowledgments

In the summer of 2016, I checked my Ph.D. admission letter with a crappy borrowed phone on the rooftop of a military base while doing my military service. It was a challenging and complicated path to leave that rooftop and land at Barcelona airport. I want to thank my family and friends, who made it possible.

Angel was a great supervisor for my Ph.D. in all aspects; I would like to thank him for his professional mentorship. Also, I express my gratitude to Anders, our collaborator, who guided us in the machine learning world. Lastly, I would like to thank Giovanni, who generously and meticulously reviewed our published and unpublished works.

My personal life in Barcelona has had many ups and downs, but I have made many friends and had a wonderful time in Barcelona. I want to thank you all.

A close friend taught me how to use the MANUAL. Once again, thank you.

Finally, it might take a while to have another opportunity, so I say it laud now: I express my deepest gratitude to my loving parents and kind sisters, who made this journey possible, which translates in my language to ”Nokaram, Eradat Var.”

Abstract

This thesis studies parametric optimization in cellular and cell-free networks, exploring data-based and expert-based paradigms. Power allocation and power control, which adjust the transmit power to meet different fairness criteria such as max-min or max-product, are crucial tasks in wireless communications that fall into the parametric optimization category. The state-of-the-art approaches for power control and power allocation often demand huge computational costs and are not suitable for real-time applications. To address this issue, we develop a general-purpose unsupervised-learning approach for solving parametric optimizations; and extend the well-known fractional power control algorithm.

In the data-based paradigm, we create an unsupervised learning framework that defines a custom neural network (NN), incorporating expert knowledge to the NN loss function to solve the power control and power allocation problems. In this approach, a feedforward NN is trained by repeatedly sampling the parameter space, but, rather than solving the associated optimization problem completely, a single step is taken along the gradient of the objective function. The resulting method is applicable for both convex and non-convex optimization problems. It offers two-to-three orders of magnitude speedup in the power control and power allocation problems compared to a convex solver—whenever appliable.

In the expert-driven paradigm, we investigate the extension of fractional power control to cell-free networks. The resulting closed-form solution can be evaluated for uplink and downlink effortlessly and reaches an (almost) optimum solution in the uplink case.

In both paradigms, we place a particular focus on large scale gains—the amount of attenuation experienced by the local-average received power. The slow-varying nature of the large-scale gains relaxes the need for a frequent update of the solutions in both the data-driven and expert-driven paradigms, enabling real-time application for both methods.

Resum

Aqueta tesis estudia l'optimització paramètrica a les xarxes *cel.lulars* i xarxes *cell-free*, explotant els paradigmes basats en dades i basats en experts. L'assignació i control de la potencia, que ajusten la potencia de transmissió per complir amb diferents criteris d'equitat com *max-min* o *max-product*, son tasques crucials en les telecomunicacions inalambriques pertanyents a la categoria d'optimització paramètrica. Les técniques d'última generació per al control i assignació de la potencia solen exigir enormes costos computacionals i no son adequats per aplicacions en temps real. Per abordar aquesta qüestió, desenvolupem una tècnica de propòsit general utilitzant aprenentatge no supervisat per resoldre optimitzacions paramètriques; i al mateix temps ampliem el reconegut algoritme de control de potencia fraccionada.

En el paradigma basat en dades, creem un marc d'aprenentatge no supervisat que defineix una xarxa neuronal (NN, sigles de Neural Network en Anglès) especifica, incorporant coneixements experts a la funció de cost de la NN per resoldre els problemes de control i assignació de potència. Dins d'aquest enfocament, s'entrena una NN de tipus feedforward mitjanccant el mostreig repetit en l'espaï de paràmetres, però, en lloc de resoldre completament el problema d'optimització associat, es pren un sol pas en la direcció del gradient de la funció objectiu. El mètode resultant és aplicable tant als problemes d'optimització convexos com no convexos. Això ofereix una acceleració de dos a tres ordres de magnitud en els problemes de control i assignació de potencia en comparació amb un algoritme de resolució convexa—sempre que sigui aplicable.

En el paradigma dirigit per experts, investiguem l'extensió del control de potencia fraccionada a les xarxes sense celules. La solució tancada resultant pot ser avaluada per a l'enllacc de pujada i el de baixada sense esforcc i assoleix una sol·lució (gairebé) òptima en el cas de l'enllacc de pujada.

En ambdós paradigmes, ens centrem especialment en els guanys a gran escala—la quantitat d'atenuació que experimenta la potencia mitja local rebuda. La naturalesa de variació lenta dels guanys a gran escala relaxa la necessitat d’una actualització freqüent de les solucions tant en el paradigma basat en dades com en el basat en experts, permetent d’aquesta manera l’ús dels dos mètodes en aplicacions en temps real.

Resumen

Esta tesis estudia la optimización paramétrica en las redes *celulares* y redes *cell-free*, explorando los paradigmas basados en datos y en expertos. La asignación y el control de la potencia, que ajustan la potencia de transmisión para cumplir con diferentes criterios de equidad como *max-min* o *max-product*, son tareas cruciales en las comunicaciones inalámbricas pertenecientes a la categoría de optimización paramétrica. Los enfoques más modernos de control y asignación de la potencia suelen exigir enormes costes computacionales y no son adecuados para aplicaciones en tiempo real. Para abordar esta cuestión, desarrollamos un enfoque de aprendizaje no supervisado de propósito general que resuelve las optimizaciones paramétricas y a su vez ampliamos el reconocido algoritmo de control de potencia fraccionada.

En el paradigma basado en datos, creamos un marco de aprendizaje no supervisado que define una red neuronal (NN, por sus siglas en inglés) específica, incorporando conocimiento de expertos a la función de coste de la NN para resolver los problemas de control y asignación de potencia. Dentro de este enfoque, se entrena una NN de tipo *feedforward* mediante el muestreo repetido del espacio de parámetros, pero, en lugar de resolver completamente el problema de optimización asociado, se toma un solo paso en la direccion del gradiente de la función objetivo. El método resultante es aplicable tanto a los problemas de optimización convexos como no convexos. Ofrece una aceleración de dos a tres órdenes de magnitud en los problemas de control y asignación de potencia, en comparación con un algoritmo de resolución convexo—siempre que sea aplicable.

Dentro del paradigma dirigido por expertos, investigamos la extensión del control de potencia fraccionada a las redes *cell-free*. La solución de forma cerrada resultante puede ser evaluada para el enlace *uplink* y el *downlink* sin esfuerzo y alcanza una solución (casi) óptima en el caso del enlace *uplink*.

En ambos paradigmas, nos centramos especialmente en las *large-scale gains*—la cantidad de atenuación que experimenta la potencia media local recibida. La naturaleza lenta y variable de las ganancias a gran escala re-

laja la necesidad de una actualización frecuente de las soluciones tanto en el paradigma basado en datos como en el basado en expertos, permitiendo el uso de ambos métodos en aplicaciones en tiempo real.

Contents

List of Figures

List of Tables

Chapter 1

Motivation and Outline

> *Everything should be made as simple as possible, but not simpler.*
>
> Albert Einstein

1.1 Motivation

In 2015, a deep-learning-based technique reached human level accuracy on image classification tasks. Inspired by this astonishing success, many wireless engineers (including the author) jumped into the machine learning bandwagon, hoping to repeat the same success in wireless communications. Initially, we thought that every aspect of wireless networks could be revolutionized using the data-driven paradigm. Soon after, we realized that that might not be the case because wireless networks and their subsystems are well-optimized already using relatively simple algorithms. Nonetheless, it became evident that data-driven approaches can significantly improve *current* wireless networks when used alongside the traditional ones (model-based learning). For instance, in the deep learning case, such approaches could:

- Improve the training time substantially, embedding the structure of the problem in the training process, e.g., the loss function of the neural network (NN).

- Be interpretable, meaning that a domain expert can make sense of the different components of the resulting solution.

In *future* wireless networks, data-driven optimization should be at the heart of the design procedure, unifying different blocks such as channel estimation, detection schemes, and decoding. An example of such an approach is end-to-end learning, where communication is treated as a learning problem using autoencoders.

This thesis has adopted the view of model-based learning for power control and power allocation problems in cellular and cell-free wireless networks—dense infrastructures of access points (APs) each potentially communicating with all users. Specifically, we have embedded the signal-to-interference-plus-noise ratio (SINR) in the loss function of the NN. Such an NN, after undergoing the training stage, would output close-to-optimum power coefficients.

Another avenue explored in this thesis is that of expert-driven approaches. Specifically, well-known fractional power control has been extended to cell-free networks, which has led to a simple yet close-to-optimum power allocation. Interestingly, in uplink power control, both the unsupervised-learning approach and fractional power control have led to similar solutions, indicating that whenever applicable, expert knowledge should be embraced.

1.2 Outline and contributions

Parametric optimization in wireless networks, chiefly for power control and power allocation, is the major focus of this thesis. Two paradigms, namely data-driven and expert-driven, are explored. With both paradigms, a great emphasis is placed on the large-scale gains (the amount of attenuation experienced by the local-average received power) for the following reasons:

- Large-scale gains vary slowly, leading to robust solutions without a need for frequent updates.

- Large-scale gains of close-by users are correlated, potentially enabling large-scale prediction for new coming users based on their locations.

The extension of the latter aspect leads to the concept of a power map, a map of local-average received power, which is explored in more detail in Chapter 7.

We propose an unsupervised-learning approach as a data-driven solution. This turns out to be a great tool for solving parametric optimization problems. The core idea of this approach is presented in Chapter 3, and its application to power control and power allocation for cellular and cell-free networks is discussed in Chapters 4 and 5, respectively.

In the expert-driven domain, the extension of the established fractional power control algorithm to cell-free networks is analyzed in Chapter 6. It reaches an almost optimum solution in the uplink, while providing satisfactory results for the downlink.

The outcomes and contributions of this thesis have been published as eight IEEE journal and conference, and a public GitHub repository. In what follows, the publications are listed according to topics.

We initially proposed the unsupervised-learning approach for the power control and power allocation problem in cell-free networks in:

- R. Nikbakht, and A. Lozano, "Unsupervised-learning power control for cell-free wireless systems," in IEEE Int'l Symp. Personal, Indoor and Mobile Radio Commun. (PIMRC'19), Istanbul, Turkey, 2019, pp. 1-5.

- R. Nikbakht, A. Jonsson and A. Lozano, "Unsupervised-Learning Power Allocation for the Cell-Free Downlink," IEEE Int'l Conf. Commun. (ICC'19 Workshops), Dublin, Ireland, 2020, pp. 1-5.

Later, the extended version of the unsupervised-learning approach was released as three invited letters, with the first one covering the core idea and mathematical derivation (refer to Chapter 3):

- R. Nikbakht, A. Jonsson, and A. Lozano, “Unsupervised learning for parametric optimization,” to appear in IEEE Commun. Letters, vol. 25, 2021.

The second one, focusing on cellular networks (refer to Chapter 4):

- R. Nikbakht, A. Jonsson, and A. Lozano, “Unsupervised learning for cellular power control,” to appear in IEEE Commun. Letters, vol. 25, 2021.

And, finally, the last one concentrating on the cell-free networks (refer to Chapter 5):

- R. Nikbakht, A. Jonsson, and A. Lozano, “Unsupervised learning for C-RAN power control and power allocation,” to appear in IEEE Commun. Letters, vol. 25, 2021.

The expert-driven power allocation solutions for cell-free networks are investigated in (refer to Chapter 6):

- R. Nikbakht and A. Lozano, “Uplink fractional power control for cellfree wireless networks,” IEEE Int’l Conf. Commun. (ICC’19), Shanghai, China, 2019, pp. 1-5.
- R. Nikbakht, R. Mosayebi, and A. Lozano, “Uplink fractional power control and downlink power allocation for cell-free networks,” IEEE Wireless Commun. Letters, vol. 9, no. 6, pp. 774-777, June 2020.

Then, the power maps are presented in (refer to Chapter 7):

- R. Nikbakht, A. Jonsson and A. Lozano, ”Dual-Kernel Online Reconstruction of Power Maps,” IEEE Global Telecommun. Conf. (GLOBECOM’18), Abu Dhabi, UAE, 2018, pp. 1-5.

Lastly, the TensorFlow implementation of the unsupervised-learning approach is publicly available at:

- github.com/RasoulNik/UnsupervisedNN.

A short introduction to each approach, alongside the major contributions and results, are presented in Chapter 2. The conclusions and future directions are discussed in Chapter 8.

Chapters 3-7 are the self-contained published articles, appearing in the thesis without any modifications.

Chapter 2

Introduction

> *...All the wonders of our universe can in effect be captured by simple rules, yet ... there can be no way to know all the consequences of these rules, except in effect just to watch and see how they unfold.*
>
> Stephen Wolfram

We first present the core idea of the unsupervised-learning approach and how it serves to tackle parametric optimizations. We then proceed to power control and power allocation in cellular and cell-free networks, a perfect use case of parametric optimizations.

Fractional power control follows, as a prime example of expert-driven approaches that perform satisfactorily.

Finally, we discuss power maps and present a simple yet effective learning tool to reconstruct power maps in an online fashion.

2.1 Unsupervised-learning approach: The core idea

An unsupervised-learning approach is a fast and computationally efficient method for solving parametric optimizations. It can be applied to a wide range of optimization problems with convex and non-convex objectives. In wireless applications, it provides satisfactory results and reaches near-optimum solutions.

As its name implies, parametric optimization refers to a class of optimizations with a fixed structure but different parameters. The aim is not to solve a specific instance of a problem, but rather to present a solution for a family of problems. In short, the solution of a parametric optimization is a mapping from a parameter value to the optimum solution associated with this specific parameter value.

For parametric optimization with a non-linear objective, several approximations have been developed, including outer approximations [13, 14], gradient descent [15], and piecewise quadratic fitting [16]. These algorithms often try to solve a sequence of optimization problems in order to find the desired mapping between parameters and optimum solutions. For higher-dimensional problems, the number of such optimization problems increases rapidly, eventually becoming computationally intractable. Therefore, they cannot be applied to high-dimensional problems.

Supervised learning has already been applied to parametric optimizations, where the parameter space is sampled and the optimization problems associated with each instance of the parameters are solved, globally or locally, beforehand. Then, a feedforward NN is trained in order to find the desired mapping between parameters and solutions [17, 18]. This approach has two main drawbacks:

- Computational: the training database demands extensive computations, even for a problem with reasonably small dimensions.
- Applicability: if the objective is non-convex, a dedicated solver should be developed for each family of problems.

The unsupervised-learning approach is developed to tackle both issues. In this approach, a feedforward NN is trained by repeatedly sampling the parameter space, but, rather than solving the associated optimization problem completely, a single step is taken along the gradient of the objective function.

For a given parametric optimization, the steps of the proposed algorithm can be summarized as:

- Consider the parameter values as inputs to the NN.

- Define a custom loss function for the NN based on the objective and the constraints of the optimization problem (negative of the Lagrange function). Such a loss function should be defined as

$$\text{Loss} = -\text{Objective} + \boldsymbol{\beta}\,\text{Constraints}, \tag{2.1}$$

  If we think of the loss as the Lagrange function, $\boldsymbol{\beta}$ has an effect similar to that of a Lagrange multiplier. In general, finding the optimum $\boldsymbol{\beta}$ itself is an optimization problem (the dual problem). In our work, a cross validation approach is used for finding the desired $\boldsymbol{\beta}$ values.

- Start with the initial output of the NN (which is a random vector and is a function of the initial weights of the NN) and update the weights to minimize the loss function.

The resulting NN implicitly approximates the mapping between the parameters and optimum solutions.

This approach is called unsupervised because it does not need labeled data for training. For quadratic programming (QP), for instance the schematic of the unsupervised-learning approach is shown in Fig. 2.1, where θ is the optimization parameter and L_{QP} is the loss function. For a given parameter θ, the NN outputs $\mathbf{x}^-$ and $\mathbf{x}^+$, each of them being a part of the final solution $\mathbf{x}$. The $\exp$ operation improves the dynamic range of the NN output and its convergence speed. Finally, the Adam optimizer updates the NN using the loss function L_{QP}.

Advantages

An unsupervised NN offers the following advantages:

1. It can be applied to convex and non-convex optimization problems.
2. The training time is two-to-three orders of magnitude shorter than in a supervised approach.
3. The test time is two orders of magnitude faster than with a convex solver, for those scenarios where this is available [19].

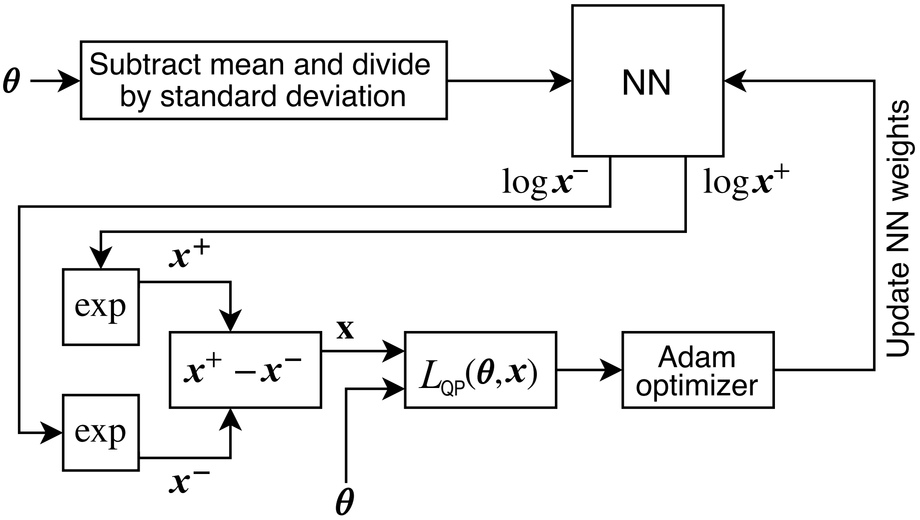


Figure 2.1: Schematic of the unsupervised-learning approach for a QP problem.

Implementation tips

If the provided problem has structure (most practical problems do), we can apply the unsupervised-learning approach by specifying three major parameters:

- Loss function: the function has to be differentiable, and any discontinuity has to be relaxed.

- Constraints: for each optimization variable $\mathbf{x}$, the effect of any constraint $g(\mathbf{x}) < 0$ can be modelled as $\text{Relue}(g(\mathbf{x}))$, with

$$\text{Relue}(x) = \begin{cases} x & x > 0 \\ 0 & \text{otherwise} \end{cases} . \tag{2.2}$$

- The preprocessing of the data is a crucial task. For example, one can use a logarithmic scale in the input and output of the NN, if it makes sense for a domain expert.

2.2 Unsupervised-learning approach: Application to cellular and cell-free systems

In this section, the unsupervised-learning approach is applied to the power control and power allocation problems in cellular and cell-free network—an infrastructure of APs with one or several central processing units, in which each AP potentially communicates with all users. In each scenario, we consider both the uplink and the downlink. The detailed derivations of the signal-to-interference-plus-noise ratio (SINR) are available in Chapters 4-5. Throughout this section, sinr and SINR refer to small-scale (instantaneous) and large-scale SINR (the local average of sinr), respectively. We consider a wireless network with N APs and K users. In the cellular system, $N = K$ whereas, in the cell-free system, $N > K$.

Optimization problems

In this section, we study the uplink power control for cellular settings. For the link between user k and AP n, the local-average signal-to-noise-ratio (SNR) is

$$\mathsf{SNR}_{n,k} = \frac{G_{n,k} P}{\sigma^2} \tag{2.3}$$

with $G_{n,k}$ the large-scale gain, P the maximum transmit power, and σ^2 the noise power.

We use two fairness criteria:

- Max-min: it focuses on the lower tail behavior, maximizing the SINR of the worst-case user.
- Max-product: it enables a trade-off between the average SINR and the lower tail behavior, maximizing the product of the SINRs for different users.

Power control based on max-min and max-product criteria is expressed as

$$\begin{aligned}\text{Max-min: } & \max_{t}\; t && (2.4)\\ & \text{s.t} \quad \mathsf{SINR}_k(\mathbf{SNR}, \mathbf{p}) \geq t \quad k = 1, .., K \\ & 0 \leq \mathbf{p}_k \leq 1 \quad k = 1, .., K \\ \text{Max-product: } & \max_{\mathbf{p}} \sum_k \log\left(\mathsf{SINR}_k(\mathbf{SNR}, \mathbf{p})\right) && (2.5)\\ & \text{s.t} \quad 0 \leq \mathbf{p}_k \leq 1 \quad k = 1, .., K\end{aligned}$$

where $\mathbf{p}$ is the power control vector and $\mathbf{SNR}$, defined as $\{\mathsf{SNR}_{n,k}\}$, is the matrix of SNR for all the links in the network. This is an example of parametric optimization, with $\mathbf{SNR}$ being the parameter. It is worth repeating that all approaches proposed in this thesis are developed based on the large-scale gains, but the actual performance evaluations are carried out using sinr.

As is shown in Chapter 4, the worst-case users drag down the max-min solution, so we define a soft counterpart to it

$$\begin{aligned}\text{Soft max-min: } & \min_{\mathbf{p}} \log \sum_k \exp\left(\frac{\alpha_k}{\mathsf{SINR}_k(\mathbf{SNR}, \mathbf{p}}\right) && (2.6)\\ & \text{s.t} \quad 0 \leq \mathbf{p}_k \leq 1 \quad k = 1, .., K\end{aligned}$$

where the parameters $\{\alpha_k\}$ determine how stringent the max-min criterion is, converging to the max-min solution for growing $\{\alpha_k\}$.

Because of the $\exp$ function, users with lower SINR values have a much greater impact on the objective, leading to an approximate max-min behavior. However, for small values of $\{\alpha_k\}$, users with high SINRs also contribute to the optimization problem, improving the upper tail behavior of the SINR CDF while having a negligible impact on the lower tail.

Loss functions

In cellular settings, by following the steps in the unsupervised-learning approach (refer Section 2.1) and using (2.6), we define the soft max-min loss function as

$$L_{\mathsf{MM}} = \frac{1}{K}\left[\underbrace{\sum_{k=0}^{K-1}\exp\left(\frac{\alpha_k}{(\mathsf{SINR}_k + 0.01)^{0.4}}\right)}_{\text{Objective}} + 0.1\underbrace{\sum_{k=0}^{K-1}[p_k - 1]^+}_{\text{Constraints}}\right] \tag{2.7}$$

where the parameter α_k, similar to (2.6), determines how stringent the max-min criterion is. The constraints are modeled as an extra term in the loss function, with multiplier equal to 0.1. The exponent in the denominator of the objective part equals 0.4, which prevents numerical errors during the NN training, compressing the dynamic range of the loss function.

Again for the cellular setting, using (2.5), the max-product cost function can be derived as

$$L_{\mathsf{MP}} = \frac{1}{K}\left[\underbrace{\sum_{k=0}^{K-1}\beta_k \log\left(0.01 + \frac{1}{\mathsf{SINR}_k + 0.01}\right)}_{\text{Objective}} \quad \underbrace{+\,0.1\sum_{k=0}^{K-1}[p_k - 1]^+}_{\text{Constraints}}\right] \tag{2.8}$$

where β is the regulating factor that prioritizes different users. The constraints are added to the loss function, again, by setting the multiplier equal to 0.1. The offset shifting SINR_k by 0.01 avoids the loss being pulled down by users below -20 dB while a second offset added to $\frac{1}{\mathsf{SINR}_k+0.01}$ lessens the pull of users above 20 dB.

With some small modifications, both loss functions can be applied to cell-free settings (refer to Chapter 5).

Insights to the learning process

For cellular settings, the learning pipeline is shown in Fig. 2.2. As discussed in the implementation tips, proper pre and post data processing are essential. In this case, to improve the dynamic range, we feed the data and read the output, both in logarithmic scale.

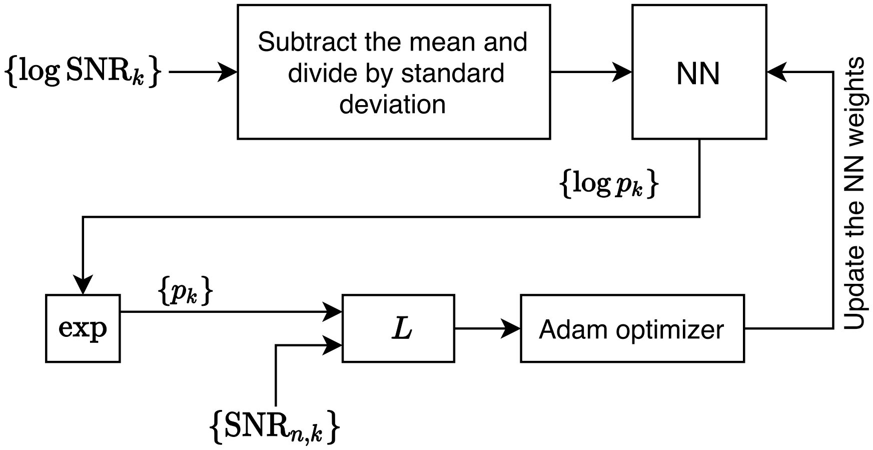


Figure 2.2: Learning pipeline for the cellular setting

Following this pipeline, the NN can learn the structure of the given optimization problem and produce the desired output. Shown in Fig. 2.3 is an example of learning curve for the uplink of a cellular setting. In this example, the soft max-min and minimum SINR are plotted side by side. As we can seen, feeding the large-scale data and updating the NN weights, the loss value drops rapidly and the minimum SINR increases. By feeding more data, the NN converges to a stable solution and the fluctuations of the loss function decrease. It is possible to derive a similar chart for cell-free settings.

Performance evaluation in the cellular setting

Plotted in Fig. 2.4 is the comparison, for the soft max-min setting, between the unsupervised-learning approach and the convex solver. We see how the NN closely approximates the convex solver. Also, in the inset, the CDF of the optimum objective values is compared with the L_{MM}. Again, it is seen that the NN almost optimally minimizes the cost function.

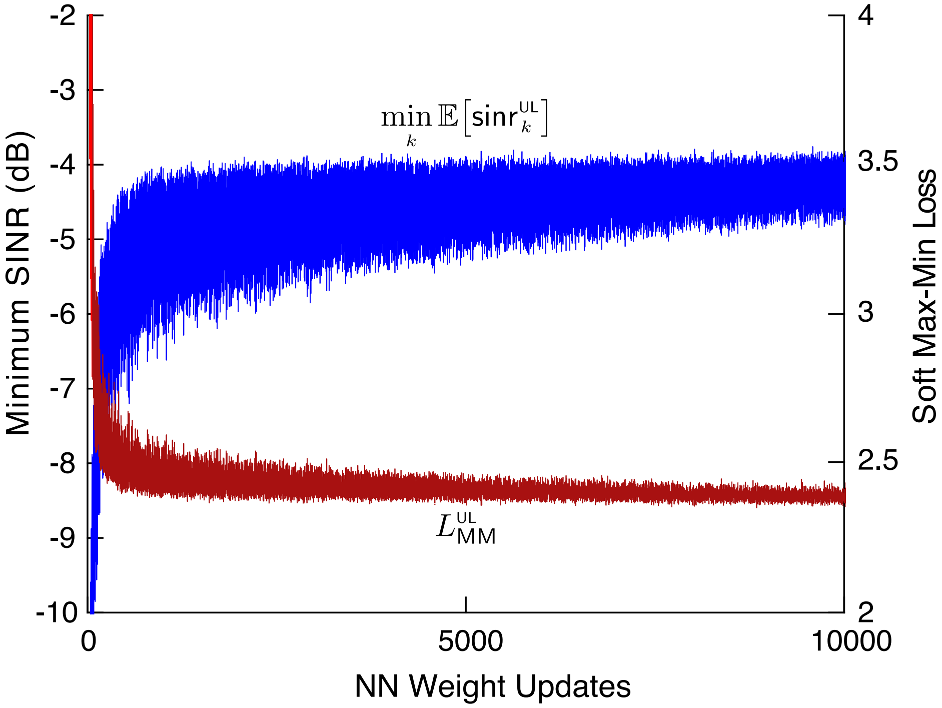


Figure 2.3: Uplink soft max-min learning curve with $\alpha_k = 1$: soft max-min loss and minimum SINR as a function of the NN weight updates.

Performance evaluation in the cell-free setting

The NN performance for the cell-free uplink, alongside its convex solver counterpart, is presented in Fig. 2.5. We can make the following observations:

- In cell-free settings, the soft max-min outperforms the max-min solution almost uniformly.
- The NN can reliably replace the convex solver.
- By increasing the $\{\alpha_k\}$, the soft max-min gets progressively closer to max-min.

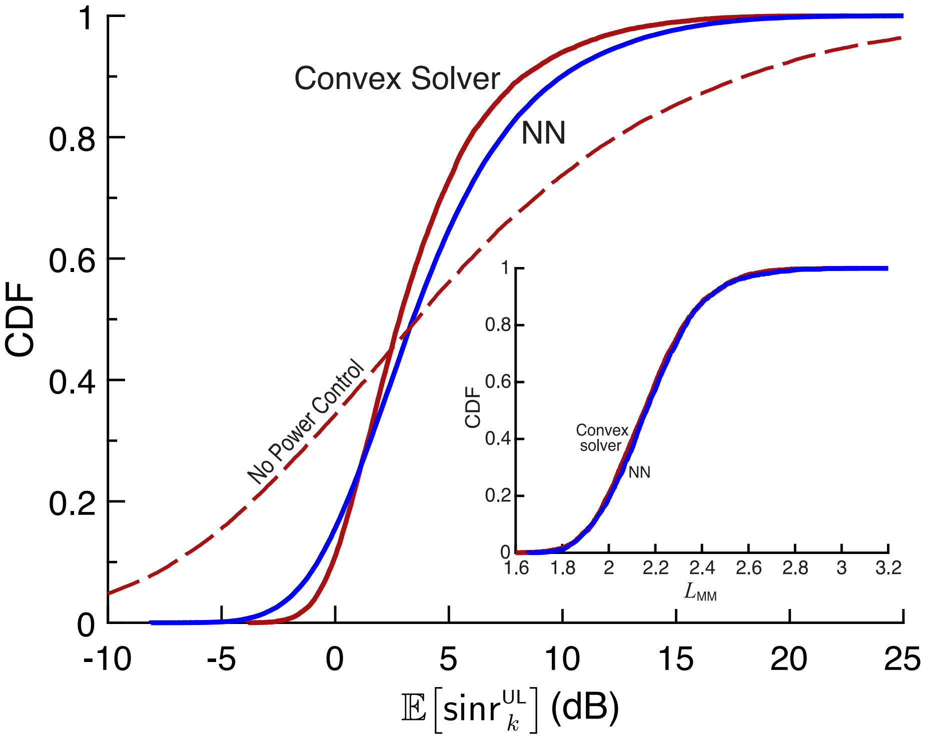


Figure 2.4: CDF of $\mathbb{E}\left[\mathsf{sinr}_k^{\mathsf{UL}}\right]$ for the soft max-min loss function ($\alpha_k = 1$): NN vs convex solver, alongside the baseline without power control ($p_k = 1$). In the inset, CDF of L_{MM} for the NN and the convex solver.

The downlink case is even more interesting because the power allocation cannot reliably[1] be cast in a convex form. The NN performs satisfactorily in this scenario as well, as discussed in Chapter 5.

Computational advantage

We use a convex solver as a benchmark for the test time, which is the time spent outputting a solution for a single instance of the optimization problem. In the cell-free setting, for $N = 30$ and $K = 12$, the unsuper-

[1] It is a quasi-convex problem and can be tackled by solving a sequence of convex optimization problems.

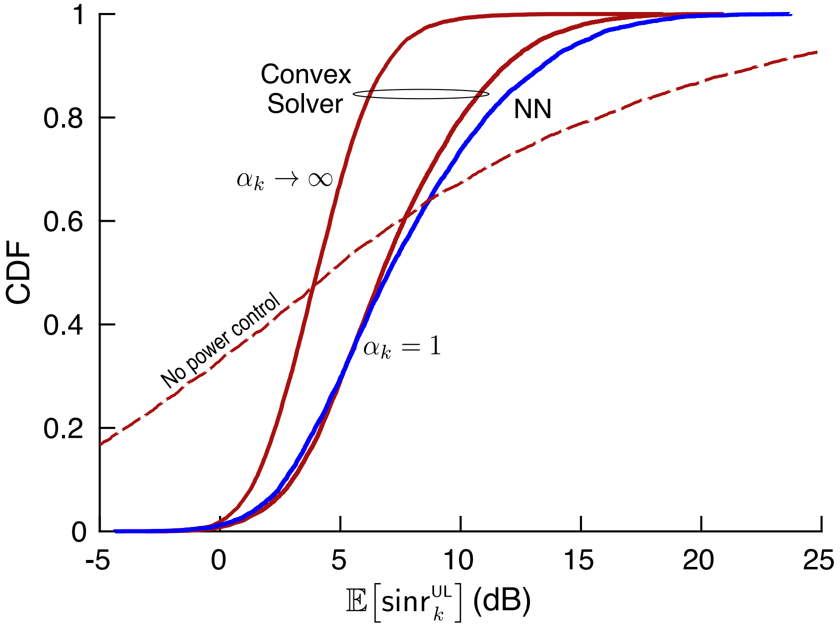


Figure 2.5: CDF of $\mathbb{E}\left[\mathsf{sinr}_k^{\mathsf{UL}}\right]$ for soft max-min: NN vs convex solver ($\alpha_k = 1$). Also shown is the baseline performance without power control, and the hard max-min performance ($\alpha_k \to \infty$).

vised-learning approach is over two (uplink) and three (downlink) orders of magnitude faster than the convex solver in terms of test time.

Also, in terms of training, the unsupervised-learning approach is compared with the supervised one, which treats parametric optimization as a supervised learning problem by computing a database of desirable solutions for a large set of parameters. Such a database is not required in the unsupervised-learning approach, resulting in orders of magnitude advantage in terms of training.

2.3 Extension of fractional power control to cell-free networks

Different fairness criteria, e.g., max-min or max-product, are enabled by power control. To unlock the benefits of power control, a large body of literature has focused on deriving optimum or sub-optimum power allocation polices, but often these approaches are computationally expensive and therefore not applicable in practical scenarios.

Fractional power control is one of the most successful policies that has been applied in wireless communications thus far. It features a single parameter and only depends on the large-scale gains. In this work, we extend this approach to cell-free networks and analyse its performance. Our derived solutions have three main features:

- They only depend on large-scale gains, ensuring the scalability.
- They allow for a trade-off between equalizing the performance among users (similar to max-min) and the average system performance (similar to max-product).
- They have an analytical form and can easily be computed in real-time.

Fractional power control

Fractional power control originally was derived to minimize, in log scale, the variance of the large-scale signal-to-interference ratio (SIR)—an approximation of the instantaneous SIR based on the large-scale gains. Initially, in [6], large-scale SIRs were computed for two users and, for user k, the power allocation coefficient was computed as $p_k \propto 1/\sqrt{G_k}$, where G_k is the large-scale gain between user k and its serving base station. This was subsequently generalized in [7, 8] to

$$p_k \propto \frac{1}{G_k^{\vartheta}} \tag{2.9}$$

where $\vartheta \in [0, 1]$ regulates the extent to which the range of dB-scale received powers is compressed. The proportionality factor is set on the basis of the power constraints.

Uplink

Assume a cell-free network with k users and N APs where matched filter detection and conjugate beamforming are used in uplink and downlink, respectively. For the uplink case, our work [9–11] formulated the fractional power control coefficient as

$$p_k \propto \frac{1}{\left(\sum_{n=0}^{N-1} G_{n,k}\right)^{\vartheta}}, \tag{2.10}$$

where $G_{n,k}$ is the large-scale gain between user k and AP n and $\vartheta \in [0, 1]$ is the power control parameter. The derived power control depends on all the large-scale channel gains that involve a given user, reflecting the effective connection between such user and the network.

For an interference-limited scenario, the detailed derivation of the large-scale SIR is discussed in [10, 11]. Similar to the previous section, the proposed power allocation scheme is derived based on the large-scale SIR, while the performance evaluation is carried out using $\mathbb{E}[\mathsf{sir}_k]$ (the instantaneous signal-to-interference ratio averaged over the small-scale fading).

The effectiveness of (2.10) is illustrated in Fig. 2.6. We make the following observations:

- The parameter ϑ effectively compresses and expands the distribution of $\mathbb{E}[\mathsf{sir}_k]$ (in log scale), enabling a trade-off between lower tail behaviour and average value.

- The max-min is too strict and dragged down by worst-case users.

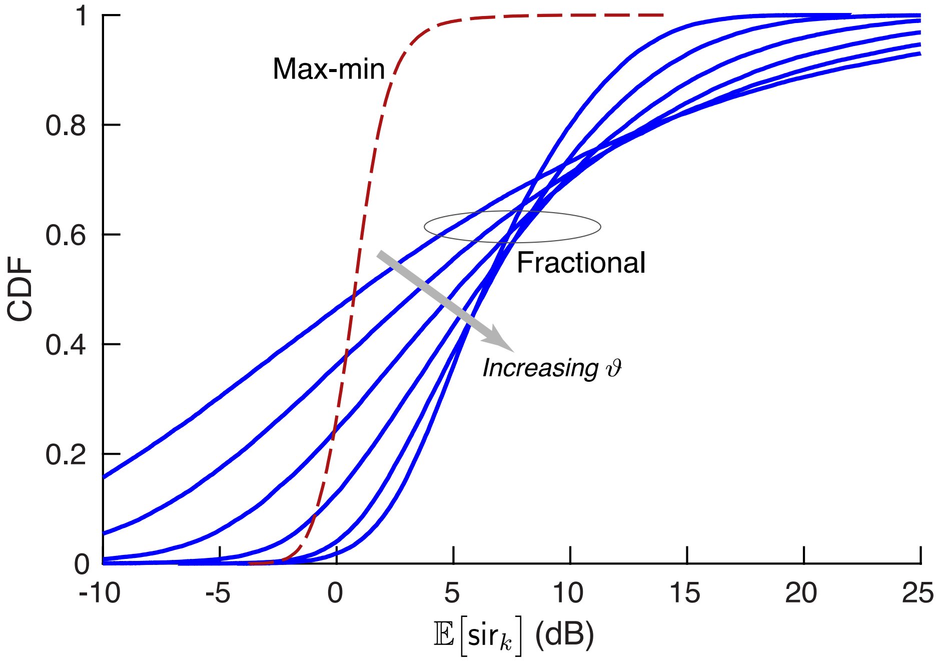


Figure 2.6: CDF of $\mathbb{E}[\mathsf{sir}_k]$ for the uplink with $N/K = 2.5$, parameterized by $\vartheta = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. Also shown is the max-min solution.

Downlink

Inspired by the uplink performance, we proposed the following policy for downlink power allocation [10]:

$$p_{n,k} \propto \frac{G_{n,k}}{\left(\sum_{m=0}^{N-1} G_{m,k}\right)^{\vartheta} \left(\sum_{\ell=0}^{K-1} \frac{G_{n,\ell}}{\left(\sum_{m=0}^{N-1} G_{m,\ell}\right)^{\vartheta}}\right)^{\gamma}}, \tag{2.11}$$

where, as in the uplink, $\vartheta \in [0, 1]$ while γ is an additional parameter best set in the range $[0.4, 1.6]$.

This policy has two major components:

- $\frac{G_{n,k}}{\left(\sum_{m=0}^{N-1} G_{m,k}\right)^{\vartheta}}$, which is the dual form of the uplink power allocation.

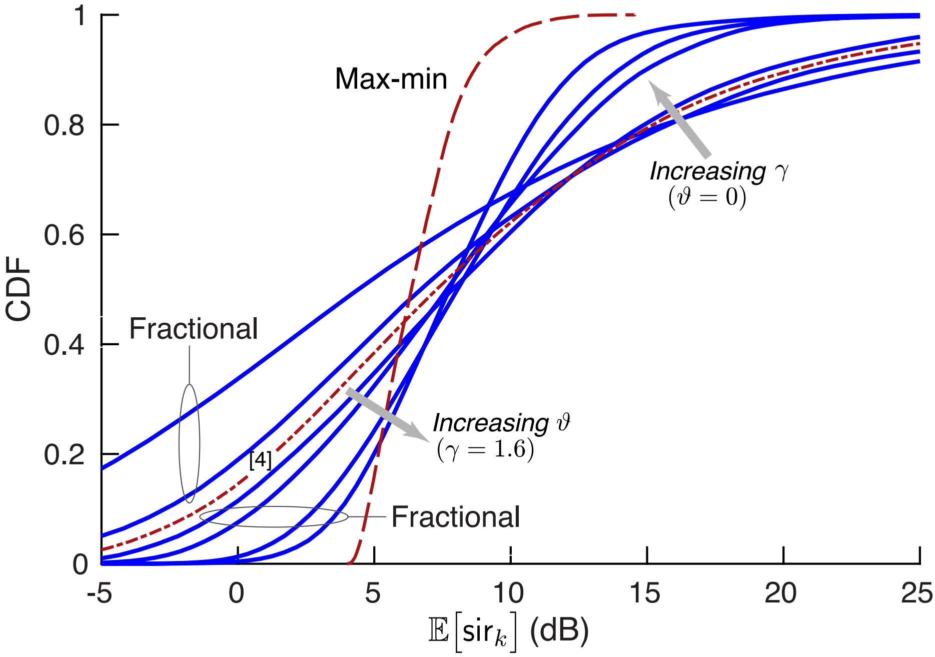


Figure 2.7: CDF of $\mathbb{E}[\mathsf{sir}_k]$ for the downlink with $\eta = 3.8$ and $N/K = 2.5$, parameterized by $\vartheta = \{0, 0.4, 0.8\}$ and $\gamma \in \{0.4, 0.8, 1.2, 1.6\}$. Also shown are the max-min solution and the benchmark from [12, Sec. III-D].

- A normalization factor

$$\frac{1}{\left(\sum_{\ell=0}^{K-1} \frac{G_{n,\ell}}{\left(\sum_{m=0}^{N-1} G_{m,\ell}\right)^{\vartheta}}\right)}, \tag{2.12}$$

  that makes sure all APs have the same maximum transmit power. Later, by introducing the parameter γ, we relax this condition.

Shown in Fig. 2.7 is the performance evaluation of the proposed method (more details are provided in [10]). Because of the two parameters, the performance is more versatile. Similar to the uplink, a trade-off is observed between fairness and lower tail behaviour. However, the lower tail falls short of the max-min solution.

2.4 Power maps

A power map determines the level of local-average received power at every point of a 2-D environment. Unlike parametric modelling, such as the COST-231 Hata [1] or the Stanford University interim model [2], which are suitable for a class of environments, power maps focus on site-specific behaviors.

Historically, ray tracing, recreating an environment in a computer simulator and applying wireless wave propagation mechanisms, has been used for modelling site-specific behaviors. Ray tracing demands a very detailed recreation for both the geometry and the wireless properties of the constituent materials. Therefore, it is mostly restricted to indoor sites.

Measurement-driven reconstruction of the wireless environment or power map is a viable option for outdoor environments. In typical cellular networks, a large number of users are constantly reporting their pathloss or local-average received power. By exploiting such a database, we can recreate the propagation environment from the perspective of any AP. Power maps are of great benefit for (proactive) resource allocation, network planning, and interference mitigation.

Online and batch methods are two approaches for reconstructing power maps. In the batch method, all the measurements need to be available beforehand, which has the drawback of not being able to adapt to environment changes. Online power map reconstruction (see Chapter 7) processes measurements one at a time. After having received a certain number of measurements, it can build a 2-D map of the local-average received power from the viewpoint of the given AP. Because of the dynamic nature of the online method, the effect of environment changes, e.g, construction sites or traffic, could also be incorporated to the online power maps.

Dual-kernel method

In Fig. 2.8, pathloss values (the decay in power level in log scale from the given AP to the different users or a user in different locations) are shown for a specific site. It can be seen that pathloss has a varying part oscillating around a DC component, which itself varies as new measurements are being collected. Also, the pathloss is bounded from below and above. In this specific site, the pathloss measurements fall between 65 and 115 dB.

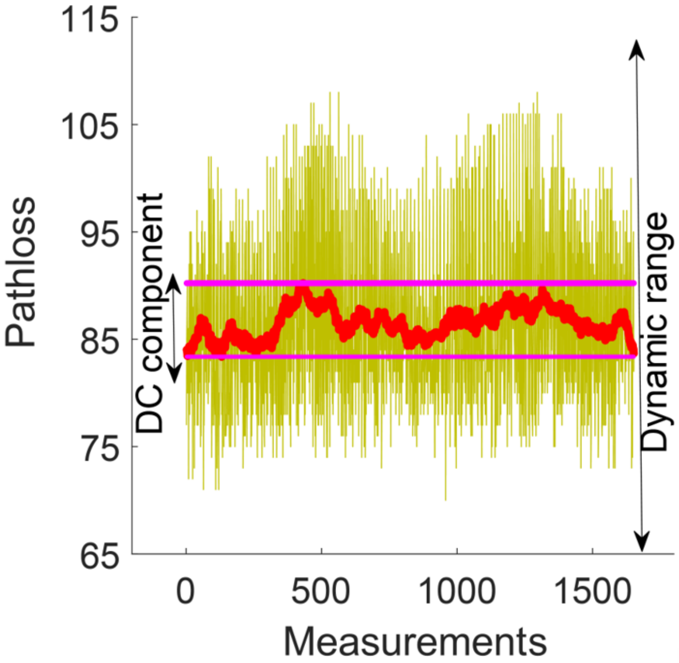


Figure 2.8: Pathloss variation as function of new measurements

Dual-kernel is the main contribution of our proposed method for online power map reconstruction. A kernel is a function that measures the similarities between two points in 2-D space. For instance, a well-defined kernel could suggest how similar is the level of received power at two points of space separated by distance D. In our dual-kernel approach, we have the DC and varying kernel, defined as:

- Varying kernel: it models the varying part of pathloss (variations around the DC value) a Gaussian kernel $e^{-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|}{D}}$ is adopted. In our case, $\mathbf{x}_1$ and $\mathbf{x}_2$ are user locations and the kernel parameter D is the correlation distance. If two users are closer than D on the 2-D plane, they will have similar pathloss values.

- DC kernel: it captures the DC component of the signal. We have used another Gaussian kernel with much larger kernel parameter D. It ensures that the location-independent DC component of the pathloss (a signal similar to Fig. 2.8) can be captured by the DC kernel.

In summary, the dual kernel approach provides just enough complexity to capture the underlying phenomena. Its simple structure avoids the overfitting problem—choosing a very complex model to explain a simple behaviour. As a result, the dual kernel approach outperforms both the multi-kernel and the single-kernel approaches, offering the right amount of the complexity.

Kernel-based regression

The power map reconstruction is an online regression problem—the goal is to predict the value of the pathlosss for a given location in 2-D space based on the available samples of locations and pathloss values. In any regression task, a group of functions or hypotheses are assumed. In such a model, the objective of learning is to choose a subset of these functions that can both explain the current data (learning set) and unseen data (test set) with the least minimum mean square error (MMSE).

A kernel function defines these hypotheses. For instance, for a given point $\mathbf{x}_n$ on the 2-D plane, the Gaussian kernel defines a function as $f(x) = e^{-\frac{\|\mathbf{x}_n - \mathbf{x}\|}{D}}$. Therefore, for a given data set of locations and pathloss values, the regression hypothesis could be defined as

$$\mathbb{F} = \left\{ \sum_n \alpha_n \, e^{-\frac{\|\mathbf{x}_n - \mathbf{x}\|}{D}} \right\}, \quad a_n \in \mathbb{R} \tag{2.13}$$

where the index n refers to the data points. Any function in the regression hypothesis $\mathbb{F}$ is a linear combination of bases, each of them associated with a measurement. The set of measurements that are used to build $\mathbb{F}$ is called the dictionary, with each measurement being a dictionary element.

After defining the kernel, and consequently the regression hypothesis, the power map reconstruction boils down to finding the optimum

$$f^\star(\mathbf{x}) = \sum_n \alpha_n^\star \, e^{-\frac{\|\mathbf{x}_n - \mathbf{x}\|}{D}}, \tag{2.14}$$

or specifying the effect of each dictionary element $\alpha_n^\star$. In Chapter 7, we deep dive into this problem using a technique called adaptive projected subgradient method (APSM) [3].

Sparsification

Sparsification is a technique for controlling the size of the dictionary, which grows over time. Since the measurements arrive in an online fashion, incorporating all the measurements could result in a large dictionary. Also, some outdated measurements may not be a good basis for power map reconstruction because of the changes in the environment. To resolve this issue, the following steps are implemented:

- The function complexity or norm, defined as $\|\boldsymbol{\alpha}\|_2$ with $\boldsymbol{\alpha}$ being equal to $\{\alpha_n\}$, is set to a number like 100 or 200. The function norm roughly determines the number of effective elements or measurements in the dictionary.

- The redundant or less informative measurements are removed from the dictionary if their effect on the function norm (the function we use for power map reconstruction) is negligible.

By applying these steps in an online fashion (after a new measurement is collected), an online power map could be reconstructed that not only has a limited number of dictionary elements (lower computational burden) but also tracks the changes in the environments.

Advantage

The proposed approach provides an online regression tool for power map reconstruction. Having sufficient but not limited complexity, the dual kernel structure leads to a superior performance in the following areas:

- The convergence speed improves dramatically. It can build an accurate picture of the power map by processing 500 samples.
- The prediction variance improves significantly, since the DC kernel reliably captures a large portion of the pathloss.
- A simple sparsification technique ensures both a small dictionary size (less computation) and an online behavior that can track the environment changes.

2.5 Summary

In this chapter, without diving into the mathematical details, we have discussed the main contributions of the thesis. Also, some important results have been included.

References

[1] "Urban transmission loss models for mobile radio in the 900 and 1800 MHz bands," Tech. Rep., European Cooperation in the Field of Scientific and Technical Research EUROCOST 231, September 1991.

[2] V. Erceg, L. J. Greenstein, S. Y. Tjandra, S. R. Parkoff, A. Gupta, B. Kulic, A. A. Julius, and R. Bianchi, "An empirically based path loss model for wireless channels in suburban environments," *IEEE J. Select. Areas Commun."*, vol. 17, no. 7, pp. 1205–1211, Jul 1999.

[3] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *Signal Processing Mag.*, vol. 28, no. 1, pp. 97–123, Jan 2011.

[4] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, pp. 1834–1850, Mar. 2017.

[5] G. Interdonato, H. Q. Ngo, E. G. Larsson, and P. Frenger, "How much do downlink pilots improve cell-free massive MIMO?," *IEEE Global Commun. Conf. (GLOBECOM'16)*, pp. 1–7, 2016.

[6] J. F. Whitehead, "Signal-level-based dynamic power control for co-channel interference management," in *IEEE Veh. Techn. Conf. (VTC'93)*, May 1993, pp. 499–502.

[7] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341–1347, Sep. 1995.

[8] A. Simonsson and A. Furuskar, "Uplink power control in LTE—overview and performance," *IEEE Veh. Techn. Conf. (VTC'08 Fall)*, pp. 1–5, Sep. 2008.

[9] R. Nikbakht, "Unsupervised NN," https://github.com/RasoulNik/UnsupervisedNN, 2020.

[10] R. Nikbakht, R. Mosayebi, and A. Lozano, "Uplink fractional power control and downlink power allocation for cell-free networks," *IEEE Wireless Communications Letters*, vol. 9, no. 6, pp. 774–777, 2020.

[11] A. Lozano and R. Nikbakht, "Uplink fractional power control for cell-free wireless networks," in *IEEE Int'l Conf. on Communications (ICC'19)*, May 2019.

[12] E. Nayebi, A. Ashikhmin, T. L. Marzetta, H. Yang, and B. D. Rao, "Precoding and power optimization in cell-free massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 16, pp. 4445–4459, Jul. 2017.

[13] J. Acevedo and E. N. Pistikopoulos, "A parametric MINLP algorithm for process synthesis problems under uncertainty," *Ind. Eng. Chem. Res.*, vol. 35, pp. 147–158, 1996.

[14] V. Dua and E. N. Pistikopoulos, "An outer-approximation algorithm for the solution of multiparametric MINLP problems," *Computers them. Engng*, vol. 22, pp. S955–S958, 1998.

[15] A. Rantzer, "Dynamic dual decomposition for distributed control," in *Proceedings of the American Control Conference*, 2009, p. 884–888.

[16] P. Patrinos and H. Sarimveis, "Convex parametric piecewise quadratic optimization: Theory and algorithms," *Automatica*, vol. 47, pp. 1770–1777, 2011.

[17] L. Sanguinetti, A. Zappone, and M. Debbah, "Deep learning power allocation in massive MIMO," in *Asilomar Conf. Signals, Systems, and Computers*, 2018, pp. 1257–1261.

[18] T. Van Chien, E. Björnson, and E. G. Larsson, "Sum spectral efficiency maximization in massive MIMO systems: Benefits from deep learning," *CoRR*, vol. abs/1903.08163, 2019.

[19] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," `http://cvxr.com/cvx`, Mar. 2014.

[20] R. Nikbakht, A. Jonsson, and A. Lozano, "Unsupervised-learning power allocation for the cell-free downlink," in *IEEE Int'l Conf. on Communications (ICC'20)*, Jun. 2020.

[21] R. Nikbakht, A. Jonsson, and A. Lozano, "Unsupervised learning for cellular power control," *IEEE Commun. Letters*, vol. 24, 2020.

[22] R. Nikbakht, A. Jonsson, and A. Lozano, "Unsupervised learning for C-RAN power control and power allocation," *IEEE Commun. Letters*, vol. 24, 2020.

Chapter 3

Unsupervised Learning for Parametric Optimization

Rasoul Nikbakht, Student Member, IEEE, Anders Jonsson, Angel Lozano, Fellow, IEEE

***Abstract*— This letter proposes the unsupervised training of a feedforward neural network to solve parametric optimization problems involving large numbers of parameters. Such unsupervised training, which consists in repeatedly sampling parameter values and performing stochastic gradient descent, foregoes the taxing precomputation of labeled training data that supervised learning necessitates. As an example of application, we put this technique to use on a rather general constrained quadratic program. Follow-up letters subsequently apply it to more specialized wireless communication problems, some of them nonconvex in nature. In all cases, the performance of the proposed procedure is very satisfactory and, in terms of computational cost, its scalability with the problem dimensionality is superior to that of convex solvers.**

***Index Terms*— Machine learning, neural networks, unsupervised learning, parametric optimization, convex optimization, quadratic program**

R. Nikbakht, A. Jonsson, and A. Lozano are with Univ. Pompeu Fabra, 08018 Barcelona (e-mail: angel.lozano@upf.edu). This work was supported by the European Research Council under the H2020 Framework Programme/ERC grant agreement 694974, by the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502) as well as by MINECO's Projects RTI2018-102112 and RTI2018-101040, and by the

3.1 Introduction

In parametric optimization [1–4], the aim is not to solve a specific optimization problem, but rather to represent the solution to an entire family of problems. This family of problems is governed by parameters, such that each combination of parameter values corresponds to a particular optimization problem. The desired output of parametric optimization is a mapping from parameter values to the optimum solution of the specific associated problem.

When the objective function is nonlinear in the input, the setting becomes that of parametric nonlinear programming [5]. For this setting, a number of approximation algorithms exist, including outer approximations [5, 6], gradient descent [7], or piecewise quadratic fitting [8]. Such state-of-the-art algorithms have in common that they completely (or approximately) solve a sequence of optimization problems in order to estimate the desired mapping between parameters and solutions. When the parameter dimensionality increases, these algorithms need to solve a very large number of such optimizations in order to estimate an accurate solution, eventually becoming computationally intractable. As a result, existing algorithms scale poorly to parametric optimizations with many parameters.

This letter propounds to leverage the expressive power of neural networks (NNs) to solve large-dimensional parametric optimizations. Because of their nonlinear nature, NNs are ideal candidates for nonlinear optimizations. Indeed, NNs have been applied already to solve parametric optimizations in a supervised fashion, by sampling parameter values and solving the associated optimization problems [9, 10]. Again, for large parameter dimensionalities this becomes unfeasible because of the sheer size of the parameter space.

As an alternative, what is proposed in this letter is to train a feedforward NN by repeatedly sampling parameter values, but, rather than completely solving the associated optimization problem for each, taking a single step along the gradient of the objective function. Despite not

ICREA Academia program.

providing the NN with the solution for each parameter value, descending along the gradient allows the NN to generalize information for different such values.

To test the proposed idea, we apply it to quadratic programming (QP), a simple but very common—and convex—class of optimizations. In two follow-up letters, we turn to more involved problems motivated by wireless communications and not always exhibiting convex objectives. Precisely, power control for cellular networks is considered in [11] whereas power control for centralized radio-access networks is the object of [12].

3.2 Parametric Optimization

Letting $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^Q$ be a Q-dimensional parameter vector and $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^D$ a D-dimensional optimization vector, consider the optimization problem

$$\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{\theta}, \boldsymbol{x}). \tag{3.1}$$

In parametric optimization, the goal is to estimate a *minimizer function* $\boldsymbol{x}^\star : \Theta \to \mathcal{X}$ or a *value function* $g : \Theta \to \mathbb{R}$, both of which map parameter vectors to a solution of the corresponding optimization problem. For a specific $\boldsymbol{\theta}$, the minimizer and value functions are

$$\boldsymbol{x}^\star(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{\theta}, \boldsymbol{x}) \tag{3.2}$$

$$g(\boldsymbol{\theta}) = \min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{\theta}, \boldsymbol{x}). \tag{3.3}$$

Note that, if we have access to $\boldsymbol{x}^\star(\boldsymbol{\theta})$, the value function is trivially $g(\boldsymbol{\theta}) = f\big(\boldsymbol{\theta}, \boldsymbol{x}^\star(\boldsymbol{\theta})\big)$. However, in general it is not as easy to induce a minimizer function from a value function.

Previous work in parametric optimization typically imposes restrictions on the sets Θ and $\mathcal{X}$ (convexity) as well as on the objective function f (differentiability and, again, convexity). Under these restrictions, and in particular if both $\mathcal{X}$ and f are convex, any local minimizer is also a global minimizer and hence $\boldsymbol{x}^\star$ is well defined. One contribution of our approach is that it makes minimal assumptions about the optimization problem: the only strict requirement is that f be differentiable in $\boldsymbol{x}$. We also do take the feasible set $\mathcal{X}$ to be convex, which suffices to ensure that $\boldsymbol{x}^\star$ is well defined, but we do not restrict f to being convex. Of course, for nonconvex f, what gradient descent can do is to find a local minimizer, with no guarantee that it is also the global minimizer $\boldsymbol{x}^\star$.

As advanced in the introduction, several algorithms do exist already that efficiently approximate $\boldsymbol{x}^\star(\boldsymbol{\theta})$ or $g(\boldsymbol{\theta})$. However, these come at the expense of having to solve a sequence of optimization problems whose number increases exponentially with the parameter dimensionality, Q.

3.3 Proposed Approach

3.3.1 Neural Networks

NNs are universal function approximators that combine simple nonlinear units to form complex networks with substantial expressive power. The universal approximation theorem states that feed-forward NNs with a single hidden layer can approximate continuous functions to arbitrary precision [13–15]. Formally, a feed-forward NN represents a vector-valued function $\boldsymbol{h}(\boldsymbol{a}, \boldsymbol{w})$ with D output dimensions, where $\boldsymbol{a}$ is an input vector and $\boldsymbol{w}$ is a vector of weights. NNs are typically trained using gradient descent on a given loss function $L(\boldsymbol{y})$, where $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{a}, \boldsymbol{w})$ is the NN's output. Since

$$\begin{aligned} L(\boldsymbol{y}) &= L(\boldsymbol{h}(\boldsymbol{a}, \boldsymbol{w})) && (3.4) \\ &= (L \circ \boldsymbol{h})(\boldsymbol{a}, \boldsymbol{w}), && (3.5) \end{aligned}$$

the chain rule allows us to write

$$\nabla_{\boldsymbol{w}} L(\boldsymbol{y}) = \boldsymbol{J}(\boldsymbol{a}, \boldsymbol{w})^\top \nabla_{\boldsymbol{y}} L\big(\boldsymbol{h}(\boldsymbol{a}, \boldsymbol{w})\big), \tag{3.6}$$

where $\boldsymbol{J}(\boldsymbol{a},\boldsymbol{w})$ is the Jacobian matrix

$$\boldsymbol{J}(\boldsymbol{a},\boldsymbol{w}) = \begin{bmatrix} \nabla_{\boldsymbol{w}} h_0(\boldsymbol{a},\boldsymbol{w})^\top \\ \vdots \\ \nabla_{\boldsymbol{w}} h_{D-1}(\boldsymbol{a},\boldsymbol{w})^\top \end{bmatrix}. \tag{3.7}$$

The Jacobian is not explicitly computed; rather, backpropagation is used to disseminate the gradient through the NN in order to update its weights [16]. For example, a regression task can be formulated by means of a square loss $L(\boldsymbol{y}) = ||\boldsymbol{y} - \boldsymbol{t}||^2$, where $\boldsymbol{t}$ is the target output associated with input $\boldsymbol{a}$. The goal is to minimize the loss such that the difference between $\boldsymbol{y}$ and $\boldsymbol{t}$ becomes as small as possible for each input $\boldsymbol{a}$.

3.3.2 Unsupervised Learning for Parametric Optimization

Let us consider an NN that represents a minimizer function $\boldsymbol{x}^\star : \Theta \to \mathcal{X}$. Upon an input $\boldsymbol{\theta} \in \Theta$, the network outputs a D-dimensional vector $\boldsymbol{x} = \boldsymbol{h}(\boldsymbol{\theta},\boldsymbol{w})$. In supervised learning [10], NN training requires a learning stage in which parameter vectors $\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_{S-1}$ are sampled and each of the associated convex optimization problems is solved to obtain respective target outputs $\boldsymbol{x}_0^\star, \ldots, \boldsymbol{x}_{S-1}^\star$. The NN is then trained by means of regression as described above, with the loss being a function of the difference between the predicted and the target output.

The NN is trained by means of stochastic gradient descent applied directly on the objective function, f. To do so, we define the loss function

$$L(\boldsymbol{x}) = f(\boldsymbol{\theta},\boldsymbol{x}), \tag{3.8}$$

whose gradient equals

$$\nabla_{\boldsymbol{w}} L(\boldsymbol{x}) = \boldsymbol{J}(\boldsymbol{a},\boldsymbol{w})^\top \, \nabla_{\boldsymbol{x}} f\big(\boldsymbol{\theta}, \boldsymbol{h}(\boldsymbol{\theta},\boldsymbol{w})\big), \tag{3.9}$$

which requires f to be differentiable in $\boldsymbol{x}$. As anticipated, this is the one premise that cannot be lifted.

If the constraints are simple enough, they can be hardwired into the structure of the NN itself, say by selecting nonlinear units for the output layer that can only produce values within a certain range [17]. With a view to broader generality, though, we prefer to transform the constrained optimization problem

$$\min_{\boldsymbol{x} \in \mathcal{X}} L(\boldsymbol{x}) \tag{3.10}$$

into the unconstrained optimization problem

$$\min_{\boldsymbol{x}} L(\boldsymbol{x}) + \boldsymbol{\beta}^\top \boldsymbol{C}(\boldsymbol{x}), \tag{3.11}$$

where $\boldsymbol{C}(\boldsymbol{x})$ is the vector of constraints that define the feasible set $\mathcal{X}$ and $\boldsymbol{\beta}$ is a vector of Lagrangian (or KKT) multipliers. Hence, the gradient also involves a problem-dependent term that corresponds to $\boldsymbol{C}(\boldsymbol{x})$. Since the loss function is a combination of the objective function and of the cost that is associated with the constraints, training the NN by gradient descent has the effect of minimizing $f(\boldsymbol{\theta}, \boldsymbol{x})$ over the feasible set $\mathcal{X}$, which is precisely the aim of parametric optimization. To circumvent having to identify the optimum $\boldsymbol{\beta}$ analytically, which would require solving the dual optimization problem, we adopt the common strategy of cross-validation: apply the same multiplier β to every constraint, and select its value after trialing $\beta \in \{0.01, 0.1, 1, 10, 100\}$.

The crux of the proposed approach is that, rather than completely solving the optimization problem associated with each specific $\boldsymbol{\theta}$, we take a single step along the gradient before sampling a new parameter vector. We do not know the correct output for a given input, but we can compute the loss given the current prediction of $\boldsymbol{x}$ and update such prediction in order to minimize the corresponding loss. Even though the NN is not provided with the solution for each parameter vector, descending along the gradient allows the NN to quickly generalize across parameter vectors.

Altogether, our proposition can be summarized as follows:

- Treat the optimization parameters as inputs to the NN.
- Create a custom loss function based on the parametric optimization problem.

- Evaluate the NN and update the weights by means of any gradient-based optimizer.

3.4 Application to Constrained QP

Before proceeding, in follow-up letters [11] and [12], to more specific parametric optimizations motivated by wireless communications, let us herein entertain a simpler yet much more general one.

3.4.1 Formulation

A constrained QP is a convex instance of the parametric optimization in (3.1) with

$$f(\boldsymbol{\theta}, \boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{R} \boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x}, \tag{3.12}$$

where $\boldsymbol{R}$ is a positive-semidefinite matrix and $\boldsymbol{b}$ is a vector, together constituting the parameter $\boldsymbol{\theta} = \{\boldsymbol{R}, \boldsymbol{b}\}$. The feasible set for every entry of $\boldsymbol{x}$ is the interval $[0, 1]$, ensuring that $\mathcal{X}$ is a D-dimensional convex set.

To demonstrate the functionality of the proposed approach as broadly as possible, we do not posit any structure for $\boldsymbol{R}$ and $\boldsymbol{b}$. Rather, $\boldsymbol{R}$ is generated as

$$\boldsymbol{R} = \sum_{i=0}^{I-1} \boldsymbol{r}_i \boldsymbol{r}_i^\top \tag{3.13}$$

with the entries of $\boldsymbol{r}_i$, as well as those of $\boldsymbol{b}$, drawn uniformly within $[-1, 1]$. To solve this parametric optimization using the introduced framework, the loss function is defined as

$$L_{\mathsf{QP}} = \underbrace{\boldsymbol{x}^\top \boldsymbol{R} \boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x}}_{\text{Objective}} + \underbrace{100 \sum_{k=0}^{D-1} \big([-x_k]^+ + [x_k - 1]^+ \big)}_{\text{Constraints}}, \tag{3.14}$$

where $[z]^+ = \max(0, z)$ and the multiplier of 100 applied to the constraint portion is the result of cross-validation. By increasing it, the (very small)

Table 3.1: NN settings for QP.

	Input layer	Hidden layer	Output layer
Neurons	900	500	$D = 10 + 10$ or $30 + 30$
Activation function	RLU	RLU	Linear
Regularization	L2 norm $\lambda = .001$	L2 norm $\lambda = .001$	L2 norm $\lambda = .001$

probability that the constraints are violated and that the solution thus falls outside the feasible set could be reduced further, at the expense of the objective value. This issue is tackled in [12] within the context of a wireless communications problem.

3.4.2 Learning Stage

The learning pipeline employed to train the NN is depicted in Fig. 3.1 and the NN's parameters are summarized in Table 5.1. A single hidden layer is featured, as additional ones have not been found to improve the performance—yet the training does become longer. With D dimensions, there are $D \times D$ dimensional interactions and thus the number of neurons in the input layer should be at least D^2 while the number of neurons in the hidden layer should be between D and D^2.

After centering and scaling, $\boldsymbol{\theta}$ is fed to a feature-extracting input layer equipped with rectified linear unit (RLU) activation functions. A hidden layer then processes the data also via RLUs, and an output layer with linear activation functions generates values $\boldsymbol{x}$. To improve the dynamic range as well as the sensitivity to small values, these outputs are taken to be in log scale, yet this restricts them to being positive. This limitation is circumvented by allowing the NN to output two distinct vectors, $\boldsymbol{x}^+$ and $\boldsymbol{x}^-$, both in log scale, which are linearized into positive and negative values, respectively, and then summed. With that, the NN can produce any real vector $\boldsymbol{x}$ while preserving the advantages of an internal log rep-

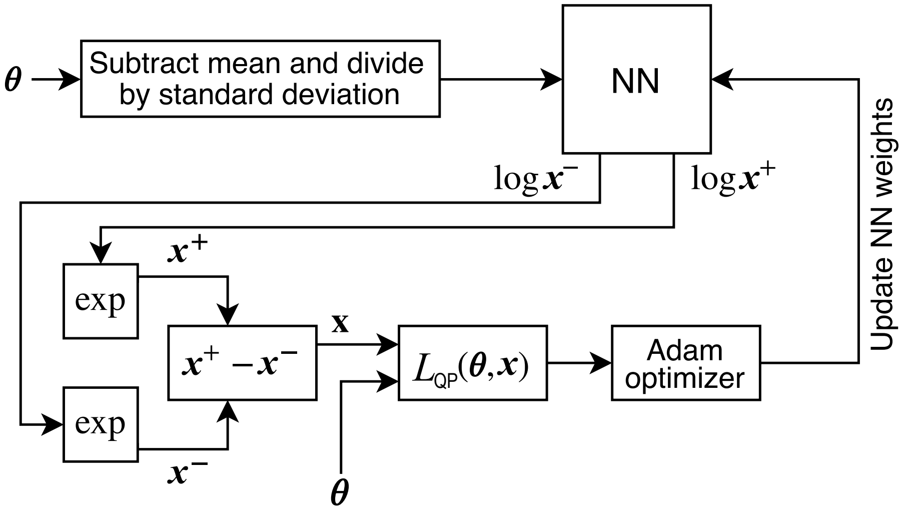


Figure 3.1: Learning pipeline.

resentation. From $\boldsymbol{\theta}$ and from the output $\boldsymbol{x}$, the loss L_{QP} is quantified and an Adam optimizer—a standard algorithm that updates NN weighs iteratively [18]—is applied to minimize it. To avoid oscillations around local optima during the weight adjustment, the learning rate, i.e., the amplitude of the gradient steps, is reduced gradually from 0.001 down to 0.0001. And, to prevent overfitting, L2-norm regularization is used in conjunction with the Adam optimizer. Precisely, a portion $\lambda = 0.001$ of the L2 norm of the weights is added to the loss.

In order to streamline the learning, rather than a single large database, we generate multiple small databases. Specifically, $M = 500$ databases of 12800 parameter realizations are generated and, over each such database, 100 updates of the NN weights take place; every update involves a randomly selected batch of 128 realizations. Altogether, $12800M$ realizations are produced for learning purposes, and the NN weights are updated $100M$ times.

3.4.3 Performance

Once the learning stage has concluded, the inference of solutions can take place. To evaluate the performance, an ensemble of realizations of the parameter $\boldsymbol{\theta}$ is produced; each realization is fed to the NN, which outputs the corresponding inference of the QP solution. Depicted in Fig. 3.2 is the cumulative distribution function (CDF) of the objective value, $\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{\theta}, \boldsymbol{x})$, for a constrained QP with either $D = 10$ or $D = 30$ dimensions and with $I = 5$ in (3.13). The match between the solutions produced by the NN and by a convex solver [19] is very satisfying for $D = 30$, when the number of neurons in the input layer is tight at its minimum value of D^2, while the match is absolute for $D = 10$, when excess neurons are available therein.

3.4.4 Computational Cost

Relative to a convex solver, the computational cost of the unsupervised-learning approach scales better with the dimensionality of the QP. Likewise, its training scales better than it would if the learning were supervised, as that would require solving the QP for each individual training parameter.

For the QP herein invoked to illustrate the performance, however, the computational savings and the improvement in scalability are modest. It is for more intricate problems such as the ones tackled in [11, 12] that the advantage becomes prominent, hence its assessment is deferred to these sequels.

3.5 Conclusion

An unsupervised NN-based procedure to tackle parametric optimizations over convex sets has been presented and put to use in the classic problem of constrained QP. The results match those produced by convex solvers with extreme accuracy. In the follow-up letters [11] and [12], more involved problems motivated by wireless communications are tackled and major advantages in computational scalability are revealed relative to convex solvers. For some of these problems, furthermore, the feasible set is convex but the objective function is not.

We believe that the restriction of the feasible set being convex could be lifted, at least in those cases in which the existence of a local minimizer can be established [20], and this is an interesting avenue for subsequent research. Also suggestive of further work is that, while computationally more scalable than convex solvers (when applicable) and appearing to perform well even with nonconvex objectives, the presented approach still suffers from weaknesses:

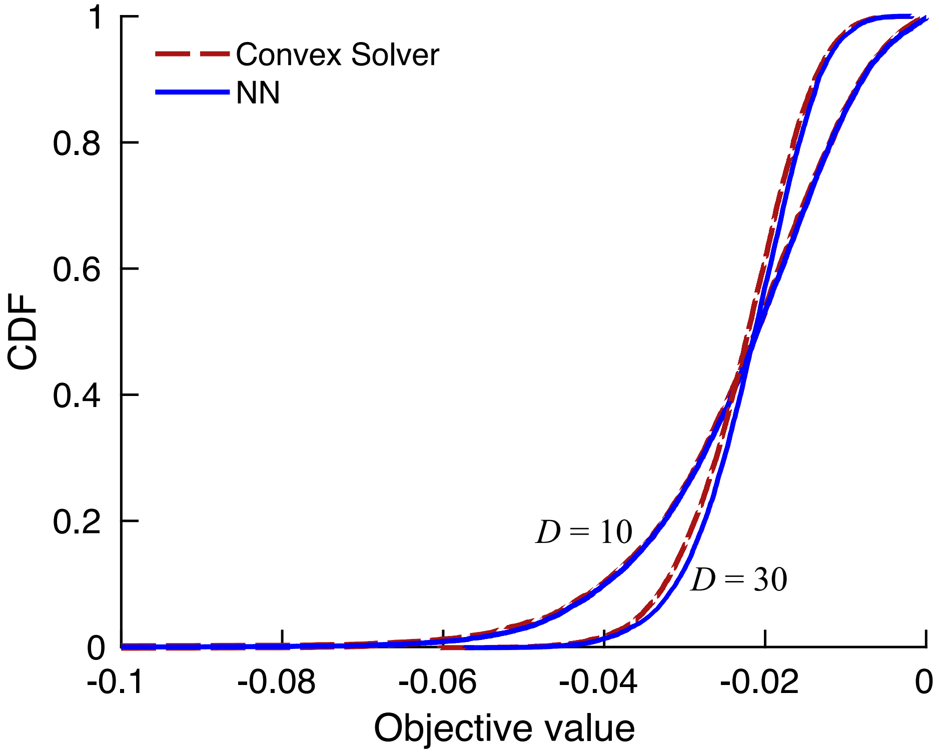


Figure 3.2: CDF of $\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{\theta}, \boldsymbol{x})$ for a QP with either $D = 10$ or $D = 30$ dimensions. The distribution is induced by that of $\boldsymbol{\theta} = \{\boldsymbol{R}, \boldsymbol{b}\}$ with $I = 5$.

- The NN is fully connected, hence the number of neuronal interconnects grows rapidly with the dimensionality of the parameter and optimization vectors, and so does the number of training samples.

- Retraining needs to take place whenever those dimensions change.

The first issue can be mitigated by exploiting the structure of each problem, say the existence of entries within $\boldsymbol{\theta}$ that are zero (or small enough to be negligible); this is often the case in the problems confronted in [11, 12]. Then, a non-fully-connected NN could be employed, at the expense of generality.

As of the second issue, one idea would be to dimension the problem for its largest possible size and then train with some of the dimensions randomly zeroed out [21]. Alternatively, a modular NN could be considered, and interesting ideas in this direction are propounded in [22, 23], also in the context of wireless communications. Modular NN designs that could be applied to generic parametric optimization would be a welcome proposition.

Acknowledgment

The invitation from the Editor-in-Chief, the handling of the Associated Editor, and the constructive feedback from the anonymous reviewers, as well as from Dr. Giovanni Geraci, are all gratefully acknowledged.

References

[1] A. V. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Academic Press, New York, 1983.

[2] G. Bank, J. Guddat, D. Klatte, B. Kummer, and D. Tammer, *Nonlinear Parametric Optimization*, Birkhauser Verlag, Basel, 1983.

[3] F. Bonnans and A. Shapiro, *Perturbation analysis of optimization problems*, Springer Series in Operations Research, Springer-Verlag, New York, 2000.

[4] D. Klatte and B. Kummer, *Nonsmooth equations in Optimization*, Kluwer Ac. Publishers Dortrecht, 2002.

[5] J. Acevedo and E. N. Pistikopoulos, "A parametric MINLP algorithm for process synthesis problems under uncertainty," *Ind. Eng. Chem. Res.*, vol. 35, pp. 147–158, 1996.

[6] V. Dua and E. N. Pistikopoulos, "An outer-approximation algorithm for the solution of multiparametric MINLP problems," *Computers them. Engng*, vol. 22, pp. S955–S958, 1998.

[7] A. Rantzer, "Dynamic dual decomposition for distributed control," in *Proceedings of the American Control Conference*, 2009, p. 884–888.

[8] P. Patrinos and H. Sarimveis, "Convex parametric piecewise quadratic optimization: Theory and algorithms," *Automatica*, vol. 47, pp. 1770–1777, 2011.

[9] L. Sanguinetti, A. Zappone, and M. Debbah, "Deep learning power allocation in massive MIMO," in *Asilomar Conf. Signals, Systems, and Computers*, 2018, pp. 1257–1261.

[10] T. Van Chien, E. Björnson, and E. G. Larsson, "Sum spectral efficiency maximization in massive MIMO systems: Benefits from deep learning," *CoRR*, vol. abs/1903.08163, 2019.

[11] R. Nikbakht, A. Jonsson, and A. Lozano, "Unsupervised learning for cellular power control," *IEEE Commun. Letters*, vol. 24, 2020.

[12] R. Nikbakht, A. Jonsson, and A. Lozano, "Unsupervised learning for C-RAN power control and power allocation," *IEEE Commun. Letters*, vol. 24, 2020.

[13] G. Cybenko, "Approximations by superpositions of sigmoidal functions," *Mathematics of Control, Signals, and Systems*, vol. 2(4), pp. 303–314, 1989.

[14] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[15] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Networks*, vol. 4(2), pp. 251–257, 1991.

[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323(6088), pp. 533–536, 1986.

[17] F. Liang, C. Shen, W. Yu, and F. Wu, “Towards optimal power control via ensembling deep neural networks,” *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1760–1776, 2019.

[18] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[19] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” `http://cvxr.com/cvx`, Mar. 2014.

[20] G. Still, “Lectures on parametric optimization: An introduction,” *Optimization Online*, 2018.

[21] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, “Adaptive neural signal detection for massive MIMO,” *IEEE Trans. Wireless Commun.*, 2020.

[22] W. Cui, K. Shen, and W. Yu, “Spatial deep learning for wireless scheduling,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1248–1261, 2019.

[23] W. Lee, M. Kim, and D.-H. Cho, “Deep power control: Transmit power control scheme based on convolutional neural network,” *IEEE Commun. Letters*, vol. 22, no. 6, pp. 1276–1279, 2018.

Chapter 4

Unsupervised Learning for Cellular Power Control

Rasoul Nikbakht, Student Member, IEEE, Anders Jonsson, Angel Lozano, Fellow, IEEE

Abstract— **This paper applies a feedforward neural network trained in an unsupervised fashion to the problem of optimizing the transmit powers in cellular wireless systems. Both uplink and downlink are considered, with either centralized or distributed power control. Various objectives are entertained, all of them such that the problem can be cast in convex form. The performance of the proposed procedure is very satisfactory and, in terms of computational cost, the scalability with the system dimensionality is markedly superior to that of convex solvers. Moreover, the optimization relies on directly measurable channel gains, with no need for user location information.**

Index Terms— **Machine learning, neural networks, unsupervised learning, power control, cellular systems**

R. Nikbakht, A. Jonsson, and A. Lozano are with Univ. Pompeu Fabra, 08018 Barcelona (e-mail: angel.lozano@upf.edu). This work was supported by the European Research Council under the H2020 Framework Programme/ERC grant agreement 694974, by the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502) as well as by MINECO's Projects RTI2018-102112 and RTI2018-101040, and by the ICREA Academia Program.

4.1 Introduction

A preceding letter proposed leveraging the expressive power of neural networks (NNs) to solve large-dimensional parametric optimizations [1]. In this follow-up letter, we apply this approach to the all-important problem of power control in cellular systems. In both uplink and downlink, power control is instrumental to prevent huge performance disparities among users. By defining suitable loss functions, it is possible to optimize the transmit powers so as to balance the user signal-to-interference-plus-noise ratios (SINRs) as desired. The more balanced the SINRs, the more fairness—customarily at the expense of a reduced average performance.

The use of NNs for cellular power control has been explored under supervised, semi-supervised, or outright unsupervised learning, with promising performance [2–6]. The unsupervised framework avoids the taxing precomputation of labeled training data for every conceivable environment and speeds up the retraining whenever there are changes in the system.

A relevant attribute of the supervised power control in [2] is that the input parameters are the user locations, from which a mapping is established to the transmit powers. Since location information could—especially indoors—be unreliable or outright unavailable, our designated parameters are the channel gains, which can be measured directly and reliably. However, in contrast with [4–6], we desire for the power control to respond only to the *large-scale* behavior of the channel gains, i.e, to their local-average values, and not to the *small-scale* fluctuations that occur on scales of milliseconds (in time) and hundreds of kilohertz (in frequency). This ensures relatively stable transmit powers, leaving the small-scale swings to be dealt with by link adaptation [7].

4.2 Cellular System Model

In its basic abstraction, a cellular system consists of N cells, each served by an AP. On a given time-frequency signaling resource, a single user is active within each cell. Both APs and users are equipped with an omnidirectional antenna.

The formulation in the sequel applies regardless of how the APs are positioned and, to be on the same footing as the cell-free systems considered in [8], we draw those positions uniformly at random. The Voronoi region of each AP then spawns a cell, and one user is dropped uniformly at random therein. Every user communicates with its same-cell AP while interfering with the rest of APs.

4.2.1 Channel Features

Signals are subject to distance-dependent power decay with exponent $\eta = 3.8$, and this decay gives rise to a large-scale channel gain $G_{n,k}$ between the kth user and the nth AP. Correspondingly, that link's signal-to-noise ratio is

$$\mathsf{SNR}_{n,k} = \frac{G_{n,k} P}{\sigma^2} \tag{4.1}$$

with P the maximum transmit power and σ^2 the noise power.

Recent results have shown that shadowing renders cellular systems approximately Poisson-like from the vantage of any user [9], a behavior that sharpens as the shadowing intensifies, but that is precise already for shadowing strenghths of interest [10]. Relying on these findings, APs and users are positioned uniformly at random, with shadowing implicitly embedded in that randomness.

For the sake of a cleaner notation, P/σ^2 is taken as equal for uplink and downlink, but asymmetries could be had by simply discriminating the respective SNR variables. We set P/σ^2 such that $\mathsf{SNR}_{n,k} = 20$ dB at a distance d, where d would be the inter-AP spacing if the network were arranged as a hexagonal grid. Under reasonable values for P and σ^2, this is compatible with a dense microcellular deployment [11].

In addition to $G_{n,k}$, the channel that connects the kth user with the nth AP includes a small-scale fading coefficient $h_{n,k} \sim \mathcal{N}_{\mathbb{C}}(0, 1)$, independent across users and APs. Each fading coefficient is presumed known by the respective receiver, an amply justified premise in cellular systems [12, ch. 4].

4.2.2 Uplink

In the uplink, the nth AP is only interested in the signal from the nth user, the one in its cell, while the rest of users constitute interference. The observation at such nth AP is

$$y_n = \sqrt{G_{n,n}} h_{n,n} \sqrt{p_n P} s_n + \sum_{k \neq n} \sqrt{G_{n,k}} h_{n,k} \sqrt{p_k P} s_k + v_n, \tag{4.2}$$

where s_n is the unit-power symbol emitted by user n whereas $p_n \in [0,1]$ is its power control coefficient and $v_n \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2)$ is the noise. The SINR for the nth user is then

$$\begin{aligned} \mathsf{sinr}_n^{\mathsf{UL}} &= \frac{p_n \, G_{n,n} \, |h_{n,n}|^2}{\sum_{k \neq n} p_k \, G_{n,k} \, |h_{n,k}|^2 + \sigma^2 / P} && (4.3) \\ &= \frac{p_n \, \mathsf{SNR}_{n,n} \, |h_{n,n}|^2}{\sum_{k \neq n} p_k \, \mathsf{SNR}_{n,k} \, |h_{n,k}|^2 + 1} && (4.4) \end{aligned}$$

and its expectation over the small-scale fading yields the operationally meaningful local-average SINR [7], namely

$$\mathbb{E}\big[\mathsf{sinr}_n^{\mathsf{UL}}\big] = \mathbb{E}\left[\frac{p_n \, \mathsf{SNR}_{n,n} \, |h_{n,n}|^2}{\sum_{k \neq n} p_k \, \mathsf{SNR}_{n,k} \, |h_{n,k}|^2 + 1}\right]. \tag{4.5}$$

In order to formulate loss functions that, as desired, drive the NN based only on large-scale quantities, it is convenient to have an SINR expression that depends on those quantities without the cumbersome outer expectation over the small-scale fading. To that end, replacing $|h_{n,n}|^2$ and $|h_{n,k}|^2$ by their expected value (unity), we obtain the very accurate proxy

$$\mathsf{SINR}_n^{\mathsf{UL}} = \frac{p_n \, \mathsf{SNR}_{n,n}}{\sum_{k \neq n} p_k \, \mathsf{SNR}_{n,k} + 1}. \tag{4.6}$$

4.2.3 Downlink

In the downlink, the local-average SINR at user n is the dual of (4.5), precisely

$$\mathbb{E}\big[\mathsf{sinr}_n^{\mathsf{DL}}\big] = \mathbb{E}\left[\frac{p_n \, \mathsf{SNR}_{n,n} \, |h_{n,n}|^2}{\sum_{k\neq n} p_k \, \mathsf{SNR}_{k,n} \, |h_{k,n}|^2 + 1}\right] \tag{4.7}$$

and its large-scale proxy is

$$\mathsf{SINR}_n^{\mathsf{DL}} = \frac{p_n \, \mathsf{SNR}_{n,n}}{\sum_{k\neq n} p_k \, \mathsf{SNR}_{k,n} + 1}. \tag{4.8}$$

4.3 Loss Functions

Armed with suitable proxies for the local-average SINRs, loss functions of interest can now be formulated. First, we consider a max-min loss, attractive when the sole focus is fairness, and we suitably soften the function to allow relaxing that focus individually for each user. Then, we consider a max-product loss, enticing when the objective is to favor the average performance without a complete sacrifice in fairness.

4.3.1 Soft Max-Min

The soft max-min function is

$$\begin{aligned} L_{\mathsf{MM}} = \frac{1}{N}\Bigg[& \underbrace{\sum_{n=0}^{N-1} \exp\left(\frac{\alpha_n}{(\mathsf{SINR}_n + 0.01)^{0.4}}\right)}_{\text{Objective}} \\ & + 0.1 \underbrace{\sum_{n=0}^{N-1} [p_n - 1]^+}_{\text{Constraints}} \Bigg] \end{aligned} \tag{4.9}$$

where $\{\alpha_n\}$ are regulating factors and $[z]^+ = \max(0, z)$. The cross-validation process described in [1] yields a multiplier of 0.1 for the constraints, relative to the objective. Readers interested in the probability that some constraint is violated are referred to [8], where such probability and its impact on the ultimate performance is quantified for the more involved constraints that arise in centralized radio-access networks.

As $\{\alpha_n\}$ grow large, L_{MM} becomes dominated by the lowest SINR and the optimization hardens to a maximization of such smallest SINR, i.e., to a max-min policy. Conversely, for decreasing $\{\alpha_n\}$, this max-min behavior softens as SINRs other than the lowest become progressively relevant.

The small offset 0.01 added to SINR_n avoids having the loss being dragged down by users below -20 dB and prevents numerical problems in the NN learning stage. In turn, the exponent 0.4 compresses the dynamic range, improving the high-SNR performance and making the learning more stable.

4.3.2 Max-Product

Our second loss function relates to the product of the SINRs, whose maximum is a desirable operating point in wireless systems [12, sec. 7.5]. The maximization of $\Pi_{n=0}^{N-1} \mathsf{SINR}_n$, or equivalently of its logarithm, amounts to the minimization of

$$L_{\mathsf{MP}} = \frac{1}{N} \Bigg[\underbrace{\sum_{n=0}^{N-1} \beta_n \log\!\left(0.01 + \frac{1}{\mathsf{SINR}_n + 0.01}\right)}_{\text{Objective}} + \underbrace{0.1 \sum_{n=0}^{N-1} [p_n - 1]^+}_{\text{Constraints}} \Bigg] \tag{4.10}$$

where the multiplier applied to the constraints is again 0.1.

When the regulating factors $\{\beta_n\}$ are equal, the combination $\{\mathsf{SINR}_n\}$ that minimizes L_{MP} exhibits satisfactory properties in terms of the trade-

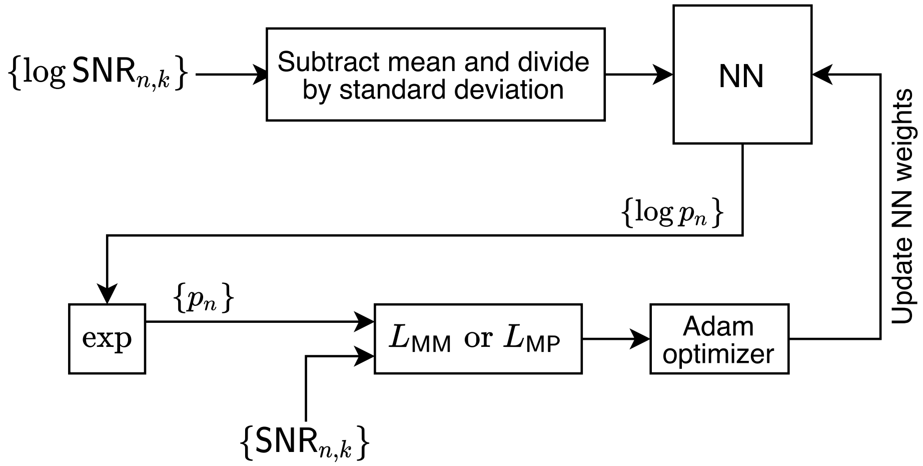

$\{\log \mathsf{SNR}_{n,k}\}$
Subtract mean and divide by standard deviation
NN
Update NN weights
$\{\log p_n\}$
exp
$\{p_n\}$
L_{MM} or L_{MP}
Adam optimizer
$\{\mathsf{SNR}_{n,k}\}$

Figure 4.1: Learning pipeline.

off between average performance and fairness [12, sec. 7.5]. And, by tinkering with $\{\beta_n\}$, specific users could be afforded higher performance priorities.

Again, the offset shifting SINR_n by 0.01 avoids it being pulled down by users below -20 dB while a second offset added to $\frac{1}{\mathsf{SINR}_n+0.01}$ lessens the pull of users above 20 dB.

4.4 Centralized Power Control

Connecting back with the parametric optimization description in [1], the input parameter $\boldsymbol{\theta}$ consists of $\{\mathsf{SNR}_{n,k}\}$ while the optimization vector $\boldsymbol{x}$ packs the N power coefficients $\{p_n\}$, each constrained to the convex interval $[0, 1]$. The loss function is given by either L_{MM} or L_{MP}. And, as the transmit power in each cell is adjusted on the basis of the SNRs for all the users in the system, the power control is centralized.

Table 4.1: NN settings.

	Input layer	Hidden layer	Output layer
Neurons	1200	200	$N = 30$
Activation function	RLU	RLU	Linear
Regularization	L2 norm $\lambda = .001$	L2 norm $\lambda = .001$	L2 norm $\lambda = .001$

4.4.1 Learning Stage

The learning pipeline is illustrated in Fig. 4.1 while the parameters of the NN are summarized in Table 5.1. For preprocessing purposes, $\{\mathsf{SNR}_{n,k}\}$ are first converted to $\{\log \mathsf{SNR}_{n,k}\}$ and subsequently rendered zero-mean and unit variance. The processing then starts through an input layer equipped with rectified linear unit (RLU) activation functions. After feature extraction by this input layer, a hidden layer processes the data also via RLUs, and an output layer with linear activation functions generates power coefficients in log scale; this guarantees positive outputs and averts numerical problems. From the SNRs and the corresponding NN outputs, the loss function of choice is quantified and an Adam optimizer is applied to minimize it. To avoid oscillations around local optima during the weight adjustment, the learning rate is reduced gradually from 0.001 down to 0.0001. And, to prevent overfitting, L2-norm regularization is employed in conjunction with the Adam optimizer: a portion $\lambda = 0.001$ of the L2 norm of the weights is added to the loss.

To streamline the learning, rather than a single large database, 1000 databases of 10000 system realizations are generated and, over each, 10 updates of the NN weights take place; each update relies on a randomly selected batch of 1000 realizations. The initialization is also random.

As an example, the uplink learning curve for the soft max-min loss function with $\alpha_n = 1$ is provided in Fig. 4.2.

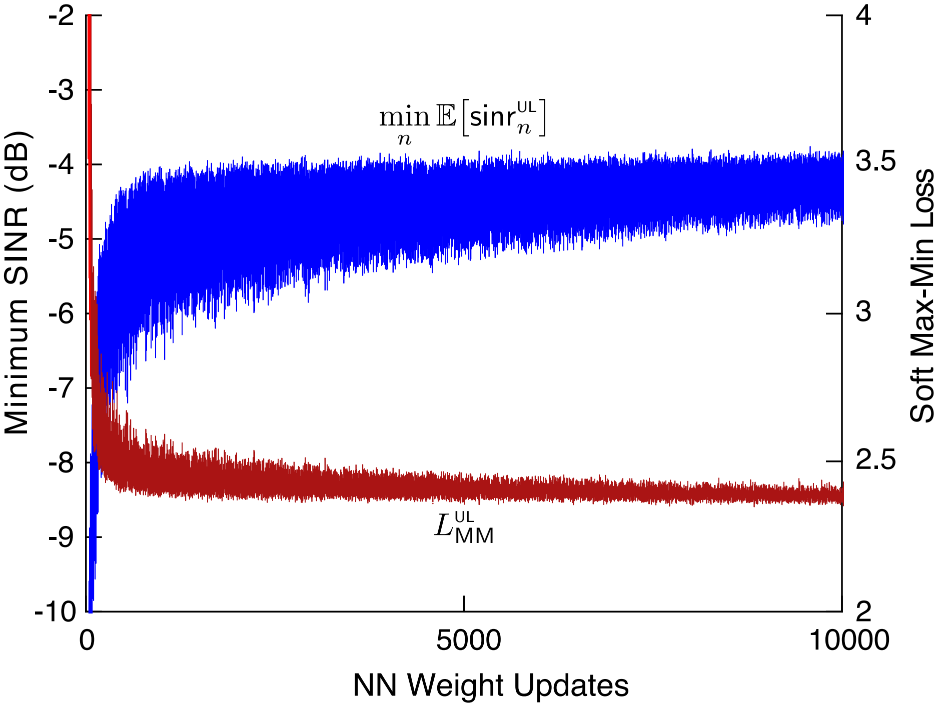


Figure 4.2: Uplink soft max-min learning curve with $\alpha_n = 1$: loss and minimum local-average SINR as a function of the NN weight updates.

4.4.2 Performance Evaluation

For both of the introduced loss functions, the cellular power control problem admits convex formulations in both uplink and downlink. Thus, performance benchmarks can be produced with an off-the-shelf convex solver [13].

We consider a system with $N = 30$ cells, and equip the NN's input layer with 1200 neurons, somewhat above the minimum of N^2. The performance is evaluated by means of the cumulative distribution function (CDF) of local-average SINRs over the ensemble of possible user and AP positions. The NN and the convex solver are driven by the proxies in (4.6) and (4.8), for uplink and downlink respectively, while the performance with the obtained powers is assessed by means of the actual local-average

SINRs in (4.5) and (4.7).

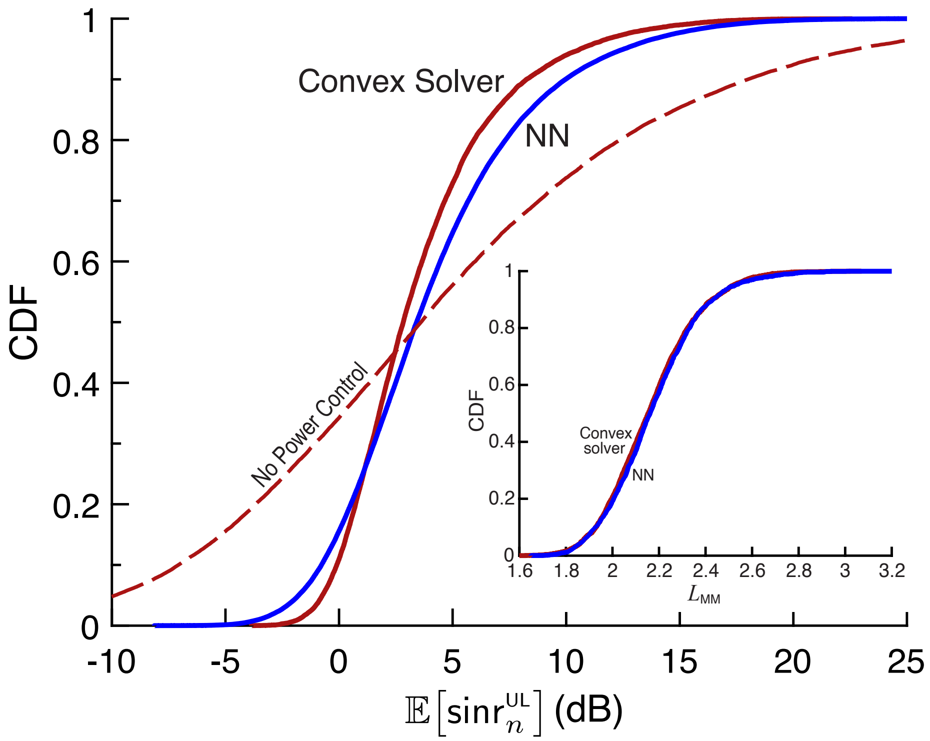


Figure 4.3: CDF of $\mathbb{E}\left[\mathsf{sinr}_n^{\mathsf{UL}}\right]$ for the soft max-min loss function ($\alpha_n = 1$): NN vs convex solver, alongside the baseline without power control ($p_n = 1$). In the inset, CDF of L_{MM} for the NN and the convex solver.

Presented in Fig. 4.3 is the uplink performance for the soft max-min loss function with $\alpha_n = 1$. The NN yields transmit powers that, relative to the convex solver, favor high-SNR users slightly over their low-SNR brethren. In terms of the loss function, the NN is uniformly shy of the convex solver, but the agreement is extremely satisfying. The figure also depicts the baseline performance in the absence of power control, which confirms the importance of optimizing the transmit powers to avoid enormous disparities among users. For the max-product loss function, Fig. 4.4 reveals an even better match between the NN and the convex solver.

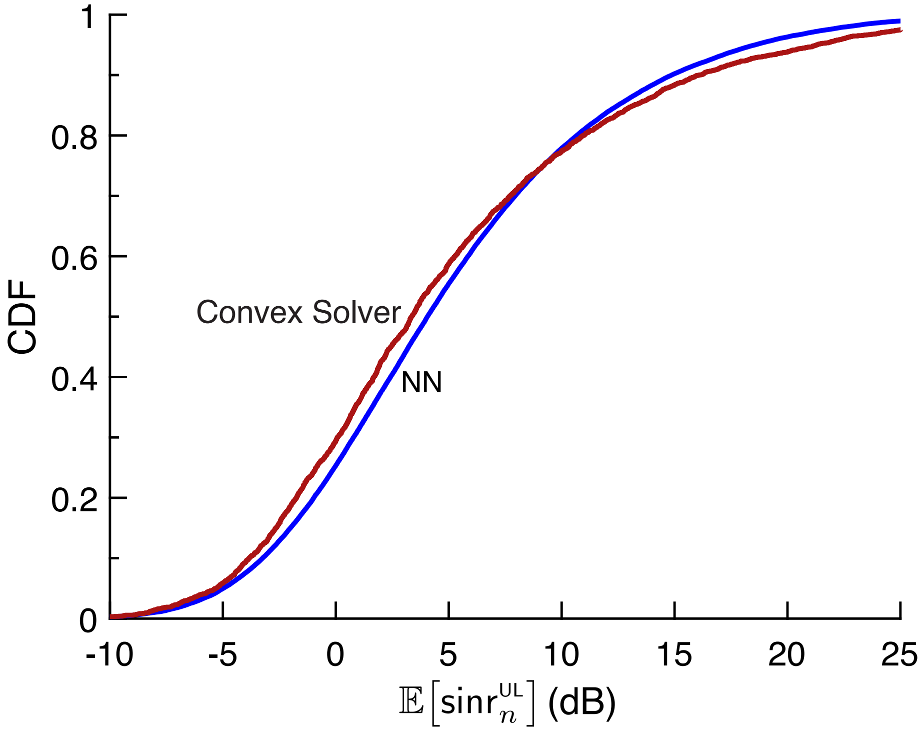


Figure 4.4: CDF of $\mathbb{E}\left[\mathsf{sinr}_n^{\mathsf{UL}}\right]$ for the max-product loss function ($\beta_n = 1$): NN vs convex solver.

The CDFs of local-average SINRs again cross over while, in terms of the loss function, the NN is uniformly shy of the convex solver.

In the downlink (see Figs. 4.5 and 4.6), the importance of power control is somewhat toned down because the intercell interference distribution is more benign [14]. Indeed, a contrast of Figs. 4.5 and 4.6 evidences that a non-power-controlled transmission approximates fairly closely the max-product solution.

4.4.3 Computational Cost

As a measure of the computational cost, we invoke the running time on a common computational platform. And, to err on the side of caution, a

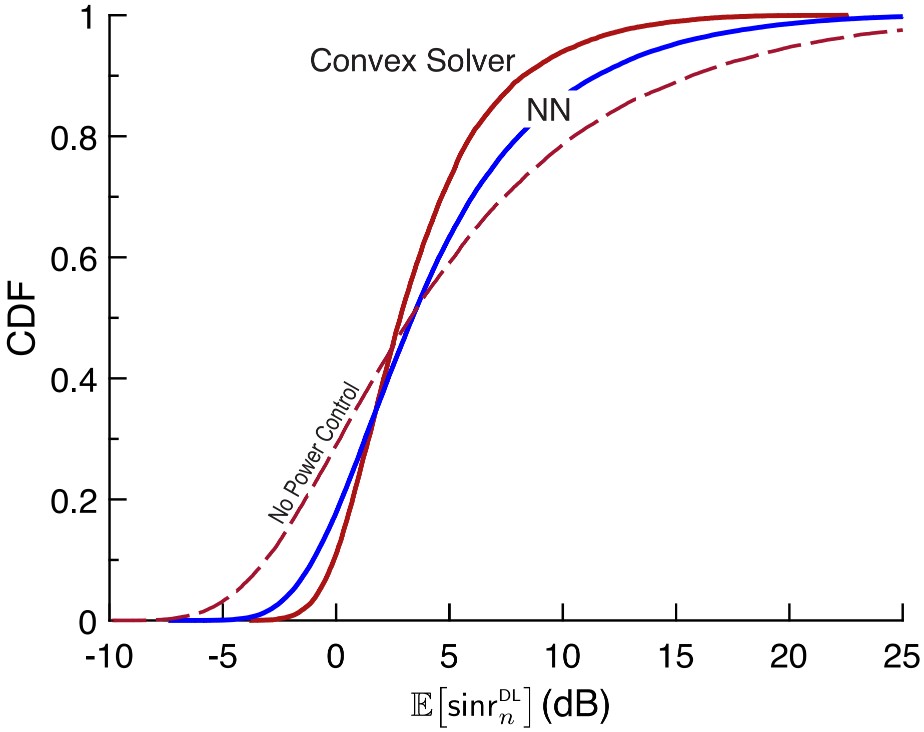


Figure 4.5: CDF of $\mathbb{E}\left[\mathsf{sinr}_n^{\mathsf{DL}}\right]$ for the soft max-min loss function ($\alpha_n = 1$): NN vs convex solver. Also shown is the baseline performance without power control, i.e., with $p_n = 1$.

CPU-based platform befitting the convex solver is chosen.

Each NN inference is in essence a matrix multiplication, over two orders of magnitude faster than a convex solver for our 30-cell system. For a growing number of cells, this advantage is likely to increase further.

In terms of learning, which needs to take place upon changes in the system or the environment, it is interesting to contrast the learning time of our unsupervised NN with that of a supervised NN of the same dimensions. While the training effort per parameter sample is essentially the same in both cases, a supervised NN would require producing those samples by solving the corresponding optimization problems in the first place, again resulting in the over-two-order-of-magnitude advantage for

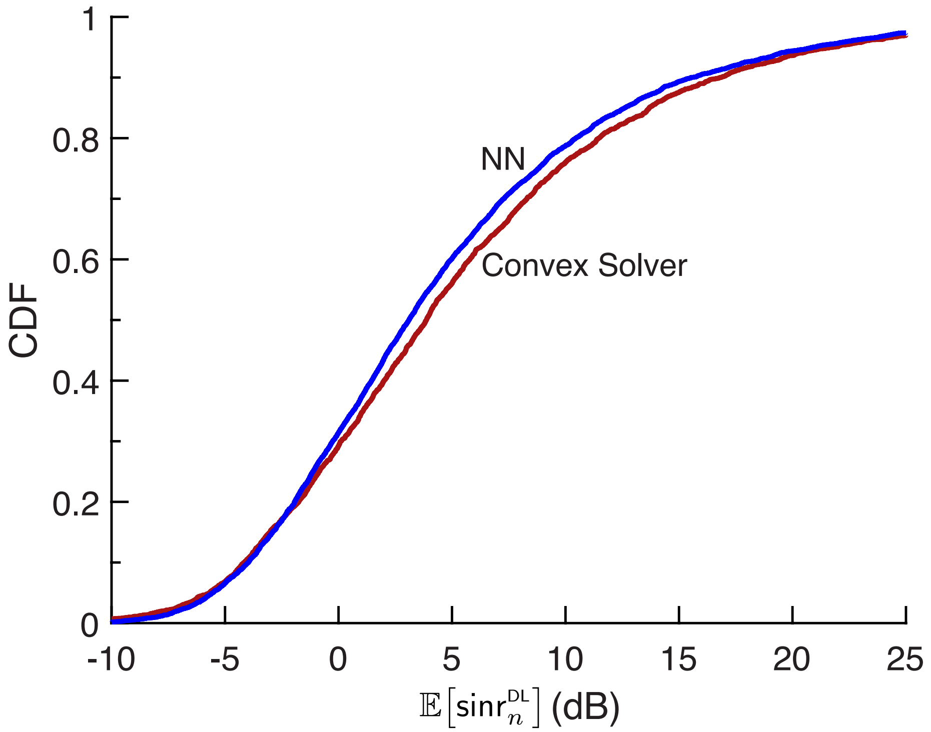


Figure 4.6: CDF of $\mathbb{E}\left[\mathsf{sinr}_n^{\mathsf{DL}}\right]$ for the max-product loss function ($\beta_n = 1$): NN vs convex solver.

the example at hand. This primacy could be tempered slightly by the possibility of reusing training samples on the part of the supervised scheme, allowing it to cope with a fraction of the data, but in a worst-case scenario the unsupervised approach would retain a net advantage of over one order of magnitude—which would then rise with the number of cells. (Allowing the NN to train on a more apt GPU-based platform, this superiority would increase by yet another order of magnitude.)

4.5 Distributed Power Control

While centralizing the power control is advantageous from a performance standpoint, allowing it to operate distributedly—each cell on its own—is preferable from an implementation perspective. We can shift towards distributedness by restricting the input parameter $\boldsymbol{\theta}$ to contain, rather than $\{\mathsf{SNR}_{n,k}\}$, only $\{\mathsf{SNR}_{n,n}\}$. The transmit powers are then optimized based solely on same-cell SNRs, and owing to the reduced parameter space, the training is considerably more expeditious.

For distributed power control, an additional benchmark exists, namely the so-called fractional power control policy [15]. This is in fact the mainstream approach to power control in current cellular networks, with the remarkably simple form

$$p_n \propto \frac{1}{\mathsf{SNR}_{n,n}^{\vartheta}}, \tag{4.11}$$

where ϑ determines the extent to which fairness is promoted [16]; typical values in 4G and 5G systems are $\vartheta \approx 0.5$–0.7, with lower values favoring the average performance while higher values foster fairness and cell-edge robustness. The proportionality factor in (4.11) should be such that $p_n \in [0, 1]$.

Presented in Fig. 4.7 is the comparison between the NN-based approach (with soft max-min and $\alpha_n = 1$) and fractional power control (with $\vartheta = 0.7$). It can be surmised that:

- Distributed power control is only modestly worse than centralized power control (cf. Figs. 4.3 and 4.7).

- Fractional power control is indeed extremely effective, its performance almost identical to that of the NN—despite the fact that the NN has access to the same-cell SNRs for the entire system while fractional power control truly operates on a cell-by-cell basis.

This latter point invites a dual remark. On the one hand, the NN is useful to certify the effectiveness of fractional power control and to calibrate the equivalence of the parameter ϑ in terms of any well-defined loss function. On the other hand, a solution far simpler and equally performing to the NN is seen to exist—less general, but emanating from the core structure of the problem at hand. Precisely, fractional power control was derived analytically as the strategy that minimizes the variance of the SIR (in dB) in a two-cell setup [15].

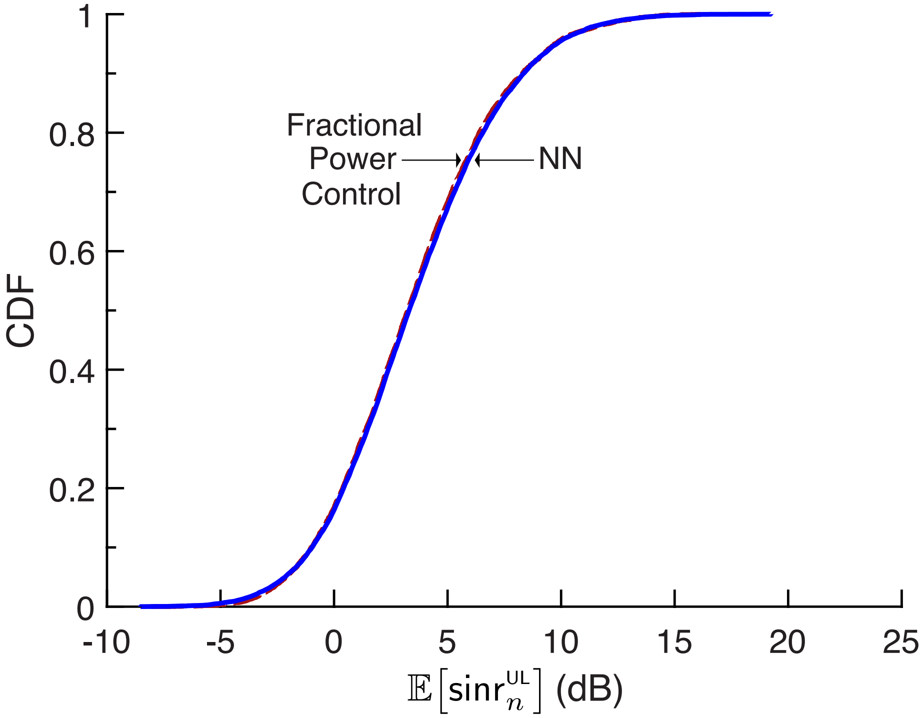


Figure 4.7: CDF of $\mathbb{E}\left[\mathsf{sinr}_n^{\mathsf{UL}}\right]$: NN for soft max-min ($\alpha_n = 1$) in solid versus fractional power control ($\vartheta = 0.7$) in dashed.

4.6 Summary

The unsupervised-learning approach to parametric optimization described in [1] is very effective for cellular power control, adding to the body of works that leverage NNs to tackle wireless communication problems [2–

5, 17–21].

With a view to having suitable benchmarks, only objectives leading to convex formulations have been considered. Other objectives can be defined that do not admit convex forms, in particular the maximization of weighed sum of the spectral efficiencies. Although benchmarks may then be lacking, these objectives can be tackled all the same with our approach, and instances of nonconvex objectives are given in [8] for centralized radio-access networks.

The distributed uplink setting dispenses a reminder of the fact that, whenever sound models are available that describe the structure of the problem, solutions that are simpler and even illuminating may sometimes be derived directly from those models. Learning-based approaches should complement and expand, rather than replace, such model-based derivations.

The Python code developed to produce the results in this letter is openly available [22].

Acknowledgment

The invitation from the Editor-in-Chief, the handling of the Associated Editor, and the constructive feedback from the anonymous reviewers, as well as from Dr. Giovanni Geraci, are all gratefully acknowledged.

References

[1] R. Nikbakht, A. Jonsson, and A. Lozano, "Unsupervised learning for parametric optimization," *IEEE Commun. Letters*, vol. 24, 2020.

[2] L. Sanguinetti, A. Zappone, and M. Debbah, "Deep learning power allocation in massive MIMO," in *Asilomar Conf. Signals, Systems, and Computers*, 2018, pp. 1257–1261.

[3] T. Van Chien, T. Nguyen Canh, E. Björnson, and E. G. Larsson, "Power control in cellular massive MIMO with varying user activity:

A deep learning solution," *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 5732–5748, 2020.

[4] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Processing*, vol. 66, no. 20, pp. 5438–5453, 2018.

[5] W. Lee, M. Kim, and D.-H. Cho, "Deep power control: Transmit power control scheme based on convolutional neural network," *IEEE Commun. Letters*, vol. 22, no. 6, pp. 1276–1279, 2018.

[6] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1760–1776, 2019.

[7] A. Lozano and N. Jindal, "Are yesterday's information-theoretic fading models and performance metrics adequate for the analysis of today's wireless systems?," *IEEE Comm. Magazine*, vol. 50, no. 11, pp. 210–217, Nov. 2012.

[8] R. Nikbakht, A. Jonsson, and A. Lozano, "Unsupervised learning for C-RAN power control and power allocation," *IEEE Commun. Letters*, vol. 24, 2020.

[9] B. Błaszczyszyn, M. K. Karray, and H. P. Keeler, "Wireless networks appear Poissonian due to strong shadowing," *IEEE Trans. Wireless Communications*, vol. 14, pp. 4379–4390, Aug. 2015.

[10] G. George, R. K. Mungara, A. Lozano, and M. Haenggi, "Ergodic spectral efficiency in MIMO cellular networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 2835–2849, May 2017.

[11] 3GPP TS 36.814, "Further advancements for E-UTRA physical layer aspects (Release 9)," Tech. Rep., 3GPP, Mar. 2017.

[12] R. W. Heath Jr. and A. Lozano, *Foundations of MIMO Communication*, Cambridge University Press, 2018.

[13] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[14] A. Lozano and D. C. Cox, "Integrated dynamic channel assignment and power control in TDMA mobile wireless communication systems," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 11, pp. 2031–2040, 1999.

[15] J. F. Whitehead, "Signal-level-based dynamic power control for co-channel interference management," in *IEEE Veh. Techn. Conf. (VTC'93)*, May 1993, pp. 499–502.

[16] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341–1347, Sep. 1995.

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

[18] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Letters*, vol. 7, no. 1, pp. 114–117, 2017.

[19] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1248–1261, 2019.

[20] D. Gündüz, P. de Kerret, N. D. Sidiropoulos, D. Gesbert, C. R. Murthy, and M. van der Schaar, "Machine learning in the air," *IEEE J. Sel. Areas Commun.*, vol. 37, pp. 2184–2199, Oct. 2019.

[21] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[22] R. Nikbakht, "Unsupervised NN," https://github.com/RasoulNik/UnsupervisedNN, 2020.

Chapter 5

Unsupervised Learning for C-RAN Power Control and Power Allocation

Rasoul Nikbakht, Student Member, IEEE, Anders Jonsson, Angel Lozano, Fellow, IEEE

Abstract— **This paper applies a feedforward neural network trained in an unsupervised fashion to the problem of optimizing the transmit powers in centralized radio access networks operating on a cell-free basis. Both uplink and downlink are considered. Various objectives are entertained, some leading to convex formulations and some that do not. In all cases, the performance of the proposed procedure is very satisfactory and, in terms of computational cost, the scalability is manifestly superior to that of convex solvers. Moreover, the optimization relies on directly measurable channel gains, with no need for user location information.**

Index Terms— **Neural networks, unsupervised learning, cell-free networks, ultradense networks, power control, power allocation, C-RAN**

R. Nikbakht, A. Jonsson, and A. Lozano are with Univ. Pompeu Fabra, 08018 Barcelona (e-mail: angel.lozano@upf.edu). Work supported by the European Research Council under the H2020 Framework Programme/ERC grant 694974, by the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502), and by the ICREA Academia program. Parts of this paper were presented at the 2019 IEEE Int'l Symp. Personal, Indoor & Mobile Radio Communications [1] and at the 2020 IEEE Int'l Conf.

5.1 Introduction

A preceding letter proposed leveraging the expressive power of neural networks (NNs) to solve large-dimensional parametric optimizations in an unsupervised fashion [3]. Then, in a follow-up letter, this approach was applied to power control for cellular systems [4]. In this final letter, it is further applied to the intertwined problems of power control and power allocation for centralized, possibly cloud-based, radio access networks (C-RAN). The more intricate optimizations that arise in this richer setting offer an excellent opportunity to further test the efficacy of the technique. In particular:

- In a C-RAN, the transmit power optimization is necessarily centralized and decidedly many-parametric; it runs into dimensionality issues faster than in cellular setups.
- The loss functions introduced in [4] continue to be applicable, but, in contrast with the cellular case, here they do not always admit convex forms. When they do not, the performance of our learning-based approach becomes itself a yardstick against which other solutions can be gauged.

C-RANs inherently operate on a cell-free basis: on each time-frequency resource unit, every access point (AP) communicates, rather than only with the user in its cell, with every user in the system [5]. This inherits ideas from network MIMO and takes to its natural limit the notion of cell cooperation [6]. C-RANs offer multiple advantages over their cellular counterparts, including large-scale diversity, interference handling, flexibility, and elasticity, and it is reasonable to anticipate that, at least for ultradense deployments, C-RANs might become the norm [7, 8].

As in [4], we desire for the power control to respond only to the *large-scale* behavior of the channel gains, and not to the *small-scale* fluctuations that occur on scales of milliseconds (in time) and hundreds of kilohertz (in frequency). This guarantees relatively stable transmit powers, leaving the small-scale swings to the province of link adaptation [9].

Communications [2].

5.2 C-RAN Model

On a time-frequency resource, a basic C-RAN consists of N APs and K users, all equipped with omnidirectional antennas. The positions of APs and users are uniformly random. Every AP can communicate with every user and, to render matched-filter beamforming effective, N is substantially larger than K.

5.2.1 Large-Scale Features

Signals are subject to distance-dependent power decay with exponent $\eta = 3.8$, for a large-scale channel gain $G_{n,k}$ between the kth user and the nth AP. Correspondingly, that link's signal-to-noise ratio is

$$\mathsf{SNR}_{n,k} = \frac{G_{n,k} P}{\sigma^2} \tag{5.1}$$

with P the maximum transmit power and σ^2 the noise power.

Both APs and users are positioned uniformly at random, with shadowing implicitly embedded in that randomness [10].

For the sake of a cleaner notation, P/σ^2 is taken as equal for uplink and downlink, but asymmetries could be had by simply discriminating the respective SNR variables. We set P/σ^2 such that $\mathsf{SNR}_{n,k} = 20$ dB at a distance d, where d would be the inter-AP spacing if the network were arranged as a hexagonal grid. Under reasonable values for P and σ^2, this is compatible with a dense C-RAN.

In addition to $G_{n,k}$, the channel that connects the kth user with the nth AP includes a small-scale fading coefficient $h_{n,k} \sim \mathcal{N}_{\mathbb{C}}(0, 1)$, independent across users and APs. These fading coefficients are presumed known, with comments on the impact of channel estimation provided in Section 5.5.

5.2.2 Uplink

In the uplink, the nth AP observes

$$y_n = \sum_{k=0}^{K-1} \sqrt{G_{n,k}} h_{n,k} \sqrt{p_k P} s_k + v_n, \tag{5.2}$$

where s_k is the unit-power symbol emitted by user k whereas $p_k \in [0, 1]$ is its power control coefficient and $v_n \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2)$ is the noise. With the observations of the N APs centrally combined so as to effect matched-filter beamforming for each user, the SINR of user k equals [5]

$$\mathsf{sinr}_k^{\mathsf{UL}} = \frac{p_k \left(\sum_{n=0}^{N-1} \mathsf{SNR}_{n,k} \, |h_{n,k}|^2 \right)^2}{\mathrm{den}_k} \tag{5.3}$$

with

$$\begin{aligned} \mathrm{den}_k = {} & \sum_{\ell \neq k} p_\ell \left| \sum_{n=0}^{N-1} \sqrt{\mathsf{SNR}_{n,k} \mathsf{SNR}_{n,\ell}} \, h_{n,k}^* \, h_{n,\ell} \right|^2 \\ & + \sum_{n=0}^{N-1} \mathsf{SNR}_{n,k} \, |h_{n,k}|^2. \end{aligned} \tag{5.4}$$

A subsequent expectation over the small-scale fading yields the operationally meaningful local-average SINR [9], i.e.,

$$\mathbb{E}\big[\mathsf{sinr}_k^{\mathsf{UL}}\big] = \mathbb{E}\left[\frac{p_k \left(\sum_{n=0}^{N-1} \mathsf{SNR}_{n,k} \, |h_{n,k}|^2 \right)^2}{\mathrm{den}_k} \right]. \tag{5.5}$$

To formulate loss functions that, as intended, drive the NN based only on large-scale quantities, we replace $|h_{n,k}|^2$ and $|h_{n,\ell}|^2$ by their expected value (unity), and $h_{n,k}^* h_{n,\ell}$ for $\ell \neq k$ also by their expected value (zero), obtaining the proxy

$$\mathsf{SINR}_k^{\mathsf{UL}} = \frac{p_k \left(\sum_{n=0}^{N-1} \mathsf{SNR}_{n,k} \right)^2}{\sum_{\ell \neq k} p_\ell \sum_{n=0}^{N-1} \mathsf{SNR}_{n,k} \mathsf{SNR}_{n,\ell} + \sum_{n=0}^{N-1} \mathsf{SNR}_{n,k}}. \tag{5.6}$$

Other approximations to (5.5) could also be legitimate proxies to drive the NN.

5.2.3 Downlink

Turning now to the downlink, with conjugate beamforming the nth AP transmits $\sum_{k=0}^{K-1} \sqrt{p_{n,k} P}\, h_{n,k} s_k$, where s_k is the unit-power symbol intended for user k while $p_{n,k}$ is the share of power that the nth AP devotes to such user, subject to $\sum_{k=0}^{K-1} p_{n,k} \in [0,1]$ in order for the AP's total power not to exceed P. User k observes

$$y_k = \sum_{n=0}^{N-1} \sqrt{G_{n,k}\, p_{n,k} P}\, h_{n,k}^* h_{n,k} s_k \tag{5.7}$$

$$+ \sum_{n=0}^{N-1} \sqrt{G_{n,k}\, P}\, h_{n,k}^* \sum_{\ell \neq k} \sqrt{p_{n,\ell}}\, h_{n,\ell} s_\ell + v_k, \tag{5.8}$$

from which its local-average SINR can be seen to be

$$\mathbb{E}\big[\mathsf{sinr}_k^{\mathsf{DL}}\big] = \mathbb{E}\left[\frac{\left(\sum_{n=0}^{N-1} \sqrt{\mathsf{SNR}_{n,k}\, p_{n,k}}\, |h_{n,k}|^2\right)^2}{\sum_{\ell \neq k} \left|\sum_{n=0}^{N-1} \sqrt{\mathsf{SNR}_{n,k}\, p_{n,\ell}}\, h_{n,k}^* h_{n,\ell}\right|^2 + 1}\right]. \tag{5.9}$$

Again replacing $|h_{n,k}|^2$ and $h_{n,k}^* h_{n,\ell}$ for $\ell \neq k$ by their expected values (respectively one and zero), we obtain the proxy

$$\mathsf{SINR}_k^{\mathsf{DL}} = \frac{\left(\sum_{n=0}^{N-1} \sqrt{\mathsf{SNR}_{n,k}\, p_{n,k}}\right)^2}{1 + \sum_{n=0}^{N-1} \mathsf{SNR}_{n,k} \sum_{\ell \neq k} p_{n,\ell}}. \tag{5.10}$$

In contrast with the C-RAN uplink, and with the cellular uplink and downlink, the optimization of the transmit powers in the C-RAN downlink actually subsumes two intertwined problems:

- Power control, which amounts to adjusting the total transmit power at every AP, i.e., $\sum_{k=0}^{K-1} p_{n,k}\ \forall n$.

- Power allocation, which entails dividing the power of each AP among the users, i.e., setting $p_{n,k}$ for every k, subject to the total power established for the nth AP.

This enhanced problem richness is reflected by the existence of NK downlink power coefficients (as opposed to K or N) and by the aggregate (rather than individual) nature of the constraint at each AP.

5.3 Loss Functions

Owing to the different form of the constraints in uplink and downlink, the formulation of the loss functions must be individualized for either case.

5.3.1 Soft Max-Min

For the uplink, our first loss function adopts the form

$$L_{\mathsf{MM}}^{\mathsf{UL}} = \frac{1}{K}\left[\underbrace{\sum_{k=0}^{K-1} \exp\left(\frac{\alpha_k}{(\mathsf{SINR}_k^{\mathsf{UL}} + 0.01)^{0.4}}\right)}_{\text{Objective}} + \underbrace{\sum_{k=0}^{K-1} [p_k - 1]^{+}}_{\text{Constraints}}\right] \qquad (5.11)$$

where $\{\alpha_k\}$ are regulating factors and $[z]^{+} = \max(0, z)$. As $\{\alpha_k\}$ grow large, the loss becomes dominated by the smallest SINR and a hard max-min policy emerges. Conversely, for decreasing $\{\alpha_k\}$, this max-min behavior softens as SINRs other than the smallest one enter the optimization.

The offset 0.01 added to SINR_k prevents the loss from being dragged down by users below -20 dB and avoids numerical problems in the learning stage. The exponent 0.4 compresses the dynamic range, improving the high-SNR behavior and making the learning more stable.[1]

[1] A variation of (5.11) that excludes this compression is considered in [1].

For the downlink,

$$L_{\mathsf{MM}}^{\mathsf{DL}} = \frac{1}{K}\Bigg[\underbrace{\sum_{k=0}^{K-1} \exp\left(\frac{\alpha_k}{(\mathsf{SINR}_k^{\mathsf{DL}} + 0.01)^{0.4}}\right)}_{\text{Objective}} + \underbrace{\sum_{n=0}^{N-1}\left[\sum_{k=0}^{K-1} p_{n,k} - 1\right]^+}_{\text{Constraints}}\Bigg]. \tag{5.12}$$

In both uplink and downlink, the multiplier applied to the constraints, obtained from cross-validation, is unity. The ensuing probability that the solution falls outside the feasible set, and the impact thereof, are quantified in Section 5.4.

5.3.2 Max-Product

The maximization of $\Pi_{k=0}^{K-1} \mathsf{SINR}_k$, or equivalently of its logarithm, amounts to the minimization of

$$L_{\mathsf{MP}}^{\mathsf{UL}} = \frac{1}{K}\Bigg[\underbrace{\sum_{k=0}^{K-1} \beta_k \log_e\left(0.01 + \frac{1}{\mathsf{SINR}_k^{\mathsf{UL}} + 0.01}\right)}_{\text{Objective}} \underbrace{+\, 0.1 \sum_{k=0}^{K-1} [p_k - 1]^+}_{\text{Constraints}}\Bigg] \tag{5.13}$$

in the uplink, and of

$$L^{\mathsf{DL}}_{\mathsf{MP}} = \frac{1}{K}\Bigg[\underbrace{\sum_{k=0}^{K-1}\beta_k \log_e\left(0.01 + \frac{1}{\mathsf{SINR}^{\mathsf{DL}}_k + 0.01}\right)}_{\text{Objective}} \\ \underbrace{+\, 0.1\sum_{n=0}^{N-1}\left[\sum_{k=0}^{K-1} p_{n,k} - 1\right]^{+}}_{\text{Constraints}}\Bigg] \tag{5.14}$$

in the downlink. Here, the constraint multiplier is 0.1. When the factors $\{\beta_k\}$ are equal, the combination $\{\mathsf{SINR}_k\}$ minimizing the above loss function offers a satisfying tradeoff between average performance and fairness. And, by regulating $\{\beta_k\}$, specific users could be afforded higher priorities.

Again, the offset shifting SINR_k by 0.01 avoids it being pulled down by users below -20 dB while a second offset added to $\frac{1}{\mathsf{SINR}_k+0.01}$ lessens the pull of users above 20 dB.

5.4 Power Control and Power Allocation

Connecting back with the description in [3], for the uplink the input parameter $\boldsymbol{\theta}$ contains the ingredients that appear in (5.6), namely $\sum_{n=0}^{N-1}\mathsf{SNR}_{n,k}$ $\forall k$ and $\sum_{n=0}^{N-1}\mathsf{SNR}_{n,k}\mathsf{SNR}_{n,\ell}$ $\forall k,\ell$, while the optimization vector $\boldsymbol{x}$ is made of the power coefficients $\{p_k\}$.[2] For the downlink, in turn, $\boldsymbol{\theta} = \{\mathsf{SNR}_{n,k}\}$ and $\boldsymbol{x} = \{p_{n,k}\}$. The loss function is given by L_{MM} or L_{MP}, as appropriate.

[2] The uplink learning could be expedited by restricting the input parameter to $\sum_{n=0}^{N-1}\mathsf{SNR}_{n,k}$ $\forall k$, i.e., the ingredients in the numerator of (5.6), disregarding those in the denominator. This alternative, explored in [1], yields an only slightly diminished performance. Relevant to this alternative, as an additional baseline, is the fractional power control for C-RANs, which relies on the same inputs [11].

Table 5.1: NN settings.

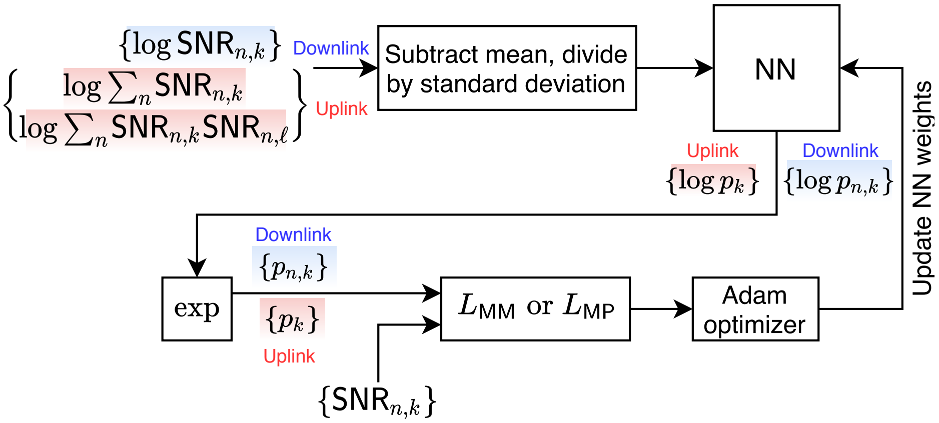
	Input layer	Hidden layer	Output layer
Neurons (Uplink)	200	50	$K = 12$
Neurons (Downlink)	1000	1000	$NK = 360$
Activation function	RLU	RLU	Linear
Regularization	L2 norm $\lambda = .001$	L2 norm $\lambda = .001$	L2 norm $\lambda = .001$

5.4.1 Learning Stage

The learning pipelines for both uplink and downlink are combined into Fig. 5.1 while the NN parameters are summarized in Table 5.1. The input parameters are first converted to log scale and subsequently rendered zero-mean and unit variance. The processing then properly starts with feature extraction by an input layer equipped with rectified linear unit (RLU) activation functions. Afterwards, a hidden layer processes the data also via RLUs, and an output layer with linear activation functions generates power coefficients in log scale; this guarantees positive outputs and averts numerical problems. From the SNRs and the corresponding NN outputs, the loss function of choice is quantified and an Adam optimizer is applied to minimize it. To prevent oscillations around local optima, the learning rate is reduced gradually from 0.001 down to 0.0001. And, to avoid overfitting, L2-norm regularization is employed in conjunction with the Adam optimizer: a portion $\lambda = 0.001$ of the L2 norm of the weights is added to the loss.

To streamline the learning, rather than a single large database, 50 (for the uplink) and 500 (for the downlink) databases of 12800 system realizations are generated and, over each, 200 updates of the NN weights take place; each update relies on a randomly selected batch of 128 realizations. The initialization is also random.

While the uplink learning curve is similar to its cellular counterpart [4] and not shown for the sake of brevity, the downlink learning curve is



$\{\log \mathsf{SNR}_{n,k}\}$
Downlink
$\left\{\begin{array}{c}\log\sum_n \mathsf{SNR}_{n,k}\\ \log\sum_n \mathsf{SNR}_{n,k}\mathsf{SNR}_{n,\ell}\end{array}\right\}$
Uplink
Subtract mean, divide by standard deviation
NN
Uplink
$\{\log p_k\}$
Downlink
$\{\log p_{n,k}\}$
Update NN weights
Downlink
$\{p_{n,k}\}$
exp
$\{p_k\}$
Uplink
L_{MM} or L_{MP}
Adam optimizer
$\{\mathsf{SNR}_{n,k}\}$

Figure 5.1: Learning pipeline.

different (see Fig. 5.2). Besides settling more slowly due to the larger number of power coefficients, it exhibits two stages that are not distinguishable in the uplink and that can be roughly mapped to the constraints and to the objective, the constituent parts of the loss function.

5.4.2 Performance Evaluation

For the C-RAN uplink, the power control problem can be cast in convex form under both of the considered loss functions. Therefore, performance benchmarks can be produced with an off-the-shelf convex solver.

For the C-RAN downlink, conversely, the feasible sets are convex but the loss functions L_{MM} and L_{MP} cannot be cast as convex in $\{p_{n,k}\}$. This is, in essence, because any increase in a user's transmit power simultaneously increases the interference to other users and detracts from their transmit powers. No convex-solver benchmarks are thus available in general. Only in the hard max-min limit (L_{MM} with $\alpha_k \to \infty$) does the downlink power optimization become quasi-convex in $\{p_{n,k}\}$ and can be solved through a tedious bisection search whose steps entail a sequence of convex feasibility optimizations [5].

We consider $N = 30$ APs and $K = 12$ users. The performance is evaluated via the cumulative distribution function (CDF) of local-average

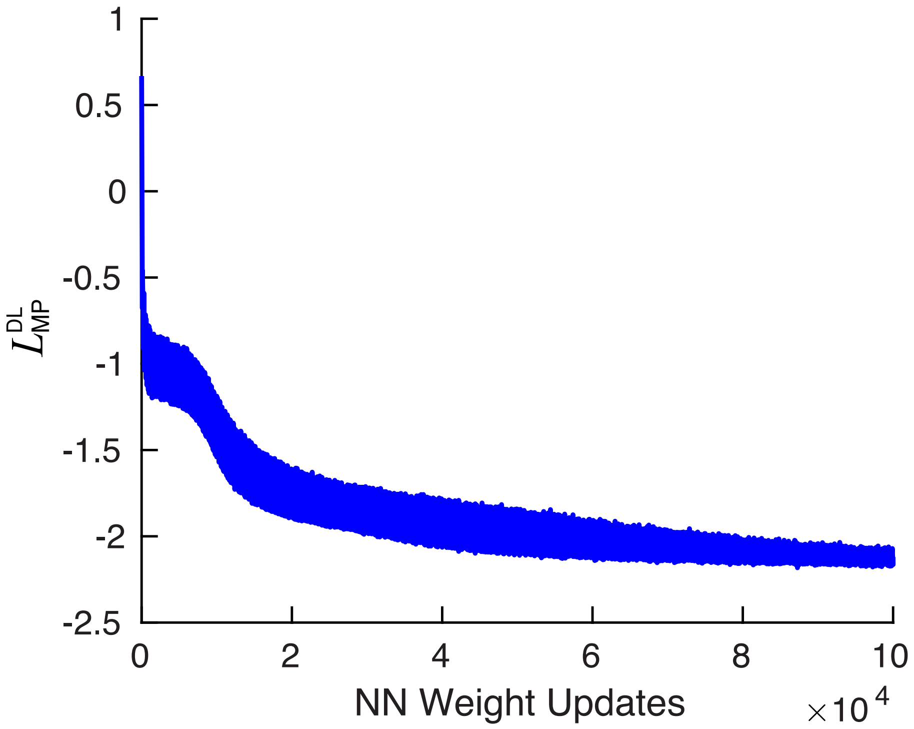


Figure 5.2: Downlink max-product learning curve with $\beta_k = 1$.

SINRs over the ensemble of user and AP positions. The NN and the convex-solver benchmarks (when available) are driven by the proxies in (5.6) and (5.10), while the performance with the obtained powers are asserted by means of the actual local-average SINRs in (5.5) and (6.9).

The uplink performance for the soft max-min loss function, shown in Fig. 5.3, prompts the following remarks:

- Power control is instrumental to avoid major disparities among users.

- For given $\{\alpha_n\}$ (unity in this case), the agreement with the convex solver is excellent. The NN slightly favors high-SNR users, falling shy of the convex benchmark in terms of the soft max-min behavior.

- Hardening the loss function is ill-advised, as that uniformly wors-

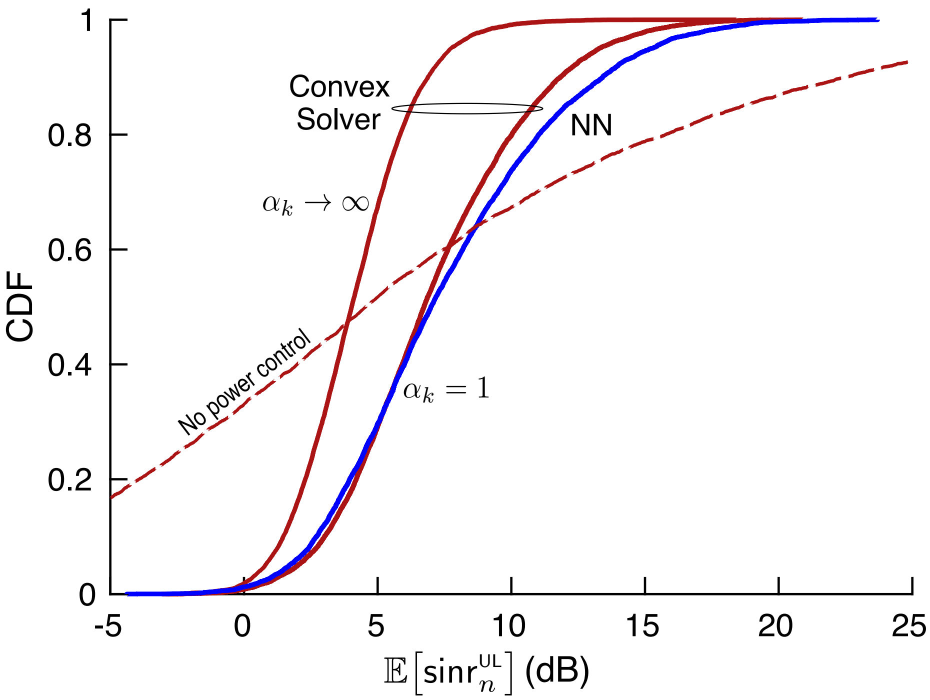


Figure 5.3: CDF of $\mathbb{E}\big[\mathsf{sinr}_k^{\mathsf{UL}}\big]$ for soft max-min: NN vs convex solver ($\alpha_k = 1$). Also shown is the baseline performance without power control, and the hard max-min performance ($\alpha_k \to \infty$).

ens the performance; in the limit, the entire system is conditioned by its worst situated user.

For the max-product loss function, Fig. 5.4 again reveals an excellent match between the NN and the convex solver.

In the downlink, where convex solvers do not in general provide optimality guarantees, we invoke as baseline

$$p_{n,k} = \frac{\mathsf{SNR}_{n,k}}{\sum_{k=0}^{K-1} \mathsf{SNR}_{n,k}} \qquad\qquad n = 0, \ldots, N-1 \qquad (5.15)$$

whereby every APs transmits its complete power and the share that the nth AP allocates to user k is proportional to the strength of the corresponding link. This baseline, rather standard as a complement to matched-filter

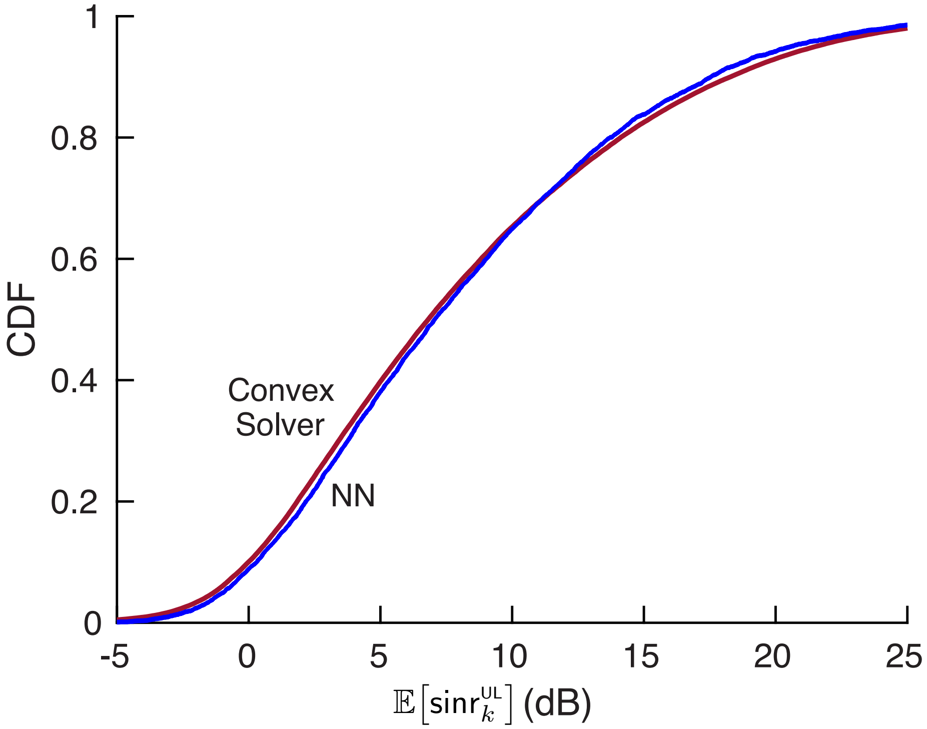


Figure 5.4: CDF of $\mathbb{E}\left[\mathsf{sinr}_k^{\mathsf{UL}}\right]$ for max-product ($\beta_k = 1$): NN vs convex solver.

beamforming, is exhibited in Fig. 5.5 alongside the NN-based performance under both soft max-min and max-product loss functions. In both cases, the NN outperforms markedly the baseline, with more emphasis on fairness—a steeper CDF—in the soft max-min case and less emphasis in the max-product case.

With very small probability, the objective part of the loss function could push the solution to violate one or several constraints. As shown by Fig. 5.6, the probability that the transmit power of an AP exceeds its maximum value is below 2% in the soft max-min case with $\alpha_k = 1$, and the impact of trimming down these power spillovers is minute (see Fig. 5.5).

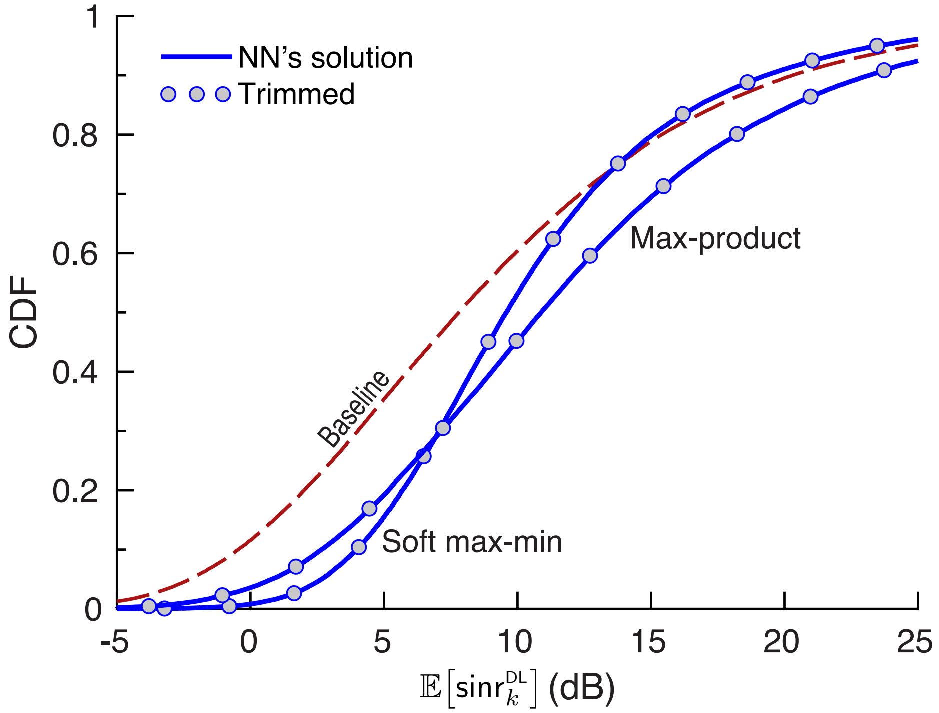


Figure 5.5: CDF of $\mathbb{E}\left[\mathsf{sinr}_k^{\mathsf{DL}}\right]$: NN (for soft max-min with $\alpha_k = 1$ and for max-product with $\beta_k = 1$) vs the baseline in (5.15). Also shown, in circles, is the modified performance when the powers are trimmed such that no constraint is violated.

5.4.3 Computational Cost

As in [3] and [4], we invoke, as a measure of the computational cost, the running time on a CPU-based platform.

Each NN inference is in essence a matrix multiplication, over two (in the uplink) and three (in the downlink) orders of magnitude faster than a convex solver for our 30-AP system.

As far as the learning, which needs to take place upon changes in the C-RAN or the environment, it is of interest to compare the learning time of our unsupervised NN with that of a supervised NN of the same dimensions [12–15]. While the training effort per parameter sample is essentially the same, a supervised NN requires producing those samples

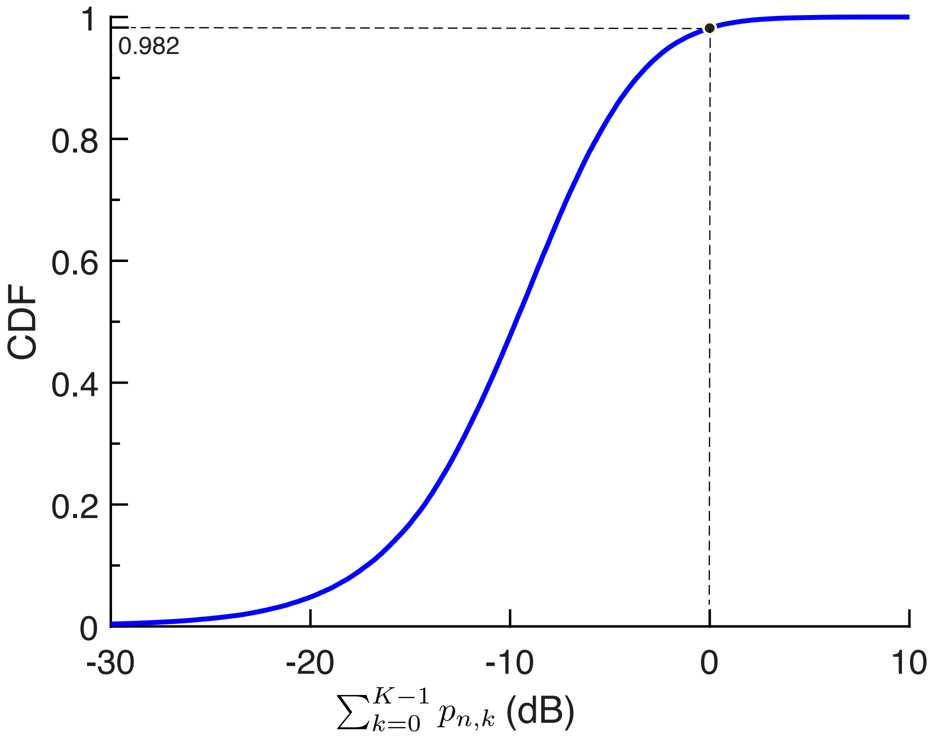


Figure 5.6: CDF of $\sum_{k=0}^{K-1} p_{n,k}$ as produced by the NN for the soft max-min loss function ($\alpha_n = 1$).

by solving the corresponding optimizations in the first place, again resulting in the two-to-three-order-of-magnitude advantage for the example at hand. (Allowing the NN to train on a more apt GPU-based platform, the superiority of unsupervised training increases by yet another order of magnitude.) And, for nonconvex objectives, it is challenging to generate those training samples in the first place.

5.5 Summary

The unsupervised learning approach to parametric optimization described in [3] has been shown to be very effective for the problems of power control and power allocation in cell-free C-RANs—even with the NN trained with a proxy to the actual objective, the local-average SINR.

For the uplink, where the considered objectives lead to convex formulations, the NN performance matches that of convex solvers while incurring orders of magnitude less computational cost. For the downlink, where those objectives cannot be tackled with certainty by convex solvers, the NN itself becomes a source of benchmark results. On that note, it would be of interest to tackle other nonconvex loss functions, for instance those related to energy efficiency with minimum performance guarantees at the users [16, 17].

The NN-based power optimizations continue to be effective if the assumption of known fading coefficients is overcome, and the fading coefficients are explicitly estimated on the basis of pilot transmission. Some results under that proviso are presented in [1, 2] for uplink and downlink, respectively.

The Python code developed to produce the results in this letter is openly available [18].

References

[1] R. Nikbakht, A. Jonsson, and A. Lozano, "Unsupervised-learning power control for cell-free wireless systems," in *IEEE Int'l Symp. Personal, Indoor and Mobile Radio Commun. (PIMRC'19)*, Sept. 2019.

[2] R. Nikbakht, A. Jonsson, and A. Lozano, "Unsupervised-learning power allocation for the cell-free downlink," in *IEEE Int'l Conf. on Communications (ICC'20)*, Jun. 2020.

[3] R. Nikbakht, A. Jonsson, and A. Lozano, "Unsupervised learning for parametric optimization," *IEEE Commun. Letters*, vol. 24, 2020.

[4] R. Nikbakht, A. Jonsson, and A. Lozano, "Unsupervised learning for cellular power control," *IEEE Commun. Letters*, vol. 24, 2020.

[5] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, pp. 1834–1850, Mar. 2017.

[6] S. Venkatesan, A. Lozano, and R. Valenzuela, "Network MIMO: Overcoming intercell interference in indoor wireless systems," *Asilomar Conf. Signals, Systems and Computers*, pp. 83–87, 2007.

[7] A. Checko et al., "Cloud RAN for mobile networks – a technology overview," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 405–426, 1st Quart. 2015.

[8] S. Perlman and A. Forenza, "An introduction to pCell," Tech. Rep., Artemis Networks LLC, White paper, Feb. 2015.

[9] A. Lozano and N. Jindal, "Are yesterday's information-theoretic fading models and performance metrics adequate for the analysis of today's wireless systems?," *IEEE Comm. Magazine*, vol. 50, no. 11, pp. 210–217, Nov. 2012.

[10] B. Błaszczyszyn, M. K. Karray, and H. P. Keeler, "Wireless networks appear Poissonian due to strong shadowing," *IEEE Trans. Wireless Communications*, vol. 14, pp. 4379–4390, Aug. 2015.

[11] R. Nikbakht, R. Mosayebi, and A. Lozano, "Uplink fractional power control and downlink power allocation for cell-free networks," *IEEE Wireless Commun. Letters*, vol. 9, no. 6, pp. 774–777, 2020.

[12] S. Chakraborty, E. Björnson, and L. Sanguinetti, "Centralized and distributed power allocation for max-min fairness in cell-free massive MIMO," in *Asilomar Conf. Signals, Systems, and Computers*, Nov. 2019.

[13] C. D'Andrea, A. Zappone, S. Buzzi, and M. Debbah, "Uplink power control in cell-free massive MIMO via deep learning," in *IEEE Int'l Workshop Computational Adv. Multi-Sensor Adapt. Processing*, 2019.

[14] M. Bashar, A. Akbari, K. Cumanan, H. Quoc Ngo, A. Burr, P. Xiao, M. Debbah, and J. Kittler, "Exploiting deep learning in limited-fronthaul cell-free massive MIMO uplink," *IEEE J. Sel. Areas Commun.*, 2020.

[15] Y. Zhao, I. Niemegeers, and S. De Groot, "Power allocation in cell-free massive MIMO: A deep learning method," *IEEE Access*, vol. 8, pp. 87185–87200, 2020.

[16] C. Yoon and D.-H. Cho, "Energy efficient beamforming and power allocation in dynamic TDD based C-RAN system," *IEEE Commun. Letters*, vol. 19, no. 10, pp. 1806–1809, 2015.

[17] P. Cao, W. Liu, J. Thompson, C. Yang, and E. Jorswieck, "Semidynamic green resource management in downlink heterogeneous networks by group sparse power control," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1250–1266, 2016.

[18] R. Nikbakht, "Unsupervised NN," https://github.com/RasoulNik/UnsupervisedNN, 2020.

Chapter 6

Uplink Fractional Power Control and Downlink Power Allocation for Cell-Free Networks

Rasoul Nikbakht, Student Member, IEEE, Reza Mosayebi, Angel Lozano, Fellow, IEEE

***Abstract*— This paper proposes respective policies for uplink power control and downlink power allocation in cell-free wireless networks. Both policies rely only on large-scale quantities and are expressed in closed form, being therefore scalable. The uplink policy, which generalizes the fractional power control employed extensively in cellular networks, features a single parameter; by adjusting this parameter, the SIR distribution experienced by the users can be compressed or expanded, trading average performance for fairness. The downlink policy dualizes the uplink solution, featuring two parameters that again allow effecting a tradeoff between average performance and fairness.**

***Index Terms*— Cell-free networks, ultradense networks, power control, power allocation, beamforming**

The authors are with Univ. Pompeu Fabra, 08018 Barcelona (e-mail: angel.lozano@upf.edu). Their work is supported by the European Research Council under the H2020 Framework Programme/ERC grant agreement 694974, and by MINECO's Projects RTI2018-102112 and RTI2018-101040. Parts of this paper were presented at the 2019 IEEE Int'l Conf. Commun. [10].

6.1 Introduction

Cell-free networks consist of a dense infrastructure of access points (APs), each potentially communicating with every user (see [1–6] and references therein). This paradigm inherits ideas from network MIMO [7] and cloud radio access [8], with the possibility of having substantially more antennas than users per time-frequency resource unit so as to render matched-filter beamforming effective.

One of the challenges posed by cell-free networks is to devise policies for uplink power control and downlink power allocation. Given the major differences in pathloss and shadowing among links, such policies are essential to avoid huge performance disparities. This is illustrated in [1], where max-min policies are seen to equalize the performance across users relative to fixed-power situations. However, max-min policies entail centralized iterative procedures whose steps, in turn, involve a sequence of convex optimizations; the implementation of these procedures does not scale to large networks [9]. Moreover, max-min solutions suffer from excessive rigidness in that they are dragged down by worst-case users [10].

Alternative schemes based on heuristics are propounded in [4, 9] and, while these do scale, their performance is decidedly inferior. In this context, we seek new policies that comply with the following desiderata.

- Retaining the virtue, exhibited by the max-min solutions in [1], of depending only on large-scale quantities.

- Being directly computable from such large-scale quantities so as to ensure scalability.

- Allowing for a tradeoff between equalizing the performance across users (i.e., ensuring fairness) and maximizing the average performance.

For cellular networks, an uplink power control policy that satisfies the foregoing desiderata was devised in the form of fractional power control, variants of which have been adopted by standards such as LTE and NR. Inspired by this highly successful approach, we seek a fractional power control formulation for the cell-free uplink and, subsequently, a dual power allocation solution for the corresponding downlink.

6.2 Network and channel models

The networks under consideration feature N single-antenna APs and K single-antenna users. Every AP can communicate with every user on each time-frequency resource unit.

Under the premise that the AP locations are agnostic to the radio propagation, shadowing makes those locations appear Poisson-distributed from the vantage of any user [11]. Although asymptotic in the shadowing strength, this phenomenon is well manifested for shadowing intensities of interest [11, 12]. Capitalizing on this result, we draw the AP positions uniformly, avoiding the need for explicit modeling of the shadowing as it is then already implicitly captured by the geometry. Likewise, the user positions are drawn uniformly.

The signals are subject to pathloss with exponent η, which yields a large-scale channel gain $G_{n,k}$ between the kth user and the nth AP. Besides $G_{n,k}$, the channel between the kth user and the nth AP features a small-scale fading coefficient $h_{n,k} \sim \mathcal{N}_{\mathbb{C}}(0,1)$, independent across users and APs.

Interference-limited conditions are considered, with noise negligible relative to the interference. Pilot contamination is also disregarded, as it can be kept at bay with procedures such as the ones described in [1, Sec. IV] or in [13,14]. It follows that, based on uplink pilot transmissions from the users, the nth AP can perfectly estimate $h_{n,0}, \ldots, h_{n,K-1}$.

6.3 Uplink

Upon payload data transmission from the users, the nth AP observes

$$y_n = \sum_{k=0}^{K-1} \sqrt{G_{n,k}} h_{n,k} \sqrt{p_k} s_k, \tag{6.1}$$

where s_k is the unit-power symbol transmitted by user k while $p_k \in [0,1]$ is its power control coefficient. With matched filtering, the signal of user k is recovered from

$$\sum_{n=0}^{N-1} \sqrt{G_{n,k}} h_{n,k}^* \, y_n = \sum_{n=0}^{N-1} G_{n,k} \sqrt{p_k} \, |h_{n,k}|^2 \, s_k \\ + \sum_{n=0}^{N-1} \sqrt{G_{n,k}} \, h_{n,k}^* \sum_{\ell \neq k} \sqrt{G_{n,\ell}} \sqrt{p_\ell} \, h_{n,\ell} \, s_\ell \tag{6.2}$$

such that the signal-to-interference ratio (SIR) of user k, expected over the known $\{h_{n,k}\}$, equals [1, 10]

$$\mathbb{E}[\mathsf{sir}_k] = \mathbb{E}\left[\frac{p_k \left(\sum_{n=0}^{N-1} G_{n,k} \, |h_{n,k}|^2 \right)^2}{\sum_{\ell \neq k} p_\ell \left| \sum_{n=0}^{N-1} \sqrt{G_{n,k} G_{n,\ell}} \, h_{n,k}^* h_{n,\ell} \right|^2} \right]. \tag{6.3}$$

6.3.1 Fractional Power Control

The origins of fractional power control can be traced back to [15]. The objective of that initial derivation was to minimize the variance of the large-scale SIR distribution (in dB) experienced by two interfering cellular users, and the solution turned out to be $p_k \propto 1/\sqrt{G_k}$ where G_k was the large-scale gain to the serving cell and with the proportionality such that $p_k \in [0,1]$. This was subsequently generalized to $p_k \propto 1/G_k^{\vartheta}$ where $\vartheta \in [0,1]$ regulates the extent to which the range of received powers are compressed [16]. The values typically featured in LTE and NR are in the range $\vartheta \approx 0.5$–0.7, with lower values favoring the average SIR while higher values promote fairness and cell-edge performance.

Derived in Appendix 6.A, our proposed generalization of the above to cell-free networks is

$$p_k \propto \frac{1}{\left(\sum_{n=0}^{N-1} G_{n,k} \right)^{\vartheta}}, \tag{6.4}$$

which now depends on all the large-scale channel gains that involve a given user, reflecting the effective connection between such user and the network.

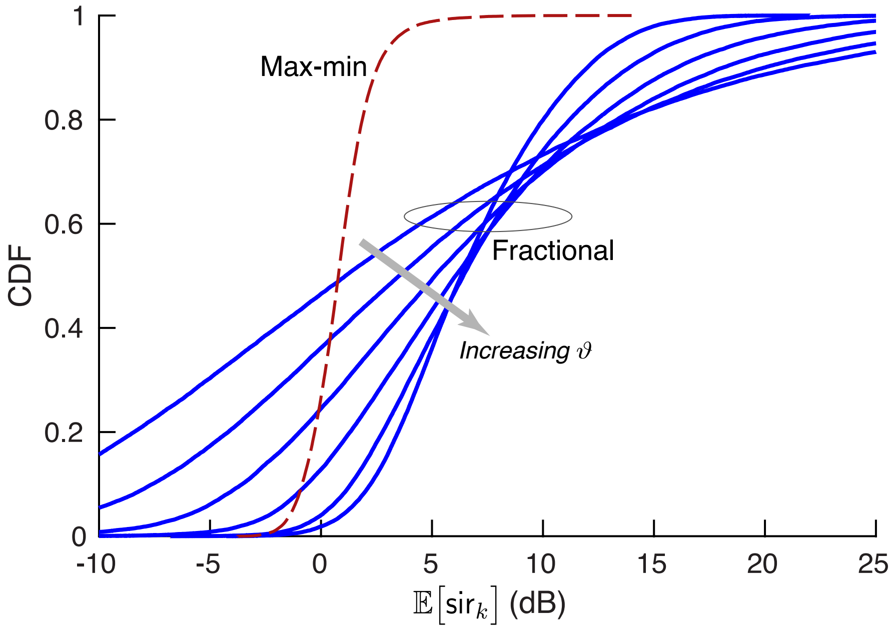


Figure 6.1: CDF of $\mathbb{E}[\mathsf{sir}_k]$ for the uplink with $\eta = 3.8$ and $N/K = 2.5$, parameterized by $\vartheta = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. Also shown is the max-min solution.

6.3.2 Performance Evaluation

To exemplify the performance, we consider a wrapped-around universe with $N = 200$ APs, $K = 80$ users, and a pathloss exponent of $\eta = 3.8$. The number of network snapshots is such that, with 95% confidence, the CDFs do not deviate from their true value by more than 0.3%.

Shown in Fig. 6.1 is the CDF of $\mathbb{E}[\mathsf{sir}_k]$, parameterized by ϑ. Sweeping this parameter from $\vartheta = 0$ (fixed transmit powers) to $\vartheta = 1$ (complete large-scale channel inversion), we observe a progressive compression of the CDF. The tradeoff that ϑ enables between the average and the 3%-

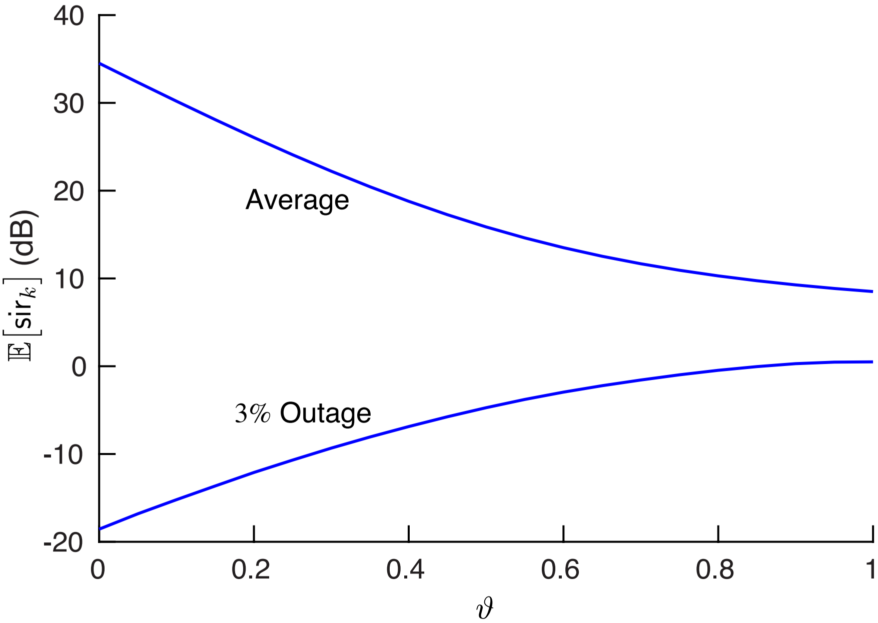


Figure 6.2: Average and 3%-outage values of $\mathbb{E}[\mathsf{sir}_k]$ for the uplink as a function of ϑ, with $\eta = 3.8$ and $N/K = 2.5$.

outage (as a proxy for fairness) can be appreciated in Fig. 6.2. Moving from $\vartheta = 0$ to $\vartheta = 1$, the 3%-outage point increases steadily at the expense of the average. Fig. 6.1 also contrasts (6.4) with the max-min solution: the $\{p_k\}$ that maximize the lowest SIR as per (6.14) are obtained through geometric programming and plugged into (6.3). Fractional power control is more elastic than the max-min solution, which seems to be dragged down by worst-case users, paying a price for its uncompromising equalization of all SIRs.

6.4 Downlink

With conjugate beamforming, the precoder applied by the nth AP to transmit to user k is $h_{n,k}$, based on which the nth AP generates the transmit signal

$$x_n = \sum_{k=0}^{K-1} \sqrt{p_{n,k}}\, h_{n,k} s_k \tag{6.5}$$

where s_k is the unit-power symbol meant for user k while $p_{n,k} \in [0,1]$ is the share of power that the nth AP devotes to such user, with $\sum_{k=0}^{K-1} p_{n,k} \leq 1$. User k then observes

$$\begin{aligned} y_k &= \sum_{n=0}^{N-1} \sqrt{G_{n,k}}\, h_{n,k}^* \, x_n && (6.6) \\ &= \underbrace{\sum_{n=0}^{N-1} \sqrt{G_{n,k}\, p_{n,k}}\, |h_{n,k}|^2 \, s_k}_{\text{Signal: } S_k} \\ &\quad + \underbrace{\sum_{n=0}^{N-1} \sqrt{G_{n,k}}\; h_{n,k}^* \sum_{\ell \neq k} \sqrt{p_{n,\ell}}\, h_{n,\ell} s_\ell \,.}_{\text{Interference: } I_k} && (6.7) \end{aligned}$$

The performance achievable on the basis of y_k hinges on the knowledge at user k of the effective channel

$$c_k = \sum_{n=0}^{N-1} \sqrt{G_{n,k} p_{n,k}}\, |h_{n,k}|^2 \tag{6.8}$$

that relates s_k with y_k. In cellular massive MIMO, it is effectual to rely solely on the mean of c_k since, because of hardening, the actual value never departs significantly from such mean [17, Ch. 10]. In single-antenna cell-free networks, however, the hardening is only partial [3, 5]. The effective channels fluctuate markedly, which gives rise to self-interference if the kth receiver is only privy to $\mathbb{E}[c_k]$.

Self-interference can be avoided by inserting precoded pilots within the downlink transmissions, enabling the explicit estimation by the users of their effective channels [2]. Recalling S_k and I_k as defined in (6.7), the

kth user can then operate at

$$\mathsf{sir}_k = \frac{\mathbb{E}\big[|S_k|^2 \, | c_k\big]}{\mathbb{E}\big[|I_k|^2 \, | c_k\big]} \tag{6.9}$$

$$= \frac{\left(\sum_{n=0}^{N-1} \sqrt{G_{n,k} \, p_{n,k}} \, |h_{n,k}|^2\right)^2}{\sum_{n=0}^{N-1} G_{n,k} \, \mathbb{E}\big[|h_{n,k}|^2 \, | c_k\big] \sum_{\ell \neq k} p_{n,\ell}}. \tag{6.10}$$

Since c_k is a function of $h_{0,k}, \ldots, h_{N-1,k}$, the realization of c_k conditions an individual $h_{n,k}$ only very weakly, hence $\mathbb{E}\big[|h_{n,k}|^2 \, | c_k\big] \approx \mathbb{E}\big[|h_{n,k}|^2\big] = 1$. It follows that the expectation of sir_k over the small-scale fading, and consequently over c_k, satisfies

$$\mathbb{E}[\mathsf{sir}_k] \approx \frac{\mathbb{E}\left[\left(\sum_{n=0}^{N-1} \sqrt{G_{n,k} p_{n,k}} \, |h_{n,k}|^2\right)^2\right]}{\sum_{n=0}^{N-1} G_{n,k} \sum_{\ell \neq k} p_{n,\ell}}. \tag{6.11}$$

6.4.1 Fractional Power Allocation

Capitalizing on the fractional power control derived for the uplink, we seek a downlink counterpart that can play a similar role. The policy we propose, derived in Appendix 6.B, is

$$p_{n,k} \propto \frac{G_{n,k}}{\left(\sum_{m=0}^{N-1} G_{m,k}\right)^{\vartheta} \left(\sum_{\ell=0}^{K-1} \frac{G_{n,\ell}}{\left(\sum_{m=0}^{N-1} G_{m,\ell}\right)^{\vartheta}}\right)^{\gamma}}, \tag{6.12}$$

where, as in the uplink, $\vartheta \in [0, 1]$ while γ is an additional parameter best set in the range $[0.4, 1.6]$.

6.4.2 Performance Evaluation

Besides the max-min solution, in the downlink we consider as an additional benchmark the heuristic policy suggested in [4, Sec. III-D]. Fig. 6.3 shows the CDF of $\mathbb{E}[\mathsf{sir}_k]$, parameterized by ϑ and γ. Equipped with these parameters, (6.12) is seen to be highly versatile: not only can it match the performance of [4, Sec. III-D], but its upper tail can surpass it (when configured for average performance) or its lower tail can amply outperform it (when configured for fairness). In this case, though, the lower tail falls short of the max-min solution.

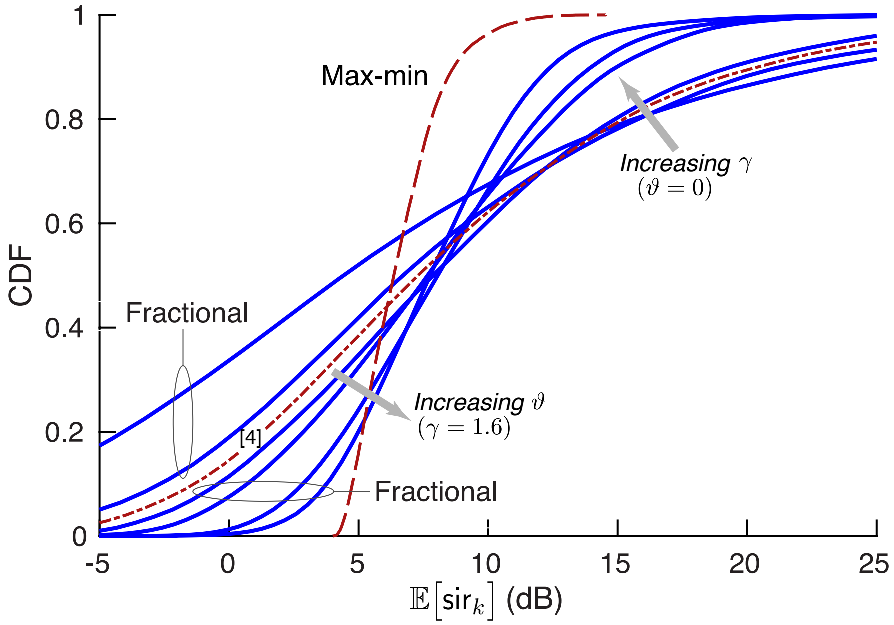


Figure 6.3: CDF of $\mathbb{E}[\mathsf{sir}_k]$ for the downlink with $\eta = 3.8$ and $N/K = 2.5$, parameterized by $\vartheta = \{0, 0.4, 0.8\}$ and $\gamma \in \{0.4, 0.8, 1.2, 1.6\}$. Also shown are the max-min solution and the benchmark from [4, Sec. III-D].

6.5 Summary

The proposed fractional power control and power allocation policies, embodied by (6.4) and (6.12), are closed-form functions of the large-scale channel gains and allow regulating the tradeoff between average performance and fairness. The regulation is exerted through one (uplink) or two (downlink) parameters, hence it does not enable favoring individual users or maximizing specific metrics (e.g., a weighted sum). If such is the goal, learning-based solutions offer an attractive compromise between performance and scalability [18].

Appendices

6.A

Inspired by [15], we seek to minimize the variance of the large-scale SIR distribution (in dB), hence the starting point needs to be an expression for such SIR. In a cellular network, this equals the large-scale signal power divided by the large-scale other-cell interference power, namely [17]

$$\mathsf{SIR}_k = \frac{p_k \, G_k}{\sum_{\ell \neq k} p_\ell \, G_\ell}, \tag{6.13}$$

which can be recovered from (6.3) by letting $N = 1$ and replacing $|h_k|^2$ and $|h_\ell|^2$ by their unit expected value. A counterpart to (6.13) for cell-free networks can be obtained by similarly replacing $|h_{n,k}|^2$ and $|h_{n,\ell}|^2$ by unity, and by zeroing out cross-terms containing $h_{n,k}^* h_{m,\ell}$ for $n \neq m$. This gives

$$\mathsf{SIR}_k = \frac{p_k \left(\sum_{n=0}^{N-1} G_{n,k} \right)^2}{\sum_{\ell \neq k} p_\ell \sum_{n=0}^{N-1} G_{n,k} G_{n,\ell}}. \tag{6.14}$$

Specializing (6.14) to $K = 2$ and focusing, without loss of generality, on user 0,

$$\mathsf{SIR}_0 = \frac{p_0 \left(\sum_{n=0}^{N-1} G_{n,0} \right)^2}{p_1 \sum_{n=0}^{N-1} G_{n,0} G_{n,1}}, \tag{6.15}$$

from which, introducing the notation $z|_{\rm dB} = 10 \log_{10} z$, we can further write

$$\mathsf{SIR}_0|_{\rm dB} = p_0|_{\rm dB} + 2 \, G_{\rm S}|_{\rm dB} - p_1|_{\rm dB} - G_{\rm I}|_{\rm dB}, \tag{6.16}$$

where $G_{\mathrm{S}} = \sum_{n=0}^{N-1} G_{n,0}$ and $G_{\mathrm{I}} = \sum_{n=0}^{N-1} G_{n,0} G_{n,1}$.

We are interested in distributed power control policies where p_0 depends on $\{G_{n,0}\}_{n=0}^{N-1}$, but not on $\{G_{n,1}\}_{n=0}^{N-1}$, hence the following can be reasonably assumed to hold:

- p_0 and p_1 are independent, as they are controlled on the basis of distinct channel gains.
- p_0 is not independent of G_{S}, nor of G_{I}.
- p_1 is not independent of G_{I}, but it is independent of G_{S}.

With that, the variance of (6.16) can be developed into

$$\begin{aligned}
\mathrm{var}[\mathsf{SIR}_0|_{\mathrm{dB}}] =& \,\mathrm{var}\left[p_0|_{\mathrm{dB}} + 2\, G_{\mathrm{S}}|_{\mathrm{dB}}\right] + \mathrm{var}\left[p_1|_{\mathrm{dB}} + G_{\mathrm{I}}|_{\mathrm{dB}}\right] \\
& - 2\,\mathrm{cov}[p_0|_{\mathrm{dB}} + 2\, G_{\mathrm{S}}|_{\mathrm{dB}}, p_1|_{\mathrm{dB}} + G_{\mathrm{I}}|_{\mathrm{dB}}] \\
=& \,\mathrm{var}[p_0|_{\mathrm{dB}}] + 4\,\mathrm{var}[G_{\mathrm{S}}|_{\mathrm{dB}}] + 4\,\mathrm{cov}[p_0|_{\mathrm{dB}}, G_{\mathrm{S}}|_{\mathrm{dB}}] \\
& + \mathrm{var}[p_1|_{\mathrm{dB}}] + \mathrm{var}[G_{\mathrm{I}}|_{\mathrm{dB}}] + 2\,\mathrm{cov}[p_1|_{\mathrm{dB}}, G_{\mathrm{I}}|_{\mathrm{dB}}] \\
& - 2\,\mathrm{cov}[p_0|_{\mathrm{dB}}, G_{\mathrm{I}}|_{\mathrm{dB}}] - 4\,\mathrm{cov}[G_{\mathrm{S}}|_{\mathrm{dB}}, G_{\mathrm{I}}|_{\mathrm{dB}}],
\end{aligned}$$

which, by virtue of the fact that $\mathrm{var}[p_0|_{\mathrm{dB}}] = \mathrm{var}[p_1|_{\mathrm{dB}}]$ and $\mathrm{cov}\left[p_0|_{\mathrm{dB}}, G_{\mathrm{I}}\right] = \mathrm{cov}\left[p_1|_{\mathrm{dB}}, G_{\mathrm{I}}\right]$, simplifies into

$$\begin{aligned}
\mathrm{var}[\mathsf{SIR}_0|_{\mathrm{dB}}] &= 2\,\mathrm{var}[p_0|_{\mathrm{dB}}] + 4\,\mathrm{cov}[p_0|_{\mathrm{dB}}, G_{\mathrm{S}}|_{\mathrm{dB}}] \\
&+ 4\,\mathrm{var}[G_{\mathrm{S}}|_{\mathrm{dB}}] + \mathrm{var}[G_{\mathrm{I}}|_{\mathrm{dB}}] - 4\,\mathrm{cov}[G_{\mathrm{S}}|_{\mathrm{dB}}, G_{\mathrm{I}}|_{\mathrm{dB}}].
\end{aligned} \tag{6.17}$$

Regrouping some terms, the above can be rewritten as

$$\begin{aligned}
\mathrm{var}[\mathsf{SIR}_0|_{\mathrm{dB}}] &= 2\,\mathrm{var}[p_0|_{\mathrm{dB}} + G_{\mathrm{S}}|_{\mathrm{dB}}] + 2\,\mathrm{var}[G_{\mathrm{S}}|_{\mathrm{dB}}] \\
&\quad + \mathrm{var}[G_{\mathrm{I}}|_{\mathrm{dB}}] - 4\,\mathrm{cov}[G_{\mathrm{S}}|_{\mathrm{dB}}, G_{\mathrm{I}}|_{\mathrm{dB}}],
\end{aligned} \tag{6.18}$$

where the last three terms do not depend on $p_0|_{\mathrm{dB}}$, hence they are immaterial to the minimization with respect to it. The quantity to minimize is $\mathrm{var}[p_0|_{\mathrm{dB}} + G_{\mathrm{S}}|_{\mathrm{dB}}]$, and the power control policy that minimizes it is $p_0|_{\mathrm{dB}} = -G_{\mathrm{S}}|_{\mathrm{dB}}$, i.e.,

$$p_0 = \frac{1}{\sum_{n=0}^{N-1} G_{n,0}}. \tag{6.19}$$

With a view to $K > 2$ users and to regulating the forcefulness of the policy, we generalize (6.19) into (6.4) by incorporating ϑ. Finally, all powers should be scaled to fit within $[0, 1]$.

6.B

Observe that, in the uplink, (6.4) depends only on the numerator of the SIR expression from which it derives. This is consistent with the nature of beamforming, whose sole purpose is to maximize the signal power from the intended user, with no regard for the interference. Moreover, with the exponent set to its highest value of $\vartheta = 1$, the power control in (6.4) sets the received signal power from user k to $\sum_{n=0}^{N-1} G_{n,k}$.

From the numerator of the downlink SIR expression in (6.10), with $|h_{n,k}|^2$ replaced once more by its unit expected value, the same behavior can be induced if $p_{0,k}, \ldots, p_{N-1,k}$ ensure that

$$\sum_{n=0}^{N-1} \sqrt{G_{n,k}\, p_{n,k}} = \sqrt{\sum_{n=0}^{N-1} G_{n,k}}, \tag{6.20}$$

which has infinitely many solutions. As the left-hand side of (6.20) is concave on $p_{0,k}, \ldots, p_{N-1,k}$, we can identify the one solution minimizing $\sum_{n=0}^{N-1} p_{n,k}$ by building the Lagrangian

$$L = \sum_{n=0}^{N-1} p_{n,k} + \lambda \left(\sum_{n=0}^{N-1} \sqrt{G_{n,k} p_{n,k}} - \sqrt{\sum_{n=0}^{N-1} G_{n,k}} \right). \tag{6.21}$$

Setting $\partial L / \partial p_{n,k} = 0$, what emerges is $p_{n,k} = \alpha\, G_{n,k}$ with α upholding (6.20). This gives $p_{n,k} = G_{n,k} / \sum_{m=0}^{N-1} G_{m,k}$, which, proceeding as in the uplink, we broaden to

$$p_{n,k} = \frac{G_{n,k}}{\left(\sum_{m=0}^{N-1} G_{m,k} \right)^{\vartheta}}. \tag{6.22}$$

The expression in (6.22), which can be regarded as the downlink dual of (6.4), can be further generalized to account for the additional issue that appears in the downlink, namely that the total power per AP is constrained. In order to equalize the per-AP total powers, we can multiply $p_{n,k}$ by α_n with

$$\alpha_n = \frac{1}{\sum_{\ell=0}^{K-1} \frac{G_{n,\ell}}{\left(\sum_{m=0}^{N-1} G_{m,\ell}\right)^{\vartheta}}} \tag{6.23}$$

ensuring equal total power per AP. Finally, introducing a new parameter γ, we can relax the per-AP power equalization into a partial equalization as per (6.12).

Acknowledgment

Discussions with Prof. Francisco Monteiro, from Instituto de Telecomunicaccões (Portugal), are gratefully acknowledged.

References

[1] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. on Wireless Communications*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.

[2] G. Interdonato, H. Q. Ngo, E. G. Larsson, and P. Frenger, "How much do downlink pilots improve cell-free massive MIMO?" *IEEE Global Commun. Conf. (GLOBECOM'16)*, pp. 1–7, 2016.

[3] Z. Chen and E. Björnson, "Channel hardening and favorable propagation in cell-free massive MIMO with stochastic geometry," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5205–5219, 2018.

[4] E. Nayebi, A. Ashikhmin, T. L. Marzetta, H. Yang, and B. D. Rao, "Precoding and power optimization in cell-free massive MIMO sys-

tems," *IEEE Trans. Wireless Commun.*, vol. 16, pp. 4445–4459, Jul. 2017.

[5] M. Attarifar, A. Abbasfar, and A. Lozano, "Modified conjugate beamforming for cell-free massive MIMO," *IEEE Wireless Commun. Letters*, vol. 8, no. 2, pp. 616–619, Apr. 2019.

[6] J. Zhang, S. Chen, Y. Lin, J. Zheng, B. Ai, and L. Hanzo, "Cell-free massive MIMO: A new next-generation paradigm," *IEEE Access*, vol. 7, pp. 99 878–99 888, 2019.

[7] S. Venkatesan, A. Lozano, and R. Valenzuela, "Network MIMO: Overcoming intercell interference in indoor wireless systems," *Asilomar Conf. Signals, Systems and Computers*, pp. 83–87, 2007.

[8] A. Checko *et al.*, "Cloud RAN for mobile networks – a technology overview," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 405–426, 1st Quart. 2015.

[9] G. Interdonato, P. Frenger, and E. Larsson, "Scalability aspects of cell-free massive MIMO," *IEEE Int'l Conf. Commun. (ICC'19)*, May 2019.

[10] R. Nikbakht and A. Lozano, "Uplink fractional power control for cell-free wireless networks," *IEEE Int'l Conf. Commun. (ICC'19)*, May 2019.

[11] B. Błaszczyszyn, M. K. Karray, and H. P. Keeler, "Wireless networks appear Poissonian due to strong shadowing," *IEEE Trans. Wireless Communications*, vol. 14, no. 8, pp. 4379–4390, Aug. 2015.

[12] G. George, R. K. Mungara, A. Lozano, and M. Haenggi, "Ergodic spectral efficiency in MIMO cellular networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 2835–2849, May 2017.

[13] O. Y. Bursalioglu, C. Wang, H. Papadopoulos, and G. Caire, “RRH based massive MIMO with ‘on the fly’ pilot contamination control,” in *IEEE Int’l Conf. on Communications (ICC’16)*, 2016, pp. 1–7.

[14] M. Attarifar, A. Abbasfar, and A. Lozano, “Random vs structured pilot assignment in cell-free massive MIMO wireless networks,” *IEEE Int’l Conf. Commun. Workshops (ICCW’18)*, May 2018.

[15] J. F. Whitehead, “Signal-level-based dynamic power control for co-channel interference management,” in *IEEE 43rd Vehicular Technology Conference*, May 1993, pp. 499–502.

[16] R. D. Yates, “A framework for uplink power control in cellular radio systems,” *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341–1347, 1995.

[17] R. W. Heath Jr and A. Lozano, *Foundations of MIMO Communication*. Cambridge University Press, 2019.

[18] R. Nikbakht and A. Lozano, “Unsupervised-learning power control for cell-free wireless systems,” in *IEEE Int’l Symp. Personal, Indoor and Mobile Commun. (PIMRC’19)*, Sep. 2019.

Chapter 7

Dual-Kernel Online Reconstruction of Power Maps

Rasoul Nikbakht, Student Member, IEEE, Anders Jonsson, Angel Lozano, Fellow, IEEE

Abstract— **We present a measurement-driven algorithm to map the large-scale channel losses observed between a cellular base station and any point in its coverage area. The algorithm is on-line, meaning that it operates on continuously arriving measurements. Its distinguishing features are the use of two kernel functions, suitably chosen for the problem at hand, and a simple technique to sparsify the dictionary of measurements retained in memory. Evaluations in campus and urban settings indicate that the proposed algorithm reduces, roughly in half, the prediction error of existing single-kernel and multikernel algorithms.**

Index Terms— **Received signal strength indicator, Regression analysis, Adaptive learning, Machine learning, Hilbert space, Supervised learning, Location awareness**

The authors are with Univ. Pompeu Fabra, 08018 Barcelona (e-mail: angel.lozano@upf.edu). Their work is supported by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502), TEC2015-66228-P, and the European Research Council (grant agreement 694974).

7.1 Introduction

7.1.1 Motivation

In stochastic channel modeling, the large-scale power loss is formulated as a combination of distance-dependent loss (or pathloss) and shadowing. The former accounts for the average value at each distance and the latter incorporates the uncertainty around this value. This approach has had a great deal of importance in the analysis and design of wireless networks, with popular pathloss models such as the Okumura-Hata [1], the COST-231 Hata [2] or the Stanford University interim model [3]. Indeed, the stochastic approach is rather effective at producing descriptions that are representative of classes of environments of interest for the purpose of design and even performance assessment. At the same time, stochastic modeling may perform quite poorly when it comes to reproducing site-specific behaviors [4].

Site-specific modeling has historically been tackled by means of ray-tracing techniques, which reproduce the propagation mechanisms in a computer recreation of the site [5]. Since this recreation must be highly accurate, not only in terms of the geometry but also of the constituent materials and their electromagnetic properties, ray tracing is mostly restricted to indoor sites. For outdoor environments, measurement-driven reconstructions are an enticing alternative, particularly given the massive number of permanently connected devices from which a network can obtain signal measurements nowadays [6–10]. We refer to a 2D reconstruction of the large-scale loss from the vantage of a base station as a *map*. (The local-average received power and the local-average SNR are scaled versions of the large-scale loss, hence their maps in dB are equivalent up to a constant.) Advantageously, measurement-driven maps are not only site-specific, but they automatically track environmental changes, e.g., due to the motion of people and vehicles. With that, measurement-driven maps are not only useful for network planning, but further for tasks such as resource allocation or interference management.

The algorithms to construct maps can be classified into two broad categories.

- *Batch.* These schemes need the measurements to be available before the processing starts, so they have a considerable computational cost and they cannot adapt until a new batch is processed [8, 11–13].

- *Online.* These schemes process measurements as they arrive, continuously adapting [9].

7.1.2 State of the Art

The measurement-driven approaches proposed for map reconstruction include Gaussian process regression [8] and online adaptive learning [9]. The former is a batch method based on a stochastic channel model combining pathloss and log-normal shadowing, whereby the large-scale loss in dB follows a multivariate Gaussian distribution with covariance determined by the shadowing's spatial correlation [14]. The latter, in turn, is an online method that does not make assumptions on the underlying channel and requires only a portion of the measurements, considerably reducing the complexity. Both methods do need accurate knowledge of the user locations.

7.1.3 Contribution

In a wireless network, the large-scale loss is confined to a dynamic range.

- It is bounded from below depending on the placement, elevation, and other aspects of the base station, as well as the frequency. For instance, the free-space pathloss at 2 GHz already equals 38.5 dB at 1 m, increasing to 60 dB at 12 m.

- It is also bounded from above because, once the signal falls well below the noise floor, the coverage is lost. While again this depends on the bandwidth, as well as the transmit power and other parameters, the in-coverage loss is typically below 160 dB [15].

Altogether, and because of sheer geometry, most users exhibit large-scale losses that fall on an interval of tens of dB. The center of this interval, which we refer to as the *DC component*, varies little across base stations and is fairly stable over time. If we subtract it out, what remains is a markedly more confined quantity, referred to as the *varying component*.

On the basis of the foregoing observation, we posit that a better estimator can be obtained if two distinct *kernel* functions—a measure of similarity on which we dwell later—are considered for the DC and varying components, respectively, and this is the idea that we explore in this paper. Complementing this dual-kernel approach with a relatively simple sparsification technique, whereby only relevant measurements are retained, we come up with an online regression tool that improves upon existing single-kernel and multikernel approaches in terms of convergence speed and accuracy.

7.2 Problem Formulation

Let us parcel the region containing the network into small pixels according to a predefined resolution, such that locations are discrete rather than continuous. Because of the noise floor, each base station covers a certain area and a threshold is defined to specify it. A pixel is deemed to be within the coverage area of a base station if the corresponding large-scale loss is below the threshold.

Given a base station, consider the relationship

$$y_n = f(\mathbf{x}_n) + \varepsilon_n \tag{7.1}$$

where $y_n \in \mathbb{R}$ is the average power measured from a transmitter located at $\mathbf{x}_n \in \mathbb{R}^2$, $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ is an unknown function, n indexes the measurement, and ε_n represents the measurement noise. Our objective is to learn an estimate $\hat{f}$ of the unknown function f from a stream of measurements $\{(\mathbf{x}_n, y_n)\}_{n\in\mathbb{N}}$ so as to be able to predict the large-scale loss at any pixel inside the coverage area of that base. This can be formulated as an online regression problem.

In online regression, the number of variables grows steadily over time and algorithms usually retain only the most relevant terms in the sequence $\{(\mathbf{x}_n, y_n)\}_{n\in\mathbb{N}}$, conforming a *dictionary*. Each online regression algorithm defines a *hypothesis set* $\mathcal{H}$ which restricts the form of the learned estimate $\hat{f}$. To avoid overfitting, it is common to *regularize* such that function estimates with smaller norm—minimizing the norm favors simpler solutions more likely to generalize—are preferred.

We focus on *kernel-based* regressions in which hypotheses within $\mathcal{H}$ are nonlinear transformations onto a reproducible kernel Hilbert space (RKHS). A kernel function, as anticipated, measures the similarity of measurements, i.e., given two locations, how similar their losses are. The choice of a kernel allows an expert to express prior knowledge about the domain structure, which can significantly speed up the learning.

7.2.1 RKHS Map Formulation

A kernel $\kappa : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ on pairs of inputs implicitly defines a RKHS $\mathcal{H}$ with inner product

$$\langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \rangle = \kappa(\mathbf{x}_1, \mathbf{x}_2) \tag{7.2}$$

and norm

$$\|\phi(\mathbf{x}_1)\| = \sqrt{\kappa(\mathbf{x}_1, \mathbf{x}_1)} \tag{7.3}$$

and, for $\mathcal{H}$ to be well-defined, κ has to be positive definite.

The representer theorem [16] states that, for several regularization methods, the optimal function estimate $f^\star$ lies in the finite-dimensional span of the training measurements mapped to the Hilbert space, i.e.,

$$f^\star = \sum_n \alpha_n \, \kappa(\mathbf{x}_n, \cdot). \tag{7.4}$$

The RKHS $\mathcal{H}$ exhibits the so-called reproducing property whereby the evaluation of $f^\star$ at a location $\mathbf{x}$ is

$$\begin{aligned} f^\star(\mathbf{x}) &= \langle f^\star, \phi(\mathbf{x}) \rangle \\ &= \sum_n \alpha_n \, \kappa(\mathbf{x}_n, \mathbf{x}). \end{aligned} \tag{7.5}$$

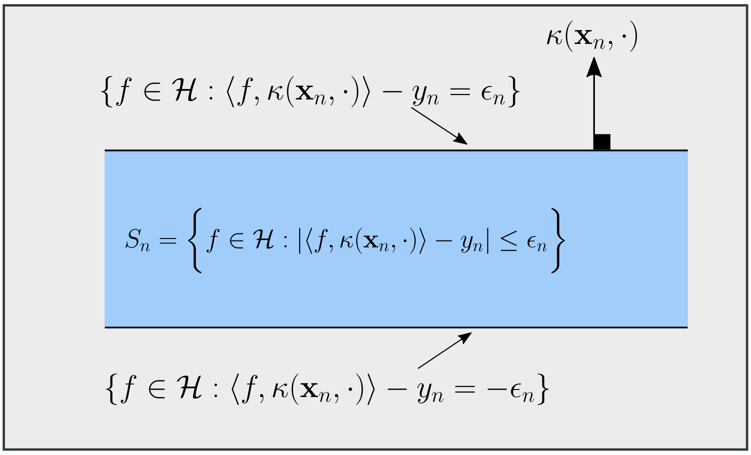

$\kappa(\mathbf{x}_n, \cdot)$
$\{f \in \mathcal{H} : \langle f, \kappa(\mathbf{x}_n, \cdot)\rangle - y_n = \epsilon_n\}$
$S_n = \left\{ f \in \mathcal{H} : |\langle f, \kappa(\mathbf{x}_n, \cdot)\rangle - y_n| \leq \epsilon_n \right\}$
$\{f \in \mathcal{H} : \langle f, \kappa(\mathbf{x}_n, \cdot)\rangle - y_n = -\epsilon_n\}$

Figure 7.1: Set S_n of functions that can approximate y_n with a maximum error of ϵ_n [17].

The online regression problem consists in estimating the coefficients $\{\alpha_n\}_{n \in N}$ that constitute $f^\star$.

Example 1. Consider the radial basis function kernel

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = e^{-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|}{D}}. \tag{7.6}$$

The optimal function estimate has the form

$$\begin{aligned} f^\star(\mathbf{x}) &= \langle f^\star, \phi(\mathbf{x})\rangle && (7.7) \\ &= \sum_n \alpha_n \, e^{-\frac{\|\mathbf{x}_n - \mathbf{x}\|}{D}}. && (7.8) \end{aligned}$$

The coefficients α_n can be obtained in batch or online fashions. Batch methods include the Support Vector Machine (SVM) regression. Online methods include the Adaptive Projected Subgradient Method (APSM), which is based on the projection onto convex sets [17]; this method models the measurements and prior knowledge as convex sets and tries to find the intersection thereof. We adopt APSM and incorporate to it a relatively simple but effective sparsification scheme.

7.3 Adaptive Projected Subgradient Method

Intuitively, APSM maintains a function estimate $\hat{f}_n$ that is updated each time a new measurement $(\mathbf{x}_n, y_n)$ arrives. The new function estimate $\hat{f}_{n+1}$ is obtained by projecting $\hat{f}_n$ onto a set that bounds the prediction error for each measurement by a predefined constant ϵ_n. It is possible that the bound ϵ_n is not satisfied for a limited number of measurements; in this case, APSM generates a sequence of function estimates that converges to a point arbitrarily close to the intersection of the remaining measurements [17]. This is important since finding a function that explains *all* measurements may be unfeasible.

For a location $\mathbf{x}_n$, the set of functions that can predict y_n with a maximum error of ϵ_n, i.e,

$$S_n = \left\{ f \in \mathcal{H} : |\langle f, \kappa(\mathbf{x}_n, \cdot) \rangle - y_n| \leq \epsilon_n \right\} \tag{7.9}$$

forms a hyperslab, depicted in Fig 7.1. The intersection

$$S = \bigcap_{n \in \mathbb{N}} S_n \tag{7.10}$$

represents a set of functions that can explain the training measurements. The APSM algorithm uses the last q measurements to produce a sequence of functions

$$\{\hat{f}_n\}_{n \in \mathbb{N}} \subset \mathcal{H} \tag{7.11}$$

where $\hat{f}_n$ is an estimation of $f^\star$ that asymptotically converges to an unspecified point in the solution set S.

Let $P_{S_n}(f)$ indicate the projection over the hyperslab S_n of a function $f \in \mathcal{H}$, given by

$$P_{S_n}(f) = f + \beta_f \, \kappa(\mathbf{x}_n, \cdot) \tag{7.12}$$

where

$$\beta_f = \begin{cases} \dfrac{y_n - \langle f, \kappa(\mathbf{x}_n, \cdot)\rangle - \epsilon_n}{\kappa(\mathbf{x}_n, \mathbf{x}_n)} & \langle f, \kappa(\mathbf{x}_n, \cdot)\rangle - y_n < -\epsilon_n \\ 0 & |\langle f, \kappa(\mathbf{x}_n, \cdot)\rangle - y_n| < \epsilon_n \\ \dfrac{y_n - \langle f, \kappa(\mathbf{x}_n, \cdot)\rangle + \epsilon_n}{\kappa(\mathbf{x}_n, \mathbf{x}_n)} & \langle f, \kappa(\mathbf{x}_n, \cdot)\rangle - y_n > \epsilon_n. \end{cases}$$

Denoting by

$$\mathcal{I}_n = \{n - q + 1, n - q + 2, ..., n\} \tag{7.13}$$

the last q measurements at time n, the sequence $\{\hat{f}_n\}$ is generated as [17]

$$\hat{f}_{n+1} = \hat{f}_n + \mu_n \left(\sum_{i \in \mathcal{I}_n} w_{i,n} P_{S_i}(\hat{f}_n) - \hat{f}_n \right) \tag{7.14}$$

where $\mu_n \in (0, 2M_n)$ is the step size, M_n is a scalar given by

$$M_n = \begin{cases} \dfrac{\sum_{i \in \mathcal{I}_n} w_{i,n} \|P_{S_i}(\hat{f}_n) - \hat{f}_n\|^2}{\|\sum_{i \in \mathcal{I}_n} w_{i,n} P_{S_i}(\hat{f}_n) - \hat{f}_n\|^2} & \hat{f}_n \notin \bigcap_{i \in \mathcal{I}_n} S_i \\ 1 & \text{otherwise} \end{cases}$$

and $w_{i,n}$ are weights satisfying

$$\sum_i w_{i,n} = 1. \tag{7.15}$$

A schematic interpretation of (7.14) is presented in Fig 7.2. At step n, $\hat{f}_n$ is projected onto the q most recent sets and the weighted sum of these projections is computed. The update direction is obtained by subtracting from the projection the previous $\hat{f}_n$ and the step size is μ_n. The parameter q regulates a tradeoff between performance and complexity: a larger q accelerates the convergence at the expense of higher computational cost.

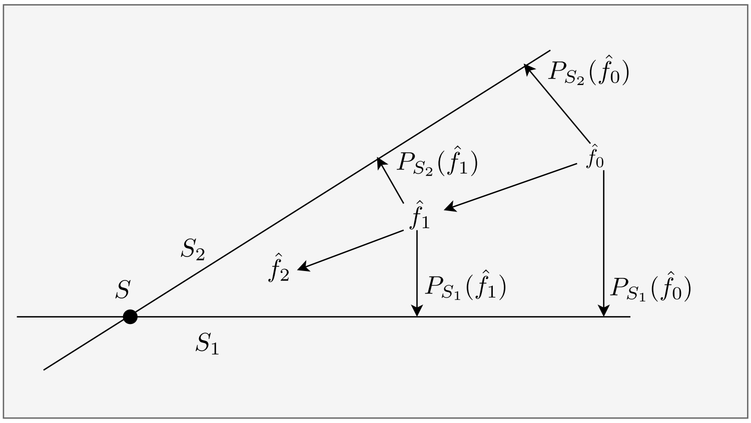


Figure 7.2: Sequence of functions f_n that converge to the intersection of two convex sets S_1 and S_2 [17].

7.4 Dual-Kernel Approach

Hilbert spaces associated with a single kernel function can give rise to relatively complex regression functions. Even more complex regression functions can be obtained by allowing for multiple kernels, be it one kernel function with different parameters or actually distinct kernels. Multikernel learning thus increases the flexibility of the regression, although at the expense of enhanced susceptibility to overfitting, i.e., of modeling at the training locations perfectly while failing to accurately predict elsewhere. Also, the complexity increases in proportion to the number of kernel functions.

To reap the benefits of having multiple kernels while avoiding these complications, the number and type of kernels should be chosen carefully on the basis of the structure of the problem. In our case, as explained in the introduction, the nature of the problem invites having two kernels.

- A DC-component kernel, slowly changing over distance as pertains to a quantity that is fixed over the coverage area and rather stable over time.

- A varying-component kernel, faster changing.

For the varying component, a sensible choice is the correlation function of the shadowing as reported in [8,14], namely $\exp(-\|\mathbf{x}_1 - \mathbf{x}_2\|/D_{\mathsf{var}})$ where D_{var} is the shadowing correlation distance. In turn, for the DC component, any slowly changing (possibly even constant) function can serve, and in particular $\exp(-\|\mathbf{x}_1 - \mathbf{x}_2\|/D_{\mathsf{DC}})$ with $D_{\mathsf{DC}} \gg D_{\mathsf{var}}$.

Altogether then, we apply the dual-kernel function

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|}{D_{\mathsf{DC}}}\right) + \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|}{D_{\mathsf{Var}}}\right) \tag{7.16}$$

which, advantageously, consists of two terms with different parameters, but the same Laplacian form. (Certain alternatives to the Laplacian function, say Gaussian, also perform satisfactorily; as long as it can capture the behavior of shadowing, the precise shape of the constituent function is not critical.)

7.5 Sparsification

The act of determining which measurements conform the dictionary is called sparsification. To sparsify, one can project $\hat{f}_n$ onto a ball of a certain norm and keep only those dictionary elements with a relatively large contribution to $\|\hat{f}_n\|$, thereby effecting a regularization. The projection onto the ball $[0, \delta]$ can be obtained as [17]

$$\hat{f}_{n+1} = P_{B[0,\delta]}\left(\hat{f}_n + \mu_n \left(\sum_{i \in \mathcal{I}_n} w_{i,n} \, P_{S_i}(\hat{f}_n) - \hat{f}_n\right)\right) \tag{7.17}$$

where

$$P_{B[0,\delta]}(f) = \begin{cases} f & \|f\| \leq \delta \\ f/\|f\| & \|f\| > \delta. \end{cases} \tag{7.18}$$

(This projection can also be regarded as a forgetting factor, whereby older history is progressively removed from memory, fostering adaptivity.) Af-

Table 7.1: APSM parameters

	D_{DC}	D_{Var}	q	μ	δ	ϵ_n
Campus	∞	50	1	0.5	90	.1
Urban	∞	450	1	0.5	130	.1

ter the projection, the indices of those dictionary elements with considerable contribution can be identified as the set

$$\mathcal{J}_n = \left\{ j \mid \frac{(\alpha_{n,j})^2}{\sum_{\ell \in \mathcal{L}_n} (\alpha_{n,\ell})^2} > \tau \right\} \tag{7.19}$$

where τ is a threshold, $\mathcal{L}_n$ is the set of indexes of the dictionary elements at step n and $\{\alpha_{n,\ell}\}_{\ell \in \mathcal{L}_n}$ are the corresponding coefficients for the projected locations in the Hilbert space. One can think of (7.19) as removing those elements in (7.4) that have a negligible effect on the regression. The threshold τ regulates the tradeoff between complexity and accuracy: by increasing τ, the dictionary becomes sparser and complexity diminishes, but the accuracy drops, and vice versa.

7.6 Numerical Evaluations

In this section, the dual-kernel algorithm is evaluated for campus and urban scenarios. As in [9], to facilitate later comparisons, 70% of the available measurements are randomly chosen for training with the rest left for testing; the normalized squared error is averaged over 1000 random splits between training and test to produce the normalized mean-squared error (NMSE). The dictionary is updated every time a measurement is processed. For the APSM settings, refer to Table 7.1.

7.6.1 Campus Scenario

For this scenario, we apply the database *cu/cu_wart* [18] that provides 1277 measurements of the local-average received power in an 802.11 network with high-resolution pixels (3 m).

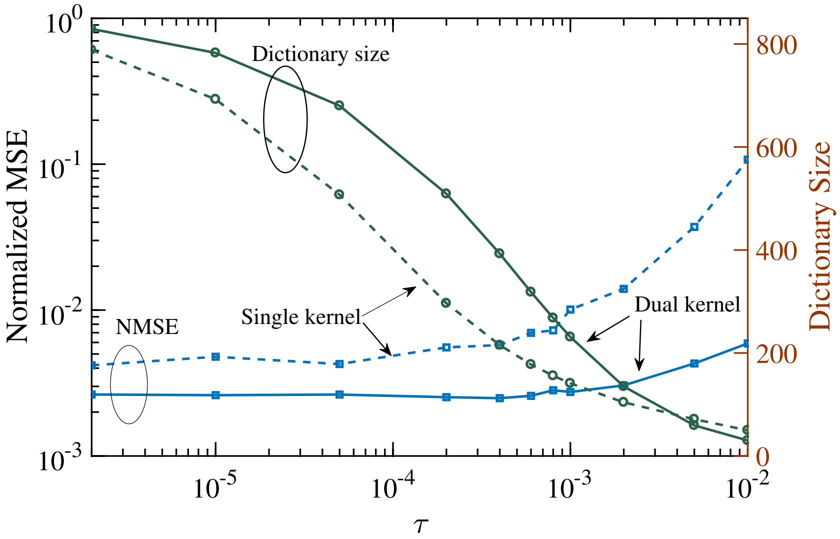


Figure 7.3: NMSE (left-hand axis) and dictionary size (right-hand axis) as function of τ, with and without the DC kernel, in the campus scenario.

To begin with, let us gauge the effect of adding the DC kernel. Fig. 7.3 presents the NMSE and the dictionary size, as function of the sparsification threshold τ, with and without the DC kernel. The addition of this kernel reduces the NMSE for a given dictionary size or, equivalently, it reduces the dictionary size for a given NMSE.

Fig. 7.4 illustrates the convergence of the NMSE with the dictionary size set to 500 and therefore, applying Fig. 7.3, with $\tau = 1.3 \cdot 10^{-4}$ for dual kernel and $\tau = 4 \cdot 10^{-5}$ for single kernel. The addition of the DC kernel improves the performance markedly, reaching at $n = 250$ the NMSE that requires $n = 890$ without the DC kernel. In turn, the final NMSE shrinks by a factor of 1.7.

Finally, Table 7.2 compares the dual-kernel algorithm with some alternatives: the single-kernel algorithm, a multikernel version [9], and a batch SVM baseline. The dual-kernel approach outperforms both online alternatives in terms of NMSE, approaching the batch SVM accuracy. The reduction in NMSE does not come at the expense of an increased variability thereabout; rather, the standard deviation of the squared error

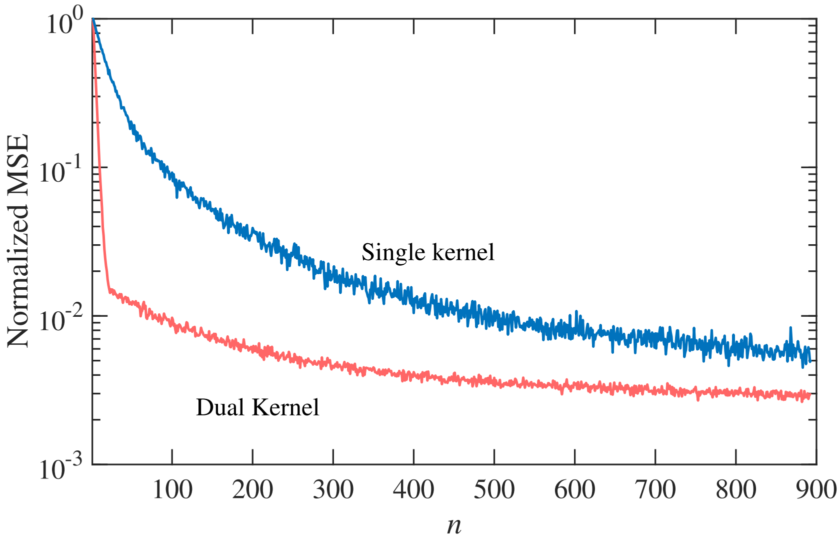


Figure 7.4: NMSE as function of n, with and without the DC kernel, in the campus scenario. The dictionary size is 500.

Table 7.2: NMSE and error standard deviation in the campus scenario.

	Batch SVM	**Online Dual Kernel**	Online Single Kernel [9]	Online Multikernel [9]
NMSE $\times 10^3$	1.7	**2.7**	4.4	6
std $\times 10^3$	2.5	**3.9**	11.2	—

also abates relative to the single-kernel algorithm. (For the multikernel solution, the variance is not available.)

7.6.2 Urban scenario

The second scenario corresponds to ray-tracing data for the central part of Berlin [19]. Data for 193 BSs with a 50-m pixel resolution over an area of 12 × 12 km is reported in this database. The APSM settings are as per Table 7.1. To make the results comparable with those in the campus scenario, the same number of data points is used, namely 1277, which are

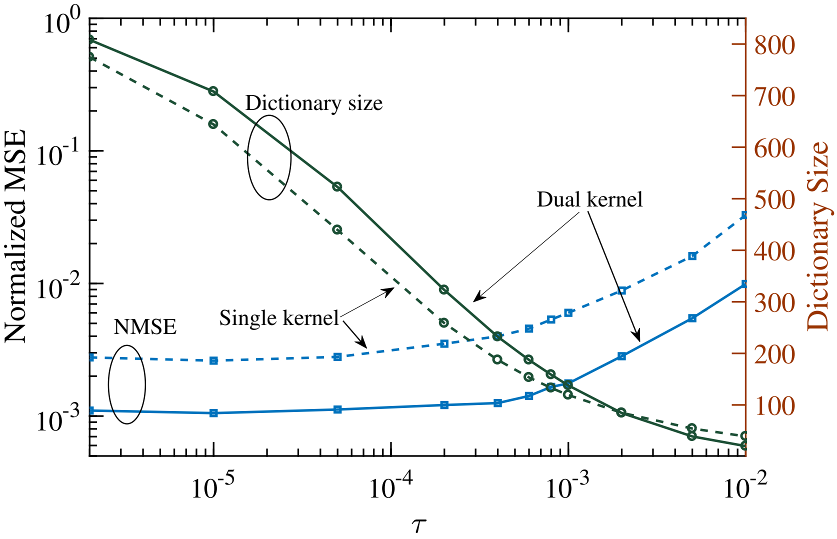


Figure 7.5: NMSE (left-hand axis) and dictionary size (right-hand axis) as function of τ, with and without the DC kernel, in the urban scenario.

randomly selected at each of 1000 passes over which the error is averaged. We set the coverage threshold to 140 dB, a reasonable value for urban settings.

The effect of adding the DC kernel is shown in Fig. 7.5, where a significant benefit is again observed. Then, in Table 7.3, we compare the dual-kernel approach with its single-kernel brethren and with the batch SVM baseline. The reduction in both NMSE and standard deviation is even more pronounced here than in the campus environment, even if the absolute standard deviations are larger on account of the more challenging nature of an urban setting.

7.7 Summary

A clean separation of the DC component from the rest of the large-scale losses, via two suitable kernel functions, benefits the online reconstruction of large-scale loss maps. The accuracy increases with respect to the

Table 7.3: NMSE and error standard deviations in the urban scenario.

	Batch SVM	**Online Dual Kernel**	Online Single Kernel
NMSE $\times 10^3$	0.4	$\mathbf{1.1}$	2.8
std $\times 10^3$	3.5	$\mathbf{5}$	20

single-kernel version, but also with respect to a multikernel solution. This confirms that the number of kernels should be chosen carefully to match the structure of the problem, and adding unnecessarily many kernels may end up being detrimental. With two kernels, the performance approaches the baseline represented by more complex batch schemes, where, rather than a limited-size dictionary, all measurements must be stored.

To reinforce the above insight it is worth mentioning that, in Section 7.6, the dual-kernel approach has been run with the parameter q (which, recall, indicates the number of recent measurement over which APSM projects at every step) set to $q = 1$, and yet this has sufficed to outperform the single-kernel result, which has been run with $q = 20$. This confirms the effectiveness of the DC kernel at tracking the map's average, something that without a DC kernel requires a larger value of q and therefore higher complexity.

References

[1] M. Hata, “Empirical formula for propagation loss in land mobile radio services,” *IEEE Trans. Veh. Technol.*, vol. 29, no. 3, pp. 317–325, 1980.

[2] “Urban transmission loss models for mobile radio in the 900 and 1800 MHz bands,” European Cooperation in the Field of Scientific and Technical Research EUROCOST 231, Tech. Rep., September 1991.

[3] V. Erceg, L. J. Greenstein, S. Y. Tjandra, S. R. Parkoff, A. Gupta, B. Kulic, A. A. Julius, and R. Bianchi, “An empirically based path

loss model for wireless channels in suburban environments," *IEEE J. Select. Areas Commun.*, vol. 17, no. 7, pp. 1205–1211, Jul 1999.

[4] C. Phillips, D. Sicker, and D. Grunwald, "Bounding the error of path loss models," in *Proc. 2011 IEEE DySPAN*, may 2011, pp. 71–82.

[5] C.-F. Yang, B.-C. Wu, and C.-J. Ko, "A ray-tracing method for modeling indoor wave propagation and penetration," *IEEE Trans. Antennas Propagat.*, vol. 46, no. 6, pp. 907–919, 1998.

[6] K. E. Stocker, B. E. Gschwendtner, and F. M. Landstorfer, "Neural network approach to prediction of terrestrial wave propagation for mobile radio," in *IEE Proceedings H (Microwaves, Antennas and Propagation)*, vol. 140, no. 4, 1993, pp. 315–320.

[7] E. Ostlin, H.-J. Zepernick, and H. Suzuki, "Macrocell path-loss prediction using artificial neural networks," *IEEE Trans. Veh. Techn.*, vol. 59, no. 6, pp. 2735–2747, 2010.

[8] R. D. Taranto, S. Muppirisetty, R. Raulefs, D. Slock, T. Svensson, and H. Wymeersch, "Location-aware communications for 5g networks: How location information can improve scalability, latency, and robustness of 5g," *IEEE Signal Processing Mag.*, vol. 31, no. 6, pp. 102–112, 2014.

[9] M. Kasparick, R. L. G. Cavalcante, S. Valentin, S. Stańczak, and M. Yukawa, "Kernel-Based Adaptive Online Reconstruction of Coverage Maps With Side Information," *IEEE Trans. Veh. Technol.*, vol. 65, no. 7, pp. 5461–5473, jul 2016.

[10] D. Romero, S.-J. Kim, G. B. Giannakis, and R. Lopez-Valcarce, "Learning power spectrum maps from quantized power measurements," *IEEE Trans. Signal Processing*, vol. 65, no. 10, pp. 2547–2560, 2017.

[11] L. S. Muppirisetty, T. Svensson, and H. Wymeersch, "Spatial wireless channel prediction under location uncertainty," *IEEE Trans. Commun.*, vol. 15, no. 2, pp. 1031–1044, Feb 2016.

[12] E. Dall'Anese, S. J. Kim, and G. B. Giannakis, "Channel Gain Map Tracking via Distributed Kriging," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 1205–1211, 2011.

[13] J. Chen, U. Yatnalli, and D. Gesbert, "Learning radio maps for UAV-aided wireless networks: A segmented regression approach," in *Proc. IEEE Int. Conf. Commun.* IEEE, may 2017, pp. 1–6.

[14] M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *IEE Electron. Lett.*, vol. 27, no. 23, pp. 2145–2146, nov 1991.

[15] H. Holma and A. Toskala, *LTE for UMTS: Evolution to LTE-advanced.* John Wiley & Sons, 2011.

[16] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Computational Learning Theory.* Springer Berlin Heidelberg, 2001, pp. 416–426.

[17] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Processing Mag.*, vol. 28, no. 1, pp. 97–123, Jan 2011.

[18] C. Phillips and E. W. Anderson, "CRAWDAD dataset cu/cu_wart (v. 2011-10-24)," Downloaded from https://crawdad.org/cu/cu_wart/20111024, Oct. 2011.

[19] MOMENTUM, "Models and Simulations for Network Planning and Control of UMTS," 2004. [Online]. Available: http://www.zib.de/momentum/data.php

Chapter 8

Conclusion and Future Work

> *When something is important enough, you do it even if the odds are not in your favor.*
>
> Elon Musk

8.1 Conclusion

Parametric optimization appears in most aspects of wireless communication, where an optimization problem with a fixed structure is being solved repeatedly for different parameter values. As a prime example, we discussed power control and power allocation in cellular and cell-free settings. The advantage and properties of the proposed unsupervised-learning approach are summarized as follows:

1. It works best for problems with structure. The degrees of freedom are much less than the number of optimization variables; therefore, the proposed unsupervised-learning approach effectively samples the parameter space and generalizes the solution across the parameter space.

2. Different fairness criteria can be tackled. In this thesis, we applied max-min and max-product criteria, with the latter being easily extendable to the sum-rate case. Having a discontinuous structure,

the max-min criterion was the hardest to tackle, which was accomplished by defining a soft max-min loss function—a continuous approximation to the max-min criteria that converges to max-min in the limit.

3. The unsupervised-learning approach works for both convex and non-convex problems. The solution accuracy, however, depends on how efficient the gradient descent is in solving a single instance of the problem. The unsupervised NN reaches the accuracy of the convex solver whenever applicable.

4. We can incorporated expert knowledge into the structure of the proposed unsupervised-learning approach. Often, a feedforward NN is treated as a black box that can approximate a function. This function can be modified by changing either the database or the loss function of the NN. The unsupervised-learning approach uses the later, introducing a custom loss function. As a result, the expert can modify the behavior of the unsupervised NN by fine-tuning the loss function. The expressive power of the NN, combined with expert knowledge, outperforms the domain expert both in terms of performance and the time spent deriving the solution.

The extension of the fractional power control to cell-free networks was another area of focus in this thesis. The developed solution:

1. Is simple and can be computed analytically.

2. Performs satisfactory and almost optimally (in the uplink case).

Finally, we developed maps of received power in a 2-D plane. Such maps could be useful in resource allocation, interference management, and proactive resource allocation. The developed dual-kernel method has the following properties:

1. It is an online method that can track environmental changes.

2. Kernel-based: expert knowledge can be implemented by choosing a proper kernel function—the measure of similarity in pathloss value for two close-by users.

3. Superior performance: the dual-kernel method, having the right amount of complexity, achieves fast convergence and low variance in pathloss prediction.

8.2 Future Work

It would be of interest to extend the unsupervised-learning approach to new domains such as zero-forcing beamforming or new objectives such as sum-rate. As long as the problem at hand is a parametric optimization and it has structure, we can apply the unsupervised-learning approach.

Improving the unsupervised-learning approach by incorporating convolutional layers is another interesting future direction. For a parametric optimization without any structure, the feedforward NN is an inevitable component of the unsupervised-learning approach, assigning at least one neuron for each parameter in the input layer. If the parameter space is highly structured and arrangeable in a meaningful way, convolutional layers can be further used. Such an approach is applicable to higher-dimensional problems, reducing the complexity of the NN drastically.