Technische Universität Berlin

Institut für Telekommunikationssysteme

Fachgebietes Netzwerk-Informationstheorie

Fakultät IV

Einsteinufer 25

10587 Berlin

http://www.netit.tu-berlin.de

Master Thesis

Multi-Kernel Methods for Channel Gain Cartography

Amer Alattar

Matriculation Number: 370336

07.05.2018

Referent:	Prof. Dr.-Ing. Slawomir Stanczak
	Chair of Network Information Theory
Co-Referent:	Prof. Dr.-Thomas Sikora
	Chair of Telecommunication Systems
Supervisors:	Miguel Angel Gutierrez Estevez, M.Sc.
	Fraunhofer Heinrich Hertz Institute (HHI)

Acknowledgment

I would like to express my grateful thanks to my first reviewer Prof. Dr.-Ing. Slawomir Stanczak for his guidance and support during my master

Furthermore I would like to thank my second reviewer Prof. Dr.-Ing. Thomas Sikora for his time and effort dedicated to review and discuss my work progress.

I owe deepest gratitude to my direct supervisor MSc. Miguel Angel Gutierrez Estevez. He offered me the adequate time, concern, and advice to help me grow in the research field. This work would not have been done without his continues guidance and generous tips.

I want to dedicate this thesis to my loving mother, father, and my mentor Ing Imad Amini. Their words of encouragement pushed me further and helped me get through the hardest times.

I also dedicate this thesis to my lovely friends: Abbud, Yasser. I hope, it will motivate and encourage them to get through diculties in pursuing their dreams.

Hereby I declare that I wrote this thesis myself with the help of no more than the mentioned literature and auxiliary means.

Berlin, 07.05.2018

. . . . . .

Amer Alattar

Abstract

In this thesis, we address the problem of reconstructing channel gain maps from sensor measurements in wireless networks. For that purpose, and similar to previous studies, we model the shadow fading with a linear tomographic projection technique [15, 16, 27]. More concretely, the spatial loss field (SLF) captures the absorption generated by objects in a field. The long-term shadow fading between two points is then estimated as the integral of the SLF and a weight function that models the influence between each position of a map and the link.

Our work builds upon [41], which is, to the best of our knowledge, the first work to reconstruct the shadowing attenuation in a map with barely no assumption on the structure of the weights function. A nonparametric regression in reproducing kernel Hilbert spaces (RKHSs) was applied to learn both the SLF and the weight function, and then reconstruct the shadowing attenuation between any two points in a map. RKHSs is a popular method in the world of machine learning due to the simplicity and computational efficiency of kernel methods relative to solving non-linear problems [7, 43].

However, The original problem is highly ill-posed. Therefore, we propose and evaluate an algorithm which adds some structure to the problem by approximating the weight function with non-linear kernels. More concretely, we rely on a multi-kernel method as a non-linear approach to approximate the weight function. The resulting problem is non-convex, but we use a block-descent technique to fix one of the variables while solving for the other one. The two new problems are convex. To assess the proposed algorithms, we evaluated a 10×10 map for different SNR levels, several frequencies and different room layouts. Results show improvements for any scenario when compared with a baseline algorithm from the literature.

Zusammenfassung

In dieser Arbeit behandeln wir das Problem, Kanalverstärkungskarten anhand von Sensorenmessungen in drahtlosen Netzwerken zu rekonstruieren. Ähnlich zu bereits durchgeführten Arbeiten wird die Dämpfung durch Hindernisse, die die Wellenausbreitung beeinflussen (Shadow Fading oder Shadowing), mit einem linearen tomographischen Projektionsverfahren modelliert [15,16,27]. Genauer, das räumliche Pfadverlustfeld (Spatial Loss Field, SLF) erfasst die Dämpfungen, die durch Objekte in einem Feld hervorgerufen werden. Auf das SLF wird dann eine Gewichtungsfunktion angewendet, die den Einfluss jeder Position auf das Shadow Fading einer drahtlosen Verbindung modelliert.

Unsere Arbeit basiert auf [41], der ersten Arbeit, die durch Shadow Fading verursachte Dämpfungen in einer Karte rekonstruiert und kaum Annahmen über die Struktur der Gewichtungsfunktion macht. Eine nicht-parametrische Regression in reproduzierenden Kernen in Hilberträumen (Reproducing Kernel Hilbert Spaces, RKHS) wurde verwendet, um das SLF und die Gewichtungsfunktion zu erlernen und damit die Dämpfung durch Shadowing zwischen zwei beliebigen Punkten der Karte zu rekonstruieren. RKHSs sind eine weit verbreitete Methode im maschinellen Lernen, da sie im Vergleich zu nichtlinearen Schätzungsfunktionen weniger komplex sind und eine höhere Berechnungseffizienz aufweisen [7, 43].

Dennoch ist das Problem inkorrekt gestellt. Deshalb schlagen wir einen Algorithmus vor und werten ihn aus, der dem Problem Struktur hinzufügt, indem die Gewichtungsfunktion mit nichtlinearen Kernen approximiert wird. Konkreter verwenden wir eine Multi-Kerne-Methode als nichtlinearen Ansatz, um die Gewichtungsfunktion zu schätzen. Das resultierende Problem ist nichtkonvex, aber wir nutzen eine Blockabstiegstechnik, um eine Variable festzulegen und das Problem mit der anderen zu lösen. Die zwei neuen Probleme sind konvex. Um den vorgeschlagenen Algorithmus zu bewerten, haben wir eine 10 x 10 Karte für verschiedene Signal-Rausch-Verhältnisse, mehrere Frequenzen und unterschiedliche räumliche Anordnungen ausgewertet. Die Resultate zeigen Verbesserungen in allen Szenarien im Vergleich zu einem Basisalgorithmus aus anderen Arbeiten.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation and Goal of the Thesis

Wireless networks are ubiquitous, and with the upcoming of the 5G systems not only people's lives are going to be impacted but also many industries are expected to undergo a revolution. However, physical effects of wireless signals propagation such as attenuation will always set limits to achievable performance. Therefore, an accurate representation of radio frequency environments is a highly desired feature, due to its crucial role in many aspects of designing and optimizing modern radio networks. Characterizing the geographical path loss distribution of radio channels in the form of so-called radio maps is considered to be essential knowledge for many applications in wireless networks. Furthermore, modern networks require not only the knowledge of the current estimated coverage map state, but also the availability expected information in a reliable manner will enhance the performance of the network and improve the usage of the scarce wireless resources.

Many approaches for reconstructing channel gain maps have been proposed over the years. This thesis relies on the approach of channel gain cartography, also known as radio frequency tomography, a groundbreaking geostatistics-inspired application that enables characterizing the RF environment of any location in space. The most appealing feature of the utilized approach consists in the non-trivial capability of inferring the channel gain between any two given points in a map, rather than just the channel gain between a base station and the users connected to it. Moreover, this ability to learn the channel characteristics is based only on the measurements taken from a collaborating sensor network.

Channel gain cartography builds upon the fact that different objects absorb electromagnetic waves differently, and consequently, they have different absorption factors. Therefore, if we are able to measure or learn the absorption factors in each pixel of a map, then we would be able to obtain information about the size and the position of these objects.

This approach is beneficial for many applications where the communication is peer to peer, such

as device-to-device communtication (D2D), machine-type communications(MTC) or cognitive radio (CR). A plethora of theoretical and empirical channel cartography works have been proposed in the literature, mainly for applications related to learning the position and characteristics of objects located in a map, such as objects detection, see-through walls, motion tracking, and many others [15, 32, 51–53].

The abilities of channel gain cartography to A2A maps may improve the usage of the limited wireless resources and optimize the network quality of service (QoS).

In order to reconstruct the channel gain map using channel gain cartography, we propose and evaluate an algorithm that utilizes the powerful methods of reproducing kernel Hilbert spaces (RKHS). We choose this method according to their ability to solve nonlinear problems in an efficient manner [7, 43]. The main strength of kernel methods lies in their ability to efficiently map their input data space into a higher-dimensional (even infinite dimensional) feature space by utilizing simple kernel functions, such kernels perform a nonlinear projection to a high dimensional feature space without increasing the tunable parameters, in which the respective problems become linearly solvable.
Kernel methods lean on the assumption that close data points in the feature space should have similar output, therefore we only require a means of measuring similarity in the feature space. However, computing every coordinate of a vector in the feature space might be impossible if the feature space has very high or infinite dimensions. Furthermore, it is sufficient to compute the inner products in the feature space by using the kernel function associated with the reproducing kernel Hilbert space. This process is known in the literature as the kernel trick [10, 17]. Kernel methods are used in many machine learning tools such as support vector machines (SVM) and ridge regression. Moreover, these machine learning tools are frequently used in the field of map reconstruction.

1.2 Outline

The mathematical background and an introduction to the kernel and multi-kernel learning methods are presented in chapter 2. Chapter 3 gives an overview of radio map reconstruction in general and reconstructing path loss maps in more detail. In chapter 4, the system model and the problem statement are presented, in addition to multi-kernel methods for channel gain cartography. Chapter 5 presents the experimental results and gives a numerical evaluation of the performance. Chapter six summarizes the work and gives an outlook for future work.

2 Kernel Methods

This chapter introduces the most important theoretical concepts and mathematical fundamentals of multiple kernel learning which defines in this thesis the tool used by channel gain cartography to reconstruct the path loss maps.

2.1 Preliminaries

Vector Spaces: A vector space X is a non-empty collection of elements (the elements of a vector space are called vectors) on which two operations are defined [29].

Let X be a vector space and F scalar field $X \times X \to X,\ F \times X \to X$:

1- Addition : $\forall\ \mathbf{x}, \mathbf{y} \in X$ there is $\mathbf{x}+\mathbf{y} \in X$.

2- Scalar Multiplication: $\forall\ \mathbf{x} \in X$ and $\forall\ \alpha \in \mathbb{F}$, $\alpha \cdot \mathbf{x} \in X$.

So that for each pair of elements $\mathbf{x}$,$\mathbf{y}$ in X there is a unique element $\mathbf{x}+\mathbf{y}$ is defined as the sum of $\mathbf{x}$ and $\mathbf{y}$ in X, and for each scalar multiplication $\alpha \cdot \mathbf{x}$ exists a unique element $\alpha\mathbf{x}$ defined as the scalar multiple of $\mathbf{x}$ by α in X.

The two vector space operations satisfy common axioms [29, 40]:

1. Commutativity of addition: for each pair of elements: $\mathbf{x} + \mathbf{y} \in X, \mathbf{x} + \mathbf{y} = \mathbf{y} + \mathbf{x}$.
2. Associativity of addition: $\forall\ \mathbf{x}, \mathbf{y}, \mathbf{z} \in X,\ (\mathbf{x} + \mathbf{y}) + \mathbf{z} = \mathbf{x} + (\mathbf{y} + \mathbf{z})$.
3. Identity element of addition: there exists an element $0 \in X$, called the zero vector, so that: $\mathbf{x} + 0 = \mathbf{x},\ \ \forall\ \mathbf{x} \in X$.
4. Inverse elements of addition: for each element $\mathbf{x} \in X$ there exists an element $\mathbf{y} \in X$ such that $\mathbf{x} + \mathbf{y} = 0$.
5. Compatibility of scalar multiplication with field multiplication: for each scalar α, β and each $\mathbf{x} \in V$: $(\alpha(\beta)\mathbf{x}) = \alpha((\beta\mathbf{x}))$.
6. Identity element of scalar multiplication: for each element $\mathbf{x} \in X,\ 1\mathbf{x} = \mathbf{x}$ 1 is defined as the multiplicative identity in F.

7. Distributivity of scalar multiplication with respect to vector addition: for each pair of scalars α, and each $\mathbf{x} \in X$: $\alpha(\mathbf{x}+\mathbf{y}) = \alpha\mathbf{x} + \alpha\mathbf{y}$.

8. Distributivity of scalar multiplication with respect to field addition: for each pair of scalars α,β and each $\mathbf{x} \in X$: $(\alpha+\beta)\mathbf{x} = (\alpha\mathbf{x} + \beta\mathbf{x})$.

Basis: Let X be a vector space and S a set of vectors in X then if there exists a finite number of distinct vectors $\{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n\} \in S$ such that $\{\alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + ... \ \alpha_n\mathbf{x}_n\} = 0$ with scalars $\{\alpha_1, \alpha_2, ... \ \alpha_n\}$ not all equal to zero, then S is said to be Linearly independent [48].
If S is not empty and every vector in X can be expressed as a linear combination of vectors from S, then S is said to *generate*(X).
In the vector space X the set of vectors S is called a basis of X if it is linearly independent and it generates X.
A vector space having a finite number of basses is said to be finite dimensional and all other vector spaces are said to be infinite dimensional.

Normed vector Spaces: Let X be a vector space, a norm over X is a real-valued function that maps each vector $\mathbf{x} \in X$ into a real number and is denoted by $\|\mathbf{x}\|$, the norm satisfies the following axioms [48]:

1. The zero vector, 0, has a length of zero; every other vector has a positive length: $\|\mathbf{x}\| \geq 0$ and $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = 0$.

2. Multiplying a vector by a positive number changes its length without changing its direction: $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$ for any scalar α and any $\mathbf{x} \in X$.

3. The triangle inequality is taking norms as distances: $\|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\| \ \forall \mathbf{x}, \mathbf{y} \in X$.

The normed linear vector space on X is the vector space X together with its norm $\|.\|$.

Inner Product Spaces: Let X be a vector space and let $\mathbf{x}, \mathbf{y}$ be arbitrary elements in X, and F is either the field of real numbers $\mathbb{R}$ or the field of complex numbers $\mathbb{C}$. The *inner product* on X is a function that assigns to every ordered pair of vectors $\mathbf{x}$ and $\mathbf{y}$ a scalar value which is a map X x $X \rightarrow F$ and denoted by [48]:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i y_i. \tag{2.1}$$

The inner product satisfies the following axioms $\forall\ \mathbf{x}, \mathbf{y}, \mathbf{z} \in X$ and $\alpha \in F$:

1. $\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$ and $\langle \mathbf{x}, \mathbf{x} \rangle = 0$ if and only if $\mathbf{x}$=0.
2. $\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x} + \mathbf{z} \rangle + \langle \mathbf{y} + \mathbf{z} \rangle$
3. $\langle \alpha \mathbf{x}, \mathbf{y} \rangle = \alpha \langle \mathbf{x}, \mathbf{y} \rangle$
4. $\langle \mathbf{x}, \mathbf{y} \rangle = \overline{\langle \mathbf{y}, \mathbf{x} \rangle}$ where ($\overline{\mathbf{x}}$ is a complex conjugate of $\mathbf{x}$).

The requirement 1 defines the positive definiteness, 2 and 3 define the linearity in the first component and 4 defines the conjugate symmetry.
A vector space together with its inner product is an *inner product space*.

Complete Vector Spaces: A sequence $\{x_n\}$ in a normed space is defend as a *Cauchy sequence* if [29]:

$$\|\mathbf{x}_m - \mathbf{x}_n\| \rightarrow 0 \text{ as } n, m \rightarrow \infty.$$

A Banach space is a vector space $X \in \mathbb{R}$ or $X \in \mathbb{C}$, which is equipped with a norm and is complete with respect to that norm, that is to say, for every Cauchy sequence $\{\mathbf{x}_n\}$ in X, there exists a limit in X,

$$\sum_{n=1}^{\infty} \|x_n\|_X < \infty \quad \Rightarrow \quad \sum_{n=1}^{\infty} x_n \ \text{ converges in } \ X.$$

A vector space is called complete if every Cauchy sequence has a limit (and therefore convergent) in this vector space.

Hilbert Spaces: A *Hilbert space* denoted as $\mathcal{H}$ is a complete linear vector space with respect to the norm produced by the inner product which is defend on $X \times X$ [29, 40, 48].
For all $\mathbf{x}, \mathbf{y} \in \mathcal{H}$ we have:

$$\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\| + \|\mathbf{y}\|, \tag{2.2}$$

where, $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ is a norm on $\mathcal{H}$ and the equality holds if and only if $\mathbf{x} = \alpha \mathbf{y}$ for some $\alpha \in \mathbb{F}$ or $\mathbf{y} = 0$, The inner product is bounded if both $\|\mathbf{x}\|$ and $\|\mathbf{y}\|$ are bounded, and several normed vector spaces can be converted to Hilbert spaces by defining an appropriate inner product.
Examples:

1. the Euclidean spaces R^n:

   The inner product is defined on the Euclidean space as:

   $$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i, y_i, \tag{2.3}$$

   $\forall\ \mathbf{x} = (x_1, x_2, ...x_n)$, $\mathbf{y} = (y_1, y_2, ...y_n) \in \mathbb{R}^{\ltimes}$.

   The resulting Euclidean norm is defined as:

   $$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{\infty} |\mathbf{x}_i|^2} \ \ \forall\ \mathbf{x} \in \mathbb{R}^{\ltimes}. \tag{2.4}$$

   The space $\mathbb{R}^{\ltimes}$ is complete with respect to the Euclidean norm, thus the Euclidean space $\mathbb{R}^{\ltimes}$ is a Hilbert space.

2. l^2 spaces: The inner product is defined on the l^2 space as:

   $$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i, \overline{y_i}, \tag{2.5}$$

   where $\langle \mathbf{x}, \mathbf{y} \rangle < +\infty$ as $\|\mathbf{x}\| < +\infty$ and $\|\mathbf{y}\| < +\infty$.

   The resulting norm in l^2 is:

   $$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{\infty} |\mathbf{x}_i|^2} \ \ \forall \mathbf{x} \in l_2. \tag{2.6}$$

   The l^2 space is also complete with respect to the l^2 norm thus the l_2 space is a Hilbert space.

3. L^2 spaces: let $\mathbf{f} = f(t),\ \mathbf{g} = g(t) \in L^2$ .The inner product is defined on the L^2 space as:

   $$\langle f(t), g(t) \rangle = \int_a^b f(t)\overline{g(t)}dt, \tag{2.7}$$

where $\langle f, g\rangle < +\infty$.

The resulting norm in L^2 is:

$$\|\mathrm{f}\|_2 = \Big(\int_{-\infty}^{+\infty} |f(t)|^2 \mathrm{d}t\Big)^{1/2} \forall\ \mathrm{f} = f(t) \in L^2. \tag{2.8}$$

The L^2 space is also complete with respect to the L^2 norm, thus the L^2 space is a Hilbert space.

Linear Regression: Regression is a set of techniques for estimating relationships between variables. The linear regression model is a statistical procedure that aims to model the relationship between scalar dependent variable $\mathbf{Y} = f(\mathbf{x}) \subseteq R$ (response variable) and one or more independent variables (predictor variables) $\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_p$, as a linear line. The main challenge in linear regression is to obtain a linear function [10, 45]:

$$f(\mathbf{x}) = \langle \mathbf{w}.\mathbf{x}\rangle + b, \tag{2.9}$$

that best models a given set of training points labeled from $\mathbf{Y} = \{1, 2, ...m\}$, where $\mathbf{w}$ is the weight function and b is called the intercept, or Y intercept.

Figure 2.1 depicts a one dimensional linear regression function, where ξ denotes the error for a particular training example.

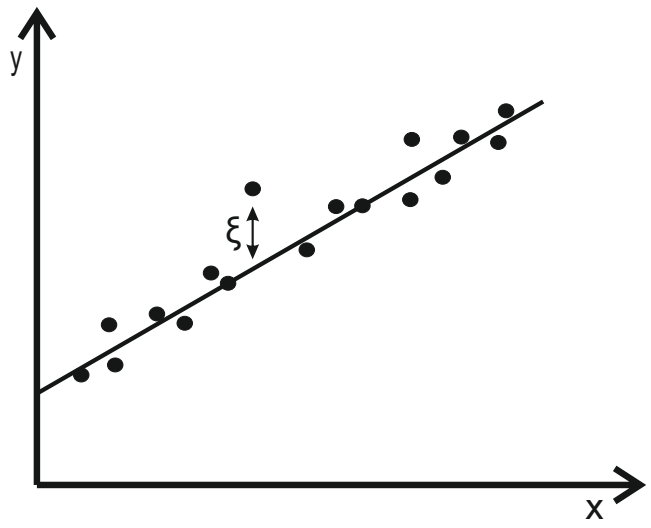

Figure 2.1: Linear regression for one dimensional function

.

Least squares: let S be a training set with $\mathbf{x}_i \in X \subseteq \mathbb{R}^n$, $\mathbf{y}_i \in \mathbf{Y} \subseteq \mathbb{R}$. The main challenge in linear regression, as mentioned in equation 2.9, is to find a linear function f that interpolates the data, such as: $f(\mathbf{x}) = \langle \mathbf{w}.\mathbf{x} \rangle + b$. Least squares approach aims to approximate the solution of over-determined systems by finding the parameters ($\mathbf{w}$,b) to minimise the sum of the squared deviation of the data [10],

$$SSE(\mathbf{w}, b) = \sum_{i=1}^{l} (y_i - \langle \mathbf{w}.\mathbf{x}_i \rangle - b)^2, \tag{2.10}$$

The function SSE is defined as the square error function which calculates the error value as the sum of squares. The function SSE can be minmised by differentiating with respect to the parameters $(\mathbf{w}, b)$ and sitting the resulting $n + 1$ linear expression to zero.

Suppose $d = 1$:

$$\frac{\partial SSE}{\partial w} = -2 \sum_{i=1}^{l} (y_i - \langle \mathbf{w}.\mathbf{x}_i \rangle - b) = 0. \tag{2.11}$$

For more generalization and letting d be arbitrary:

$$\mathbf{\Phi} = \begin{pmatrix} w_{(1)} \\ w_{(2)} \\ \vdots \\ w_{(d)} \end{pmatrix},$$

then the sum of square error function can be defined as:

$$SSE(\mathbf{\Phi}) = \sum_{i=1}^{l} (y_i - \langle \mathbf{w}.\mathbf{x}_i \rangle - b)^2 = \|\mathbf{A}\mathbf{\Phi} - \mathbf{y}\|_2^2, \tag{2.12}$$

where $\|\cdot\|_2$ is the Euclidean norm and,

$$\mathbf{y} = \begin{pmatrix} y_{(1)} \\ y_{(2)} \\ \vdots \\ y_{(d)} \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} x_1(1) & x_1(2) & ... & x_1(d) \\ x_2(1) & x_2(2) & ... & x_2(d) \\ \vdots & \vdots & \ddots & \vdots \\ x_l(1) & x_l(2) & ... & x_l(d) \end{pmatrix}.$$

This approach is defined as a standard approach in regression analysis which was presented by Gauss and Legendre independently in the 18th century depending on the idea of minimising the sum of the squared deviations of the offsets (the residuals)

Ridge Regression: Since Least squares depend on the inverse of the matrix $\mathbf{A}$, which might cause problems in computing $\hat{\mathbf{w}}LS$ if this matrix is singular or nearly singular, and the fact that any small changes to elements of $\mathbf{x}$ lead to large changes in $\mathbf{A}^{-1}$, in addition to the situation when this matrix is not full of rank we can use the following solution [10, 45]:

$$\hat{\mathbf{w}} = \left(\hat{\mathbf{X}}'\hat{\mathbf{X}} + \lambda \mathbf{I_n}\right)^{-\mathbf{1}} \hat{\mathbf{X}}'\mathbf{y}, \tag{2.13}$$

obtained by adding a multiple $\lambda \in \mathbb{R}^+$ of the diagonal matrix I_n to the matrix $\hat{\mathbf{X}}'\hat{\mathbf{X}}$, where $\mathbf{I_n}$ is the identity matrix with $\big(n+1, n+1\big)$ entry set to zero. This solution is called *ridge regression* also known as *Tikhonov Regularization* which is named for Andrey Tikhonov. This type of Regression penalize candidate solutions for using too many features, in order to give preference to a particular solution with desirable properties,

$$L(\mathbf{w}, b) = \lambda \langle \mathbf{w}.\mathbf{w} \rangle + \sum_{i=1}^{l} (\langle \mathbf{w}.\mathbf{x}_i \rangle + b - y_i)^2, \tag{2.14}$$

where the parameter λ controls a trade-off between low square loss and low norm of the sollution.

2.2 Kernel Methods

Linear learning machines have a limited computational power. This point was highlighted in the 1960s by Minsky and Papert [31], but the real world applications demand more expressive hypothesis spaces than the simple linear function.

Kernel methods tackle this problem by offering a suitable kernel function that performs a nonlinear projection to a high dimensional feature space without increasing the tunable parameters. This method raises the computational capability of the linear learning machine [45].

The way of representing the target function defines the function complexity, which will afterwards affect the difficulty of the learning task. The best case is to find a representation that matches the desired learning problem, where the representation of the data in machine learning is considered as a pre-processing step such that [10]:

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n) \longmapsto \phi(\mathbf{x}) = (\phi_1(\mathbf{x}), ..., \phi_n(\mathbf{x})). \tag{2.15}$$

This is equal to mapping the input space X into a new space $F = (\phi(\mathbf{x}) | \mathbf{x} \in \mathbf{X})$. This simple mapping of the data into another space can greatly simplify the task of selecting the best representation of the data.

The space X is defined as the input space, and F is referred to feature space. The quantities used to characterize the data are defined as features, and the original quantities are called attributes. Further, the task of choosing the most suitable representation is defined as the feature selection. A kernel function can be considers as a function that takes its input vectors from the input space X, and returns the dot product of the vectors in the feature space F such as:

$$K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{\Phi}(\mathbf{x}) \cdot \mathbf{\Phi}(\mathbf{z}) \rangle, \ \forall \, \mathbf{x}, \mathbf{z} \in X, \tag{2.16}$$

where Φ is a mapping function from the input space into the feature space , and can be called as kernel feature map. The feature map can also be formulated as any fixed linear transformation $\mathbf{x} \longmapsto \mathbf{A}\mathbf{x}$ for some matrix $\mathbf{A}$, in this case the kernel function is denoted as :

$$\mathbf{k}(\mathbf{x}, \mathbf{z}) = \langle \mathbf{A}\mathbf{x} \cdot \mathbf{A}\mathbf{z} \rangle = \acute{\mathbf{x}}\acute{\mathbf{A}}\mathbf{A}\mathbf{z} = \acute{\mathbf{x}}\mathbf{B}\mathbf{z}, \tag{2.17}$$

where $\mathbf{B} = \acute{\mathbf{A}}\mathbf{A}$ is a squared symmetric positive semi definite matrix.

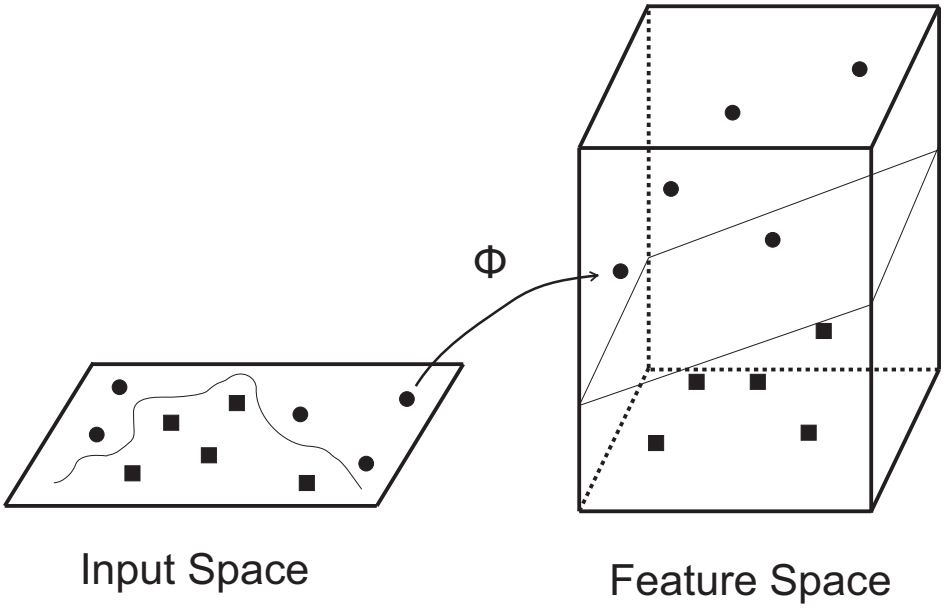

Figure 2.2: Feature Map

Figure 2.2. The function ϕ maps the data from the input space into a higher order feature space where the nonlinear pattern can be shown as linear pattern.

Gram matrix: Let S be a set of vectors $S = \{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n\}$ in an inner product space. The *Gram matrix* is defined [45] as Hermitian matrix of inner products, $n \times n$ matrix $\mathbb{G}$ whose entries are:

$$G_{i,j} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = k(\mathbf{x}_i, \mathbf{x}_j). \tag{2.18}$$

The Gram matrix has the following properties:

- G is Hermitian and positive-semidefinite.
- G is positive-definite if and only if the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$ are linearly-independent.
- $\operatorname{rank} G = \dim \operatorname{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$.

The Gram matrix can be also called as the kernel matrix:

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & ... & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & ... & k(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & k(\mathbf{x}_n, \mathbf{x}_2) & ... & k(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix} \tag{2.19}$$

Kernel Functions types: Kernel learning methods have received a high attention in the last years according to there ability to shift from linearity to non-linearity in a simple and efficient way. Thees properties of kernel function attracts a variety of applications to implement it, which leaded to increase the available types of kernel functions. In 2010 there was stated 25 types of kernel functions [47]. We state in the following paragraph the most two popular kernel functions, which are the polynomial kernel function and the Gaussian kernel function.

polynomial kernel function: The polynomial kernel function is defined as [17] :

$$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\mathsf{T}\mathbf{z} + c)^d, \tag{2.20}$$

where $\mathbf{x}, \mathbf{z}$ are vectors in the input space, d denotes the polynomial degree, $c \geq 0$ is a constant and when $c = 0$, the kernel is said to be homogeneous. An example of using the polynomial kernel function was done in [54] for speaker verification.

Gaussian kernel function: The *Gaussian kernel function* is a widely used in various kernelized learning algorithms and known also as radial basis kernel function or RBF kernel. This type of kernel will be used later in this thesis in chapter four and it can be defined as [13]:

$$K(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|_2^2}{2\sigma^2}\right), \tag{2.21}$$

where $\|\mathbf{x} - \mathbf{z}\|^2$ is defined as the squared euclidean distance between the two feature vectors, and σ is a user defined parameter. Alternatively, The adjustable parameter σ should be carefully selected, according to its high influence on the kernel performance, where, the function will lack regularization if it is underestimated, in the other hand, the exponential will behave almost

linearly in the overestimation case.

Gaussian kernel function can also be defined as [50]:

$$K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|_2^2),$$

where $\gamma = \frac{1}{2\sigma^2}$. There are also kernel functions proposed for particular applications, such as natural language processing [8] and bioinformatics [42].

Cauchy-Schwarz inequality: Let X be an inner product space and $\mathbf{x}, \mathbf{z} \in X$. The Cauchy-Schwarz inequality holds if [45]

$$\begin{aligned} \mathbf{k}(\mathbf{x}, \mathbf{z})^2 &= \langle \mathbf{\Phi}(\mathbf{x}).\mathbf{\Phi}(\mathbf{z}) \rangle^2 \leq \|\mathbf{\Phi}(\mathbf{x})\| \|\mathbf{\Phi}(\mathbf{z})\| \\ &= \langle \mathbf{\Phi}(\mathbf{x}).\mathbf{\Phi}(\mathbf{x}) \rangle \langle \mathbf{\Phi}(\mathbf{z}).\mathbf{\Phi}(\mathbf{z}) \rangle \\ &= \mathbf{k}(\mathbf{x}, \mathbf{x}) \mathbf{k}(\mathbf{z}, \mathbf{z}), \end{aligned} \tag{2.22}$$

where the equality holds if $\mathbf{x}$ and $\mathbf{z}$ are re-scaled from each other, which includes the case $\mathbf{x}$ or $\mathbf{z}$ is the zero vector.

Reproducing Kernel Hilbert Spaces: A Hilbert space $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS) if there exists a kernel function $k : X \text{ x } X \longmapsto R$ if:

- $\forall\, \mathbf{x} \in X, k(\mathbf{x}, \cdot) \in \mathcal{H}$.
- $\forall\, \mathbf{x} \in X,\ \forall f \in H,\ f(\mathbf{x}) = f,\ k(\mathbf{x}, \cdot)$.
  where the second property is called the reproducing property.

Let us formulate and prove some basic properties of reproducing kernels.

1. If a reproducing kernel exists for F, it is unique.
2. If K is a reproducing kernel for F, then for all $\mathbf{x} \in X$ and $f \in F$ there exist
$$|f(x)| \leq \sqrt{K(\mathbf{x}, \mathbf{x})} \|f\|_F.$$
3. If F is a RKHS, then convergence in F implies pointwise convergence of corresponding functions

The Kernel trick essentially is to define kernel function in the original space itself without defining or knowing what the transformation function is, which allow to calculate inner products in the high or infinite-dimensional feature space easily.

2.3 Extension to Multi-Kernel

In the last decade, kernel learning methods have gained a high attention where different types of kernel functions are successfully used in the literature and in various practical applications. Depending on the fact that kernel functions are showing high efficacy in solving learning problems. Multiple kernel learning (MKL) proposes using a set of predefined kernels instead of a single one and learn an optimal linear or non-linear combination of kernels. This method has the ability to combine data generated from different representations possibly from different sources or modalities, which produce different measures of similarity corresponding to different kernels. In such a case, combining kernels is a possible method to combine multiple information origins, which requires building up various kernels. According to, [20] the practical case requires often to base classifiers on combinations of multiple kernels. It has also been shown [20] that using multiple kernels instead of a single one can enhance interpretability of the decision function and improve classifier performance.

The kernel $K(\mathbf{x}, \mathbf{x}')$ in a MKL can be considered as a convex linear combination of other basis kernels [39] and it is givin as:

$$K(\mathbf{x}, \mathbf{x}') = M \sum_{m=1}^{M} d_m K_m(\mathbf{x}, \mathbf{x}'), \quad \text{with } d_m \geq 0 \, , \sum_{m=1}^{M} d_m = 1, \tag{2.23}$$

where M is the total number of kernels.

MKL approaches have been used in various applications, such as object recognition in images [6], event recognition in video [9].

Multiple kernel learning properties: The following paragraph sorts some of the MKL properties.

The Learning Method:

The existing MKL algorithms utilize various learning methods for determining the kernel combination function. These methods can be divide into five major categories:

1. *Fixed rules* are functions without any parameters such as the linear combination algorithm which use rules to set the combination of the kernels like summation and multiplication of the kernels. This approach do not need any training and the weighting is learned in the algorithm.

2. *Heuristic approaches* use a parameterized combination function. The parameters of this function are generally defined by looking at some measure obtained from each kernel function separately or some computation from the kernel matrix:

3. *Optimization approaches* use parametrised kernel combination function and learn the parameters by solving an optimization problem. The optimization Problem can be calculated by obtaining only the combination parameters or integrated to a kernel-based learner.

4. *Bayesian approaches* clarify the kernel combination parameters as random variables, then assign priors on these parameters and learn the parameter values from the priors and the base algorithm.

5. *Boosting approaches* is a MKL algorithm inspired from ensemble and boosting methods which add new kernels iteratively until reaching a stopping threshold when the performance stops improving.

The Functional Form:

The forms of the existing MKL algorithms can be divided into three basic categories, which are the linear combination, the nonlinear combination methods, and the data-dependent combination methods, where the the linear combination methods are the most popular type of combinations used in MKL algorithms.

The Target Function:

We can optimize various target functions when we choose the combination function parameters. we mention an example of target function:

1. *Similarity-based functions* which measure the similarity between the optimal kernel matrix obtained from the training data and the combined kernel matrix. These functions choose also the parameters, which improve this similarity. An approach [22] for measuring the similarity between two matrices is known as kernel alignment (KA), which measures alignment between two kernel functions.

The Training Methods:

MKL algorithms can be divided into two main groups according to their training methodology [14], which are:

1. One-step methods, in such methods, both the combination function parameters and the parameters of the combined base learner can be calculated in a single pass. There are two ways to compute the parameters in these methods, which are the sequential approach and the simultaneous approach. In the sequential approach, the combination function parameters

are computed at the first step, and afterwards a kernel-based learner can be trained by implementing the combined kernel. The simultaneous approach, is able to learn both set of parameters simultaneously.

2. Two-step methods also known as Heckman correction [37] use an alternating approach during each iteration, where first the base learner parameters are fixed and the combination function parameters are updated, and then the combination function parameters are fixed while updating the base learner parameters. These two steps keep alternating until convergence.

The Computational Complexity:

The computational complexity in a MKL algorithms is defined by the base learner computational complexity and its training method. One-step methods are generally faster than Two-step methods but it is also affected by the learning method where using fixed rules and heuristics learning method in most of the cases, do not require a long time to learn the parameters of the combination function. One-step methods, which utilize optimization approaches to learn combination parameters, require higher computational complexity, where these methods are considered as semi-definite programming (SDP) problem.

3 Radio Maps Reconstruction

This chapter gives a general overview on radio maps reconstruction methods. First we present the kriging interpolation method, then we describe the radio frequency tomography method, afterwards we introduce the method of blind channel gain cartography, which is the work that our thesis based on.

3.1 Kriging Interpolation

Kriging, also known as Gaussian process regression, is a spatial interpolation technique for which the interpolated values are modeled by a Gaussian process. It was first introduced in [26] for reconstructing mining maps based on scattered measurements. This technique is based on the methods of geostatistics. Depending on the spatial data, geostatistics can be applied to different geographical domains such as environmental science, meteorology and mining exploration. In fact, the framework of kriging estimation is based on the assumption that there is a relationship between the measured data value at a point in a space and the location of this point, which means that the estimated samples are spatially correlated. The last assumption implies that a suitable correlation model for the data is needed. Kriging Interpolation has been applied in the wireless communication realm to reconstruct different radio propagation features of a map where spatial correlation of the features is assumed. We are focused in this work to the reconstruction of path loss maps, also known as channel gain cartography.

Consider a two-dimensional area $\mathcal{A} \subset \mathbb{R}^2$, the average link gain for an arbitrary point $\mathbf{x}$ with respect to a reference point $\mathbf{x}'$ is given in logarithmic scale as [2–4]:

$$G_r(\mathbf{x}) = G_0 - 10 \cdot \gamma \cdot \log_{10}\|\mathbf{x} - \mathbf{x}'\|_2 + s_r(\mathbf{x}), \tag{3.1}$$

where $\mathbf{x}, \mathbf{x}' \in \mathcal{A}$, $||\cdot||$ denotes the Euclidean norm, G_0 is the path gain at unit distance, $\gamma > 0$ is the path loss exponent, and $s_r(\mathbf{x})$ is the shadow-fading component, which is Gaussian distributed. Note that all the parameters in (3.1) are assumed to be known except for s_r, which is unknown.

In order to compute the estimated shadow fading value $\hat{s}_r(\mathbf{x})$ at arbitrary position $\mathbf{x}'$ via krieging, The authors [26] use a weighted linear combination function of m neighboring known samples as follows:

$$\hat{s}_r(\mathbf{x}) = \sum_{i=1}^{m} w_i(\mathbf{x}) s_r(\mathbf{x}_i), \tag{3.2}$$

where $\sum_{i=1}^{m} w_i(\mathbf{x}) = 1$, $s_r(\mathbf{x}_i)$ being the sample field measurements, and the weights w_i represents the Kriging coefficients of the estimator.

The strength of Kriging interpolator performance lies as the fact that it is a data based interpolator, which means that we have to fit the correlation function to the data before the reconstruction operation. This fact adds a certain complexity to the kriging interpolation algorithms, therefore can be considered as a drawback for Kriging methods. Another drawback of Kriging interpolation is that it does not scale well for big datasets, and the complexity of its standard form makes it improper for online operation.

3.2 Model-Based Radio Frequency Tomography

In order to reconstruct path loss maps, various methods have been proposed in the literature. One of the methods [16] relies on the received signal strength (RSS) measurements from networks. These networks comprise of a set of wireless elements to detect and locate objects through radio frequency (RF) tomography. Tomographic imaging techniques have been deployed in previous works for different applications such as obtaining images of moving objects [53], or tracking motion behind walls [52].

The radio tomography framework has also been used to model the shadow fading. in this application a combination of two functions is used: One function is the spatial loss field (SLF), which measures the attenuation caused of objects placed in the propagation environment. The other function is the weight function, which models the influence by any position of a map to any possible link within the map. Shadow fading is then calculated as the weighted integral of the SLF function in this environment.

The work in [16] aimed to examine the shadow fading model between any transmitter n and receiver m in a given area. For this purpose, they utilized a spatial integral function of the SLF, denoted as $g(\boldsymbol{s})$, which has a domain of every point $\boldsymbol{s}$ in the space weighted by a function $b(\boldsymbol{s}_n, \boldsymbol{s}_m, \boldsymbol{s})$ as follow:

$$\eta_{m,n} = \int g(\boldsymbol{s})\; b(\boldsymbol{s}_n, \boldsymbol{s}_m, \boldsymbol{s})\; d\boldsymbol{s}, \tag{3.3}$$

where $\eta_{m,n}$ is the shadow fading component between the transmitter and receiver, $\boldsymbol{s}_n$ denotes the position of the transmitter, and $\boldsymbol{s}_m$ denotes the position of the receiver.

In practice the SLF function is discretized, which means aggregating the points into pixels and representing the amount of contribution done by each pixel to the total path loss. Further, the weight function should also be discretized. Therefore, $g(\boldsymbol{s})$ can be replaced by a column vector $\mathbf{g}$ and the discrete version of (3.3) is given by:

$$\eta_{m,n} = \boldsymbol{b}_{m,n}\boldsymbol{g}, \tag{3.4}$$

where $\boldsymbol{b}_{m,n}$ denotes the discrete tomographic projection row vector with elements corresponding to each discrete pixel in the field $\boldsymbol{g}$. The equation in (3.4) can be rewritten in a matrix form as:

$$\boldsymbol{\eta} = \boldsymbol{B}\boldsymbol{g}, \tag{3.5}$$

where the vector $\boldsymbol{\eta}$ contains the measurements of $\eta_{m,n}$, and the matrix $\boldsymbol{B}$ consists of the row vectors $\boldsymbol{b}_{m,n}$ for each transceiver pair (m, n). The SLF can be now estimated using the following equation:

$$\hat{\boldsymbol{g}} = \boldsymbol{B}^{-1}\boldsymbol{\eta}. \tag{3.6}$$

However, the inversion of (3.6) is not possible, since the matrix $\boldsymbol{B}$ is under-determined; i.e., there are many more SLF pixels than there are measurements. Therefore the authors utilized Tikhonov regularization as a regularization technique in order to solve this problem. In more details, the problem they aimed to solve is given by:

$$\min_{\hat{\boldsymbol{g}}} ||\boldsymbol{B}\hat{\boldsymbol{g}} - \boldsymbol{\eta}||_2^2 + w\hat{\boldsymbol{g}}^{\boldsymbol{T}}\boldsymbol{C}_{\hat{\boldsymbol{g}}}^{-1}\hat{\boldsymbol{g}}, \tag{3.7}$$

where $\boldsymbol{C_g}$ is the expected covariance matrix of the SLF and w is the regularization weight. The solution of the problem (3.7) can be stated in close form as:

$$\hat{\boldsymbol{g}} = (\boldsymbol{B}^{\boldsymbol{H}}\boldsymbol{B} + w\boldsymbol{C}_{\boldsymbol{g}}^{-1})^{-1}\boldsymbol{B}^{\boldsymbol{H}}\boldsymbol{\eta}, \tag{3.8}$$

However, and in order to obtain a solution to $\hat{\boldsymbol{g}}$, an appropriate model for $\boldsymbol{B}$ has to be chosen. There are many heuristics in the litterateur for this purpose. The authors presented three different heuristics models, namely the Nesh model, the normalized ellipse model, and the inverse area elliptical model. For this thesis, we chose the last model to model the weight function.

The NeSh model is the most common model used in computed tomography (CT), and a modification for RF Tomography was introduced as the Network Shadowing (NeSh) model [1], [33]. In

this model the path loss from shadowing is assumed to be proportional to a line integral across the spatial loss field. The NeSh model differs from traditional tomographic line integral model in that the weights are multiplied by the square root of the path length. This means the tomographic projection is given by the following formula:

$$b(\boldsymbol{s}_n, \boldsymbol{s}_m, \boldsymbol{s}) = \frac{1}{\sqrt{d_{n,m}}} \int_{\boldsymbol{s}_n}^{\boldsymbol{s}_m} \delta(|\boldsymbol{s}_n - \tilde{\boldsymbol{s}}|)\, d\tilde{\boldsymbol{s}} \tag{3.9}$$

where $\delta(x)$ is the Dirac delta function. Using this model, the shadowing component simplifies as:

$$\eta_{m,n} = \frac{1}{\sqrt{d_{n,m}}} \int_{\boldsymbol{s}_n}^{\boldsymbol{s}_m} g(\boldsymbol{s})\, d\boldsymbol{s} \tag{3.10}$$

This modification was made so that the model matches the larger scale shadowing statistical models. It was necessary because this model does not consider the effects of diffraction at all

Figure 3.1 depicts the NeSh model where The arrow corresponds to the line of sight between the transceivers, and the grey pixels correspond to the non-zero values of the discrete weight function.

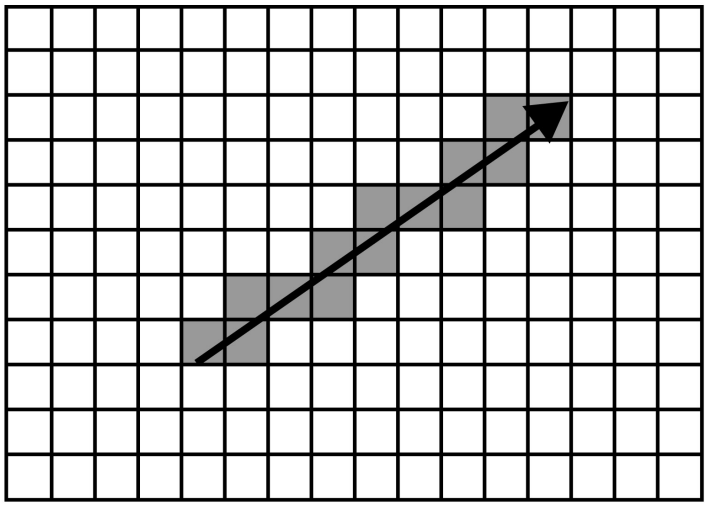

Figure 3.1: NeSh Model

The Normalized Ellipse model contains a selection function based on an ellipse similar to the Fresnel zone, where the transmitter and the receiver define the foci of the ellipse and λ defeins a semi-minor axis length of the ellipse. This model was used in [16, 41, 51–53]. The authors gave this function the value $\frac{1}{d_{n,m}}$ inside the ellipse, and 0 elsewhere :

$$b(\boldsymbol{s}_n, \boldsymbol{s}_m, \boldsymbol{s}) = \begin{cases} \frac{1}{\sqrt{d_{n,m}}} \frac{p_x(\boldsymbol{s}_n, \boldsymbol{s}_m, \boldsymbol{s})^2}{d_{n,m}^2 + \lambda^2} + \frac{p_y(\boldsymbol{s}_n, \boldsymbol{s}_m, \boldsymbol{s})^2}{\lambda^2} & < 1, \\ 0 & \text{otherwise,} \end{cases} \tag{3.11}$$

where $d_{m,n}$ is the distance between the transmitter and the receiver; and $p_x(\boldsymbol{s}_n, \boldsymbol{s}_m, \boldsymbol{s})$ and $p_y(\boldsymbol{s}_n, \boldsymbol{s}_m, \boldsymbol{s})$ are projection functions that project the point onto an axis aligned parallel to the line from $\boldsymbol{s}_n$ and $\boldsymbol{s}_m$, and centered between $\boldsymbol{s}_n$ and $\boldsymbol{s}_m$. The main problem of using this model is the lack of a clear rule for defining the variable lambda, together with the lack of no physical justification for deciding the weights of all the pixels.

Figure 3.2 depicts this model. The arrow corresponds to the line of sight between the transceivers. The ellipse defines the area with influence in the shadow fading between two transceivers placed at the beginning and at the end of the arrow respectively. The width of the ellipse is defined by the wavelength of the signal, and the grey pixels correspond to the non-zero values of the discrete weight function.

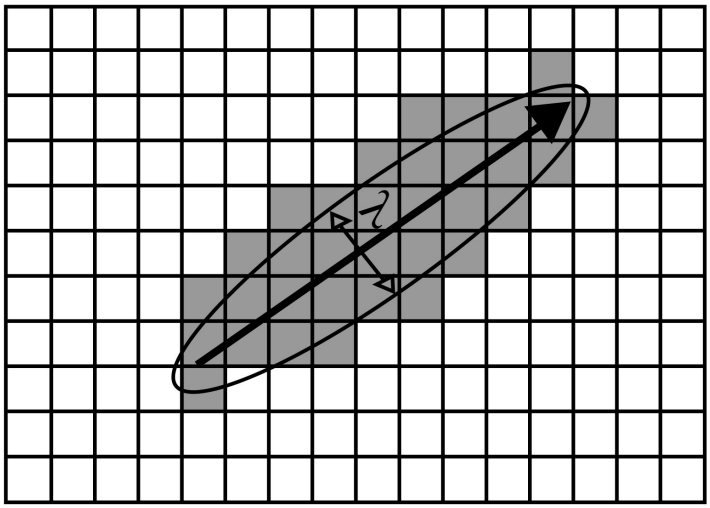

Figure 3.2: Normalized Ellipse Model

The last model presented by the authors is the inverse area elliptical model, the tomographic weight function in this model is equal to the inverse of the area of the smallest ellipse containing $\boldsymbol{s}$ that has the transmitter and receiver as foci. This model leans on the assumption that some parts of the ellipse have a greater contribution than others. Since the part of the signal travelling near the edge of the ellipse travels a longer distance than the part travelling nearer to line of sight, they will be weaker, and contribute less to the total RSS. More specifically, this assumption is presented by weighting points closer to the line of sight more than points located farther from the line of sight (contained in ellipses with larger areas).
Mathematically, this model is represent as:

$$b(\boldsymbol{s}_n, \boldsymbol{s}_m, \boldsymbol{s}) = |\frac{1}{\pi\tilde{\mu}_{n,m}\sqrt{d_{n,m}^2 + \tilde{\mu}_{n,m}^2}}|, \tag{3.12}$$

where the parameter $\tilde{\mu}_{n,m}^2$ is the length of the semi-minor axis of the smallest ellipse containing the point $\boldsymbol{s}$. More specifically,

$$\tilde{\mu}_{n,m}^2 = \arg\min_{\mu} |\frac{p_x(\boldsymbol{s}_n, \boldsymbol{s}_m, \boldsymbol{s})^2}{d_{n,m}^2 + \mu^2} + \frac{p_y(\boldsymbol{s}_n, \boldsymbol{s}_m, \boldsymbol{s})^2}{\mu^2} - 1|. \tag{3.13}$$

Figure 3.3 depicts this model. The arrow corresponds to the line of sight between the transceivers. The multiple ellipses corresponds to different areas with influence in the shadow fading between two transceivers placed at the beginning and at the end of the arrow respectively, this model considers the area of the smallest ellipse that has the transmitter and receiver as foci. The grey pixels correspond to the non-zero values of the discrete weight function.

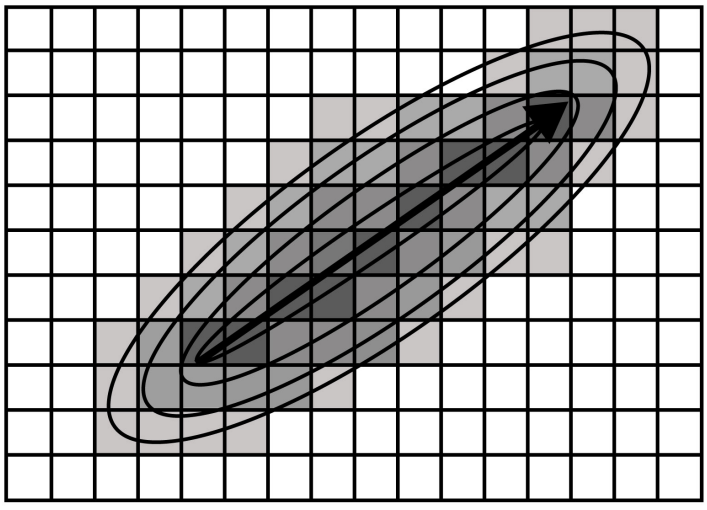

Figure 3.3: Inverse Normalized Ellipse Model

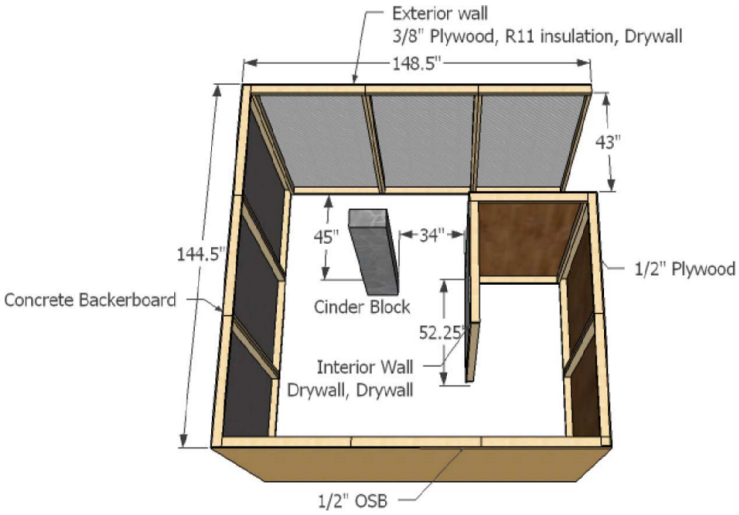

Figure 3.4: Diagram of shadow fading structure [16].

In order to emulate the environment of a practical application of RF tomography, the authors built up an artificial testbed which creates a complex radio propagation environment. The testbed was constructed of plywood, sheetrock concrete board and cinder block. Aiming to provide a more sophisticated shadow fading environment, the structure was designed with interior walls and a cinder block pillar in addition to the exterior walls. This provided a more complex environment which should challenge the shadow fading models tested. The testbed is shown in Figure 3.4.

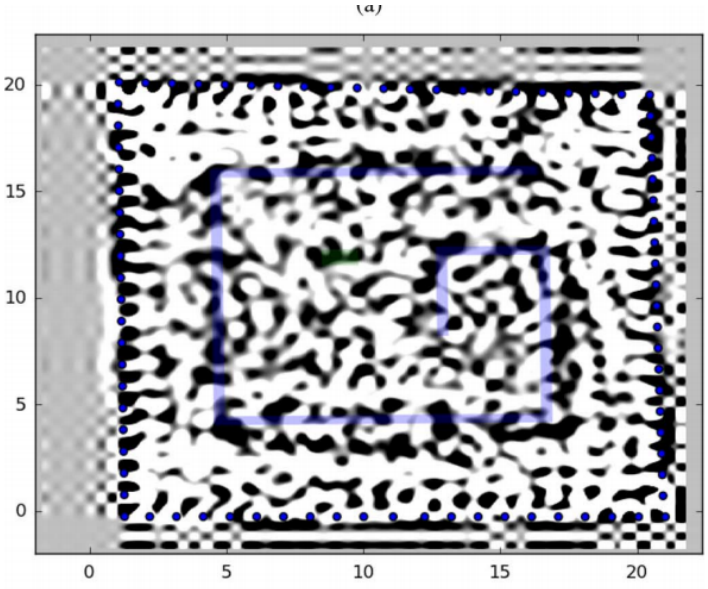

Figure 3.5: Reconstructed results [16].

This main goal in [16] was to reconstruct path loss maps using RF tomography by utilizing different shadow fading models. However, they made strong heuristic assumptions on the structure of the weights function depending on the location of the transmitters and receivers. The results of the reconstructed path loss map showed poor performance as depected in Figure 3.5.

3.3 Blind Channel Gain Cartography

The work presented in the previous section along some other studies [16, 34] rely on the physical concept of the Fresnel zone, and made heuristic assumptions on the structure of the weight function depending on the geographical locations of the transmitters and receivers.
Another work [41] was done afterwards based on the same testbed structure as shown in Figure 3.4. This work was the first to tackle the problem of reconstructing the shadow fading in an environment without depending on the assumption about the weights function structure. The proposed solution in [41] is able to learn simultaneously the SLF and the weight function between arbitrary pairs of transmitter-receiver locations, without depending on any additional side information. It reconstructs then the shadow fading between any two points in a map through a weight integral of the slf, known as tomographic projection. Their work is based on the framework of non-parametric regression in reproducing kernel Hilbert spaces (RKHSs), since as mentioned in the introduction, they are simple to compute and efficient when compared to other non-linear methods. Since the formulation leads to an algorithm that does not scale well with the number of measurements, a clustering technique was proposed to reduce the number of optimization variables. The system model and approach presented in our work builds upon this approach, therefore we will present it in more depth in the next section.

4 A Multi-Kernel Method for Nonparametric Channel Gain Cartography

This chapter is dedicated to describe the theoretical contribution of the thesis. First, we introduce the system model for the linear tomographic projection technique that our work assumes. Then, we state the problem and present the multi-kernel based algorithm to address it. This algorithm is the main contribution of the thesis.

4.1 System Model

Consider a two-dimensional area $\mathcal{A} \subset \mathbb{R}^2$, the power gain between two points $\mathbf{x}, \mathbf{x}' \in \mathcal{A}$ in a map can be given, similar to [2–4, 41], as a logarithmic function with the following form:

$$G(\mathbf{x}, \mathbf{x}') = G_0 - \gamma \log_{10}\|\mathbf{x} - \mathbf{x}'\|_2 - s(\mathbf{x}, \mathbf{x}') + \varepsilon, \tag{4.1}$$

where G_0 represents the gain at unit distance, $\gamma > 0$ is the path loss exponent, $\|\mathbf{x}-\mathbf{x}'\|$ represents the distance between two points $\mathbf{x}, \mathbf{x}'$, $s(\mathbf{x}, \mathbf{x}') : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_+$ denotes the shadow fading between $\mathbf{x}$ and $\mathbf{x}'$, and ε is the measurements noise. Note that s is the only unknown parameter in (4.1), therefore we only need to learn the shadow fading in order to reconstruct the path loss function. In order to model the shadow fading in the map, and in a similar fashion to previous works [16,27,41], we adopted a tomographical model which weights the SLF by a weight function, which in turn models the impact of each position on the link [55]:

$$\tilde{s}(\mathbf{x}, \mathbf{x}') = \int_A w(\phi_1(\mathbf{x}, \mathbf{x}'), \phi_2(\mathbf{x}, \mathbf{x}', \tilde{\mathbf{x}}))f(\tilde{\mathbf{x}})d\tilde{\mathbf{x}}, \tag{4.2}$$

where $w : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ is the weight function; $\phi_1 : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R} : (\mathbf{x}, \mathbf{x}') \mapsto \|\mathbf{x} - \mathbf{x}'\|_2$ denotes the Euclidean distance between two points $\mathbf{x}, \mathbf{x}'$; $\phi_2 : \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R} : (\mathbf{x}, \mathbf{x}', \tilde{\mathbf{x}}) \mapsto \|\mathbf{x} - \tilde{\mathbf{x}}\|_2 + \|\tilde{\mathbf{x}} - \mathbf{x}'\|_2$, denotes the sum of NLOS distances between these two points to a third one $\tilde{\mathbf{x}}$ in the map, and $f : \mathcal{A} \to \mathbb{R}_+$ denotes the SLF. The value of the weight function in RF

tomography [16] depends on the distances ϕ_1 and ϕ_2, which in turn depends on the positions of $(\mathbf{x}, \mathbf{x}', \tilde{\mathbf{x}})$ and the. For practical reasons and similar to [16, 34, 41], a discrete form of 4.1 is used:

$$s(\mathbf{x}, \mathbf{x}') = c \sum_{i=1}^{N_p} w(\phi_1(\mathbf{x}, \mathbf{x}'), \phi_2(\mathbf{x}, \mathbf{x}', \tilde{\mathbf{x}}_i)) f(\tilde{\mathbf{x}}_i) \approx \tilde{s}(\mathbf{x}, \mathbf{x}') \tag{4.3}$$

where N_p denotes the total number of pixels in the map, and $\tilde{\mathbf{x}}_i$ corresponds to the coordinate of the pixel i in the map. where $i \in \{1, ..., N_p\}$. The constant c can be set to one without loss of generality by absorbing any scaling factor into the shadow fading function. After discretizing the shadow fading function, we can now discretize the SLF and the weight function. The discrete SLF vector of a map with N_p pixels can be represented as $\boldsymbol{f} = (f_1, ..., f_n) \in \mathbb{R}^{N_p}$, where $f_i := f(\tilde{x_i})$ and $i = 1, ..., N_p$. Assuming channel reciprocity in the path loss between any two points in a map, we can define the maximum number of links in the map as:

$$L = \binom{N_p}{N_p - 2} = \frac{N_p(N_p - 1)}{2}, \tag{4.4}$$

where L denotes the number of links in a map with N_p pixels.
In order to discretize the weight function, we define the matrix $W \in \mathbb{R}^{L \times N_p}$ containing all possible weight values in the map. The total number of available measurements is given by:

$$N_s = \frac{N(N - 1)}{2}. \tag{4.5}$$

The discrete shadow fading can be considered as a discrete version of 4.2, is calculated as follows:

$$\boldsymbol{s} = W\boldsymbol{f}. \tag{4.6}$$

4.2 Problem Statement

In order to estimate the channel gain in a map, N transceivers obtain measurements at locations $\{\mathbf{x}_1, ..., \mathbf{x}_N\} \subset \mathcal{A}$ with a total number of N_s measurements.
The channel gain between transceivers located at $\mathbf{x}$ and $\mathbf{x}'$ can be obtained by solving (4.1), if the value of $\boldsymbol{s}$ is known. As mentioned previously, in order to estimate the channel gain we need to estimate $\boldsymbol{s}$, where all the parameters in (4.1) are known expect for the $\boldsymbol{s}$. The estimated value of the shadow fading between transceivers can be defined as $\hat{\boldsymbol{s}} = \boldsymbol{s} + \epsilon$, where ϵ denotes the measurement noise. The problem can be defined then using the following formula:

$$\min_{W, \boldsymbol{f}} . \|\hat{\boldsymbol{s}} - W\boldsymbol{f}\|_2^2 + \mu_1 \|\text{vec}(W)\|_2^2, + \mu_2 \|\boldsymbol{f}\|_2^2. \tag{4.7}$$

where μ_1, μ_2 are properly selected regularization parameters.
Unfortunately, simulations have shown that solving (4.7) in this application domain fails to give good results because the problem is highly ill posed. Therefore, to address this challenge, we make use of a multi-kernel method as a non-linear approach.

4.3 Multi-kernel-Based Algorithm

To avoid the severe ill-posedness of the inverse problem described above, we further assume that the weight function can be written as:

$$w(\boldsymbol{\phi}) = \sum_{m \in \mathcal{M}} \sum_{i=1}^{N_s N_p} \alpha_i^m k^m(\boldsymbol{\phi}_i, \boldsymbol{\phi}), \tag{4.8}$$

where $\mathcal{M} = \{1, ..., M\}$ denotes the index set of kernels; α_i^m is the scalar to be determined; k^m is the m_{th} kernel function, and $\boldsymbol{\phi} = [\phi_1, \phi_2]^T \in \mathbb{R}^2$.
In this thesis, we implement the radial basis function, which is defend as:

$$k_{n,n'}^m := k^m(\boldsymbol{\phi}_n, \boldsymbol{\phi}_{n'}) = \exp\big(\frac{\|\boldsymbol{\phi}_1 - \boldsymbol{\phi}_2\|_2^2}{2\sigma_k^2}\big), \tag{4.9}$$

where the value of $\sigma_k^2 > 0$ is a user defined parameter. Choosing the value of σ_k^2 is highly relevant to obtain a good performance. In order to create a multi-kernel system associated with a set of predefined kernels, we fixed each kernel k^m to have a different value of σ_k, which produces a $\mathbf{K_m}$ matrix. This matrix can be defined as:

$$\mathbf{K}_m = \begin{pmatrix} k_{1,1}^m & ... & k_{1,LN_p}^m \\ \vdots & \ddots & \vdots \\ k_{LN_p,1}^m & ... & k_{LN_p,LN_p}^m \end{pmatrix}. \tag{4.10}$$

The full kernel matrix which contains the entries of all m kernels is a diagonal matrix in which every elements of the diagonal is the k_{th} kernel:

$$\mathbf{K} = \begin{pmatrix} K_1 & 0 & ... & 0 \\ 0 & K_2 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & K_M \end{pmatrix}. \tag{4.11}$$

Unfortunately, the number of variables for LN_p increase very fast which makes the problem intractable even for small map size. In order to solve this iisue and similar to [41], we replace

$\boldsymbol{\phi}_{l,i} \in \{1, ..., N_s N_p\}$ with $\boldsymbol{\phi}_c \in \{1, ..., N_c\}$, where $N_c << (N_s N_p)$. The entries of W are then approximated as:

$$w(\boldsymbol{\phi}) \approx \sum_{m \in \mathcal{M}} \sum_{i=1}^{N_c} \alpha_c^m K^m(\boldsymbol{\phi}_c, \boldsymbol{\phi}), \tag{4.12}$$

where $\boldsymbol{\phi}_c$ represents the cluster centroids obtained from applying K-means to $\phi_{l,i} \in \{1, ..., N_s N_p\}$ with N_c clusters. In order to adapt these changes to the algorithm, we replace each $\boldsymbol{\phi}_{(l,i)}$ with $\boldsymbol{\phi}_{r(l,i)}$, where $\boldsymbol{\phi}_{r(l,i)}$ represents the closest element of $\boldsymbol{\phi}_c$ to the original element of $\boldsymbol{\phi}_{(l,i)}$. Mathematically it is described in the following way:

$$r(l,i) := \arg \min_{c \in \{1,...,N_c\}} \|\boldsymbol{\phi}_{l,i} - \boldsymbol{\phi}_c\|_2, \tag{4.13}$$

which produces a many-to-one mapping. We define $K_m \in \mathbb{R}^{N_c \times N_c}$ as the m^{th} kenel matrix, and $\mathbf{R} := [\mathbf{R}_1^T, \ldots, \mathbf{R}_L^T]$, where $\mathbf{R}_l \in \mathbb{R}^{N_s N_p \times N_c}$ is a matrix whose l_{th} column has a one at the centroid position and zeros elsewhere. We can now define $\bar{\alpha}_m = \mathbf{R}\alpha_m$ and approximate K_m as:

$$K_m \approx R \bar{K}_m R^T. \tag{4.14}$$

The full multi-kernel matrix $\bar{K} \in \mathbb{R}^{MN_c \times MN_c}$ id´s given by:

$$\bar{K} = \begin{pmatrix} \bar{K}_1 & 0 & ... & 0 \\ 0 & \bar{K}_2 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & \bar{K}_n \end{pmatrix}. \tag{4.15}$$

The problem we want to solve can be formulated as:

$$\min_{\bar{\boldsymbol{\alpha}}, \boldsymbol{f}} \|\hat{s} - (\mathbf{I}_{MN_c} \otimes \boldsymbol{f}^T) R \bar{K} \bar{\boldsymbol{\alpha}}\|_2^2 + \mu_1 \bar{\boldsymbol{\alpha}}^T \bar{\mathbf{K}} \bar{\boldsymbol{\alpha}} + \mu_2 \|\boldsymbol{f}\|_2^2, \tag{4.16}$$

where $\otimes$ stands for the Kronecker product, and $\mathbf{I}_{MN_c} \in \mathbb{R}^{MN_c \times MN_c}$ is the identity matrix. This equation is separately convex in $\bar{\boldsymbol{\alpha}}$ and $\boldsymbol{f}$. In order to solve a separately convex problem,we use a coordinate block descent method, where we fix one of the variables and solve a sub-problem for this variable, and the resulting solution is then fixed in the next sub problem. This process keeps iterating until reaching a predefined threshold for stopping. The algorithm used to address (4.16) is presented in Algorithm 1.

Consider two sub-problems of (4.16), namely (4.17a) and (4.17b), where we solve for $\boldsymbol{f}$ and $\bar{\boldsymbol{\alpha}}$ when the variables $\boldsymbol{f}$ and $\bar{\boldsymbol{\alpha}}$ are fixed. We can define (4.17a) and (4.17b) as:

$$\min_{\bar{\boldsymbol{\alpha}}} .\|\hat{\boldsymbol{s}} - (\mathbf{I}_{MN_c} \otimes \boldsymbol{f}^T) R \bar{K} \bar{\boldsymbol{\alpha}}\|_2^2 + \mu_1 \bar{\boldsymbol{\alpha}}^T \bar{\mathbf{K}} \bar{\boldsymbol{\alpha}}, \tag{4.17a}$$

$$\min_{\boldsymbol{f}} .\|\hat{\boldsymbol{s}} - (A_{\bar{\boldsymbol{\alpha}}} \boldsymbol{f}) R \bar{K} \bar{\boldsymbol{\alpha}}\|_2^2 + \mu_2 \|\boldsymbol{f}\|_2^2, \tag{4.17b}$$

where $A_{\bar{\boldsymbol{\alpha}}} := \text{diag}(A_{\bar{\boldsymbol{\alpha}}_1}, ...A_{\bar{\boldsymbol{\alpha}}_M}), A_{\bar{\boldsymbol{\alpha}}_m} = \sum_{i=1}^{N_s} \boldsymbol{e}_i \otimes (\bar{\boldsymbol{\alpha}}_m^T \bar{K}_m R_i^T)$ and $\boldsymbol{e}_i$ is the unitary vector with all zeros but ith entry one. The solution of 4.17a is given by:

$\bar{\boldsymbol{\alpha}} = [A_f A_f^T + \mu_2 M N_c \bar{K}]^{-1} A_f \hat{\boldsymbol{s}}$

In a similar way, the solution of 4.17b is given by:

$\boldsymbol{f} = [A_\alpha^T[k] A_\alpha[k] + \mu_1 N_p I_{N_p}]^{-1} A_\alpha^T \hat{\boldsymbol{s}}$

Algorithm 1 Multi-kernel algorithm to solve 4.16

Input : $\mu_1, \mu_2, \bar{\boldsymbol{\alpha}}[0], \boldsymbol{f}[0], \hat{\boldsymbol{s}}, K, N_c, M,\ R, maxIteration, \varepsilon$

Input Variables : converged $\leftarrow False$, $\boldsymbol{f}[0] \leftarrow random$

while $NOT\, converged\ AND\ i < maxIteration$ **do**

- $\bar{\boldsymbol{\alpha}}$ updates:
- $A_f \leftarrow \bar{K}^T R^T (\mathbf{I}_{MN_c} \otimes \boldsymbol{f}[k])$
- $\bar{\boldsymbol{\alpha}}[k+1] \leftarrow [A_f A_f^T + \mu_2 M N_c \bar{K}]^{-1} A_f \hat{\boldsymbol{s}}$
- $\boldsymbol{f}$ updates:
- $A_\alpha \leftarrow \sum_{i=1}^{MN_c N_p} \boldsymbol{e}_i \otimes (\bar{\boldsymbol{\alpha}}^T][k] K_i)^T$
- $\boldsymbol{f}[k+1] \leftarrow [A_\alpha^T[k] A_\alpha[k] + \mu_1 N_p I_{N_p}]^{-1} A_\alpha^T \hat{\boldsymbol{s}}$
- Check end of while loop
- $k \leftarrow k+1$
- **if** $|\boldsymbol{f}^k - \boldsymbol{f}^{k+1}| < \varepsilon\ OR\ i = maxIteration$ **then**
  - $converged \leftarrow True$
- **end**

end

Output: $\bar{\boldsymbol{\alpha}}^{\boldsymbol{k}}$, $\boldsymbol{f}^k$

The complexity of the algorithm is controlled by the two matrix inversions required in each

iteration.The matrix dimension for the $\bar{\boldsymbol{\alpha}}$-iteration is $Nc \times Nc$, while for the $\boldsymbol{f}$-iteration the dimensionality is $N^2 \times N^2$ Accordingly, the number of operations is in $\mathcal{O}(N_c^3 + (N^2)^3)$.

5 Numerical Evaluation

This chapter gives a numerical evaluation of the non-parametric channel gain reconstruction presented in the previous chapter. It presents also a performance comparison between our proposed algorithm and the algorithm presented in [41] and briefely described in section 3.3 as baseline in different propagation environments. The different between their algorithm and ours is the use of multiple kernels instead of single one.

Both proposed algorithms were tested with synthetic data to evaluate their performance. In order to make a fair comparison between our work and the baseline, we used the same scenario as in their work. Figure 5.1 shows the original SLF map of the scenario. After the SLF

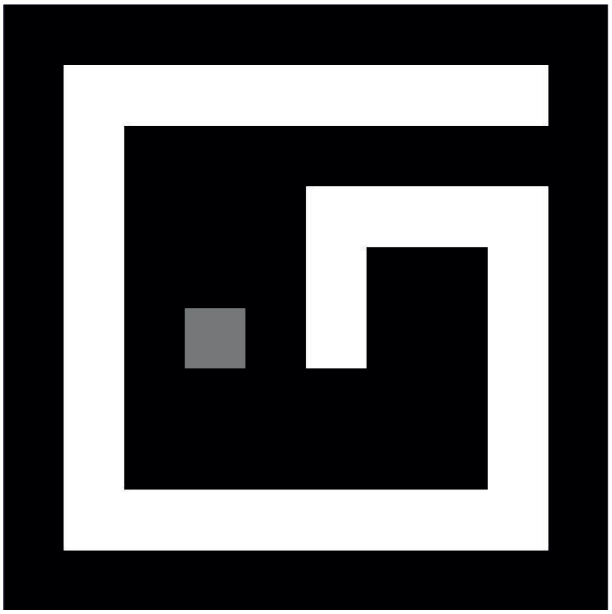

Figure 5.1: SLF map of 10 × 10 room

was normalized, high values (white) represent objects with a high absorption index, while zeros (black) represent free space regions with low absorption index. To generate a synthetic weight function, we used the inverse area elliptical model. The room was discretized into a 10 × 10 map and $N = 80$ transceivers were randomly placed. The total number of links in the map is

$L = \frac{N_p\left(N_p-1\right)}{2} = 4950$, from which we acquired $N_s = 80(80-1)/2 = 3160$ samples, corresponing to 63.8% of all possible measurements.

We assume that most values of the weight matrix W are zeros, and the non-zero values are gathered around the line of sight of each link. We also assume that most entries of the SLF vector $\boldsymbol{f}$ are zeros, and the non-zero values are grouped together. We made these assumptions because most of the map pixels represent the free space whose absorption value is negligible compared to the absorption of solid objects, while non-zero entries of $\boldsymbol{f}$ represent walls and other of solid objects in the map, therefore gathered in groups.

In order to model the weight function we used inverse area ellipse model [16], which is defined as:

$$w(\phi_1, \phi_2) := \begin{cases} 0, & \text{if } \phi_2 > \phi_1 + \frac{\lambda}{2} \\ \min(\Omega(\phi_1, \phi_2), \Omega(\phi_1, \phi_1 + \delta)) & \text{otherwise,} \end{cases} \tag{5.1}$$

where λ denotes the wavelength of the transmitted signal; $\delta > 0$, and

$$\Omega(\phi_1, \phi_2) := \frac{4}{\pi\phi_2\sqrt{\phi_2^2 - \phi_1^2}}. \tag{5.2}$$

This model was previously depicted in Figure 3.3 and also used in [53].

The number of kernels was varied between two up to four kernels. Increasing the number of kernels to more than four added complexity to the problem, but as results show the performance was limited between four and tow. Therefore we fixed the maximum number of kernels to four.

The best regularization parameters were acquired after performing independent grid-searches for our algorithm since the optimal regularization parameters differ for different number of kernels. Table 5.1 presents the main simulation parameters.

Table 5.1: Simulation parameters

Parameter	Value	Description
N_p	10×10	Map size
L	4950	Number of links
N	80	Number of transceivers
N_s	2000	Number of measurrements
N_c	1500	Number of clusters
M	1, 2, 3, 4	Number of kernels
σ_1	0.12	First kernel width
σ_2	0.13	Second kernel width
σ_3	0.14	Third kernel width
σ_4	0.15	Fourth kernel width
μ_1	5×10^{-04}	Regularization parameter of W
μ_2	6×10^{-04}	Regularization parameter of $\boldsymbol{f}$
λ	0,35	Signal wavelength
SNR	∞	Signal to noise ratio
ε	10^{-05}	Difference threshold for $\lvert \boldsymbol{f}^k - \boldsymbol{f}^{k+1} \rvert$

We asses the performance of our algorithm and the baseline algorithm in different scenarios. First of all, we created a noisy propagation environment, then we changed the wavelength of the transmitted signal, afterward and to finalize, we changed the map and examined the performance for the new map while keeping regularization parameters of Table 5.1.

5.1 Noisy Conditions

In this scenario, we change the noise level in the propagation environment in order to add more complexity to the environment and to assess the ability of our algorithm to reconstruct the path loss map in such noisy environment. We implement the same environment to the baseline algorithm in order to make a fair comparison between them. Figure 5.2 depicts the performance under noisy conditions of both approaches. More concretely, the normalized mean square error

(NMSE) is used as performance metric and calculated as:

$$\frac{||Y - \hat{Y}||_2^2}{||Y||_2^2}, \tag{5.3}$$

where Y is any vector and $\hat{Y}$ is its reconstruction. The SNR levels comprise [0, 10, 20, 30, 40] dB. All other algorithm parameters are kept as in Table 5.1.

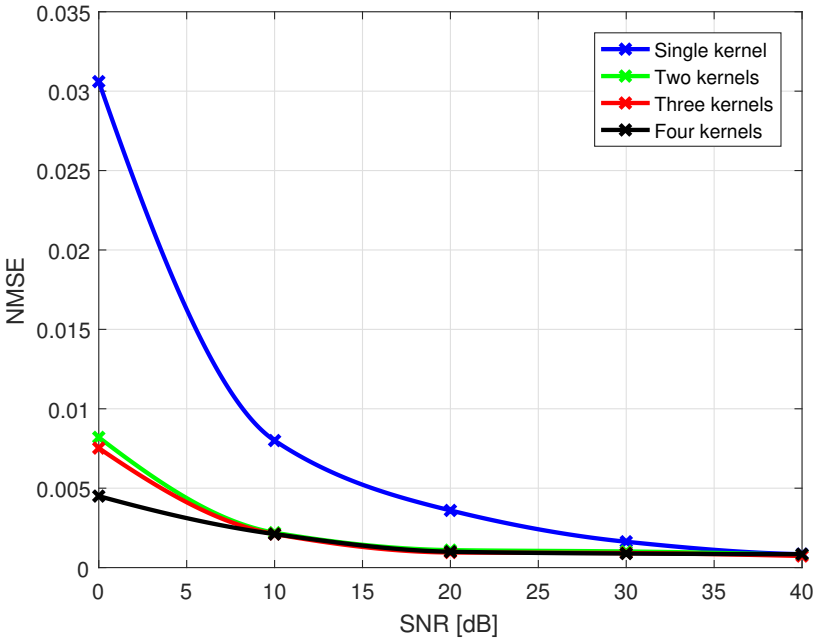

Figure 5.2: NMSE vs SNR for the SLF of single kernel (blue), two kernels (green),three kernels (red), and four kernels(black).

We can observe from Figure 5.2, that the performance of the single kernel method sufferers from high degradation when the noise level increases. The baseline algorithm measured a NMSE of the SLF at 0 dB of 0.0306. In fact, multi-kernel method shows much better performance at the same noise level. Using two kernels reduced the NMSE to 0.0082, which corresponds to accuracy improvement of over 70%. Moreover, using four kernels increased the accuracy to over 85%. The values of NMSE for 0 dB noise level are shown in Table 5.2.

Table 5.2: NMSE values for 0 dB SNR

Paremeter	M=1	M=2	Improvement	M=3	Improvement	M=4	Improvement
NMSE SLF	0.0306	0.0082	73.2%	0.0075	75.49%	0.0045	85.29%

The mean value of NMSE of the shadowing and the weight function for all levels are shown in Table 5.3.

Table 5.3: Mean NMSE values for all SNR levels

Paremeter	M=1	M=2	Improvement	M=3	Improvement	M=4	Improvement
Shadowing	0.068	0.053	22.06%	0.048	29.41%	0.042	38.42%
Weight Function	0.22	0.19	13.64%	0.15	31.82%	0.12	45.45%

In order to illustrate the performance of reconstructing the SLF, we depict in Figure 5.3 the reconstructed SLF of the baseline algorithm when the noise level is 0 dB. Figures 5.4, 5.5 and 5.6 show the reconstructed SLF of multi-kernel approach in the same environment for M=2,3 and 4 ,respectively. We can visually determine that the the baseline algorithm fails to reconstruct the SLF for this noise level, but we can notice improvement of the performance when increasing the number of kenels.

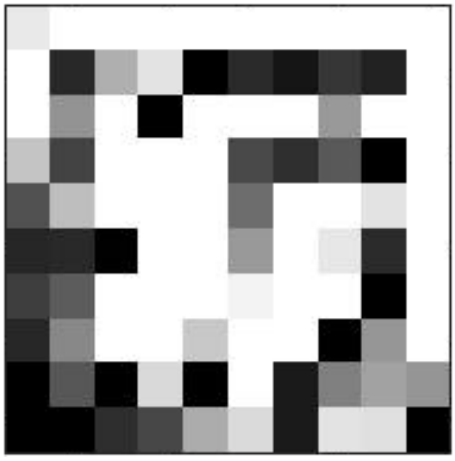

Figure 5.3: Reconstructed SLF using a single kernel

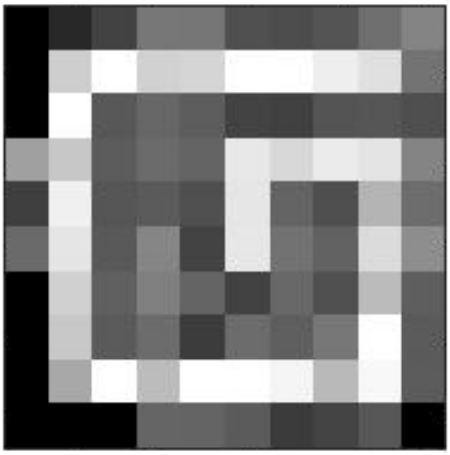

Figure 5.6: Reconstructed SLF using four kernels

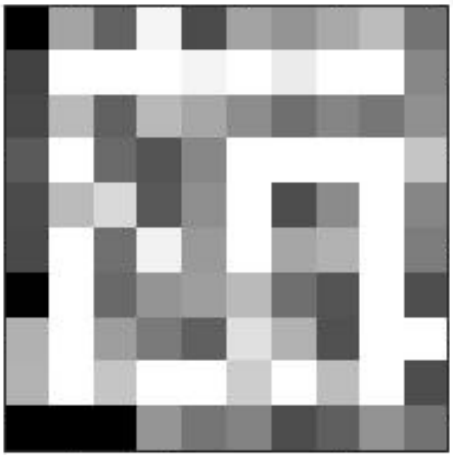

Figure 5.4: Reconstructed SLF using two kernels

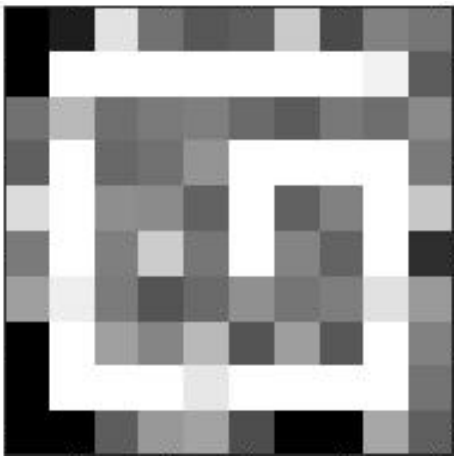

Figure 5.5: Reconstructed SLF using three kernels

5.2 Frequency variation

In this scenario, we change the wavelength of the transmitted signal λ. As mentioned before, we use the inverse area elliptical model to model the weight function. In this model, λ presents the width of the ellipse containing the transmitter and receiver as foci. We did these changes, in order to examine the ability of a multi-kernel method to reconstruct the path loss map while varying the width of the ellipse. In fact increasing the wavelength will increase the width of the ellipse, which will consequently increase the number of non-zero values of a weight function are considered and the problem becomes more complex. For each kernel, we fixed lambda to ten different values. The values of lambda comprise [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1] meters. All other parameters are kept as in table 5.1. We implement the same changes to the baseline algorithm while fixing all the parameters in order to make a fair comparison. Figure 5.7 depicts the performance of our algorithm and the base line. The performance metric is again NMSE

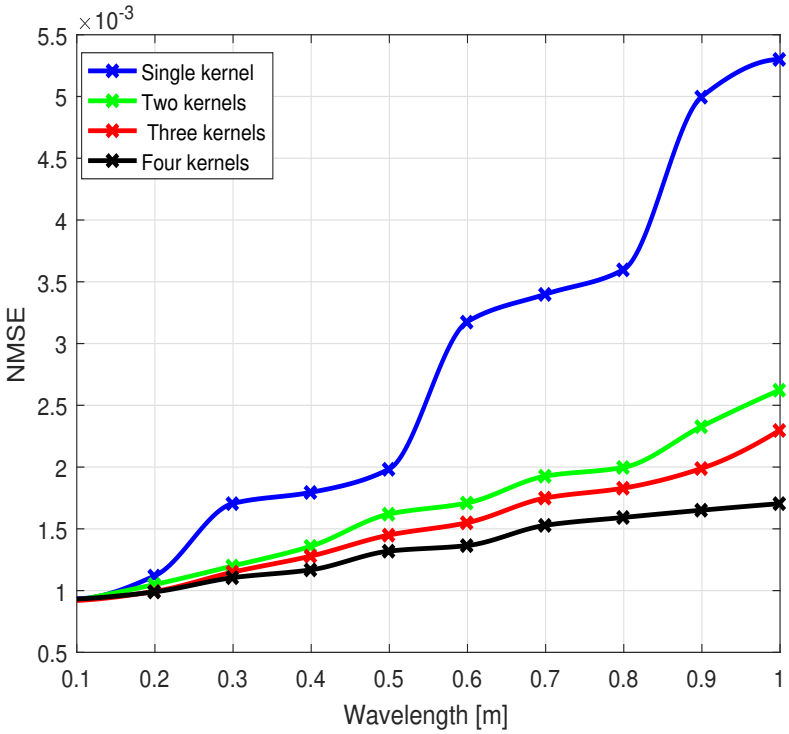

Figure 5.7: NMSE vs λ for the SLF of single kernel (blue), two kernels (green),three kernels (red), and four kernels(black).

We can clearely see from Figure 5.7, that the performance of the single kernel method sufferers from high degradation when increasing λ. This is due to the fact that increasing λ will increase the area in which the non-zero values of the weight function are considered, which results in de-

creasing the reconstruction performance of the baseline algorithm. We can also remark that our algorithm increases the considerably estimation accuracy from 0.3 m on. The baseline algorithm has measured a NMSE value of the SLF of 17×10^{-3} at lambda equals to 0.3 m. Our algorithm measured NMSE value of the SLF of 12×10^{-3} for the same lambda value, while using two kernels, which corresponds to performance improvement of over 29%.

The greatest difference in the performance of our algorithm and the baseline algorithm can be noticed at lambda value of 1 m, at which the width of the ellipse is at the largest value of both approaches. The baseline algorithm obtais a NMSE value for the SLF of 53×10^{-3}, while our algorithm gets a NMSE value of 12×10^{-3} whith using two kernels, which corresponds to accuracy improvement of over 50%. Moreover, using four kernels made a Precision improvement of over 67%. Table 5.4 shows the NMSE of SLF values for wavelength equals to 1 m.

Table 5.4: NMSE values for 1 m wavelength

Paremeter	M=1	M=2	Improvement	M=3	Improvement	M=4	Improvement
NMSE SLF	0.0053	0.0026	50.47%	0.0023	56.6%	0.0017	67.92%

The mean value of NMSE of the shadowing and the weight function for all wavelength values are shown in Table 5.5.

Table 5.5: Mean NMSE for all wavelength values

Paremeter	M=1	M=2	Improvement	M=3	Improvement	M=4	Improvement
Shadowing	0.096	0.071	26.04%	0.062	35.42%	0.058	39.58%
Weight function	0.58	0.46	20.69%	0.38	34.48%	0.31	46.55%

5.3 Different map layouts

In this scenario we change the map which corresponds to the artificial testbed introduced by [16] and shown in Figure 3.4. We replace it with a new map in order to asses the performance of our algorithm in a different environment. All other algorithm parameters are kept as in table 5.1. Furthermore, we implement the same environment to the baseline algorithm in order to make a fair comparison and depict the estimated SLF for each number of kernels. Figure 5.8 shows the original SLF map of the scenario. Figure 5.9 depicts the reconstructed SLF of. Figures 5.10, 5.11 and 5.12 show the reconstructed SLF of multi-kernel approach in the same environment for M=2,3 and 4 ,respectively.

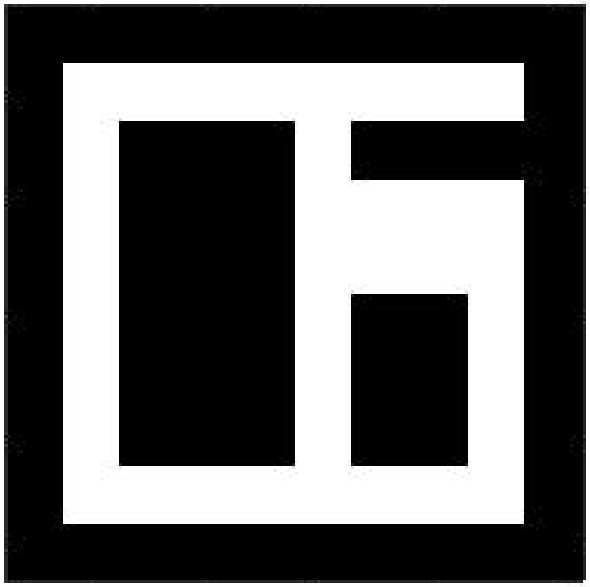

Figure 5.8: New SLF map of 10 × 10 room

The Table 5.6 presents a numerical evaluation of the SLF NMSE for estimating the new map.

Table 5.6: NMSE values for the new map

Paremeter	M=1	M=2	Improvement	M=3	Improvement	M=4	Improvement
NMSE SLF	0.0021	0.00181	16.02%	0.00174	20.69%	0.00166	26.51%

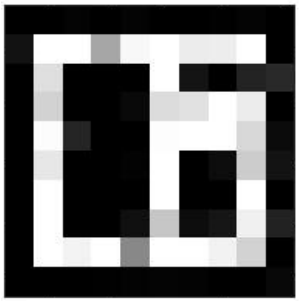

Figure 5.9: Reconstructed SLF using a single kernel

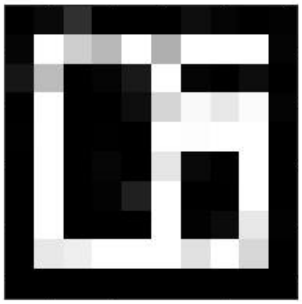

Figure 5.12: Reconstructed SLF using four kernels

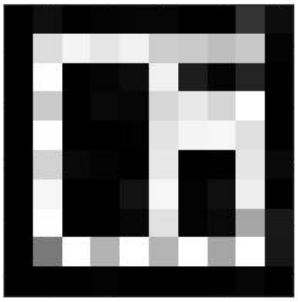

Figure 5.10: Reconstructed SLF using two kernels

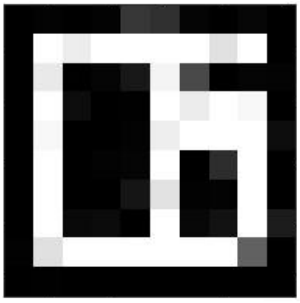

Figure 5.11: Reconstructed SLF using three kernels

6 Conclusion and Outlook

The aim of this thesis was to reconstruct channel gain maps in wireless networks. To this end, we modeled the shadow fading with a linear tomographic projection technique. We estimated the shadow fading between two points as the integral of the SLF and a weight function that models the influence of each position of a map to a link. In order to model the weight function, we used the inverse area elliptical model. The original problem is highly ill-posed. Therefore, we propose and evaluate an algorithm, which adds some structure to the problem by approximating the weight function with non-linear kernels. A nonparametric regression in RKHSs was applied to learn both the SLF and the weight function, and then reconstruct the shadowing attenuation between any two points in a map. To assess the performance of the proposed algorithm, we evaluated a 10×10 map for different scenarios, and we compared the results with the approach in [41]. First, we increased the noise in order to assess the performance in noisy environments. Then, we increased the number of non-zero values considered by the window function by increasing the wavelength, which consequently adds more complexity to the problem. Finally, we changed the room layout in order to assess the robustness of the algorithm when the scenario topology changes. All the previously mentioned scenarios were tested with to the baseline algorithm. Results show performance improvements for any scenario when a multi-kernel approach is used.

Outlook and future research:

Reconstructing path loss maps using multi-kernel approach is a really interesting and promising field of study. For the future work, we would like to extend the research and consider reducing the assumptions on the weight function. More concretely, including all the pixels outside the Fresnel zone. Furthermore, we would like to make use of different forms of regularization such as l_1. Another interesting field of study is by considering online approaches. Furthermore, we can extend our simulations by making changes such as creating our own tested to see the impact of using different materials on that shadowing of the signal. We are also interested in increasing the number of kernels used for this approach to more than four, since increasing the number of kernels showed a clear performance improvement.

Bibliography

[1] Agrawal and N. Patwari. Correlated link shadow fading in multi-hop wireless networks. *IEEE Trans. Wireless Commun*, 8,(8), 2009.

[2] A.konak. Estimating path loss in wireless local area networks using ordinary kriging. *proceedings of the winter simulation conference*, pages 2888–2896, 2010.

[3] A.konak. Predicting coverage in wireless local area networks with obstacles using kriging and neural neteorks. *International journal of mobile networks design and innovation*, 3(4):224–230, 2011.

[4] A. B. H. Alaya-feki, S. B.jemaa, B.sayrac, P. Houze, and E.Moulines. Informed spectrum usagein cognitive radio network: Interference cartography. *2008 IEEE 19th International symposium on Personal, Indoor and mobile Radio Communication,*, pages 1–5, 2008.

[5] Maria-Florina Balcan and Avrim Blum. On a theory of learning with similarity functions. 2006.

[6] Serhat S. Bucak, Rong Jin, and Anil K. Jain. Multiple kernel learning for visual object recognition: A review. 2014.

[7] C. Carmeli, E. De Vito, A. Toigo, and V. Umanita. Vector valued reproducing kernel hilbert spaces and universality. 8:19–61, 2010.

[8] Yin-Wen Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard, and Chih-Jen Lin. Training and testing low-degree polynomial data mappings via linear svm. 2010.

[9] Lin Chen, Lixin Duan, and Dong Xu. Event recognition in videos by learning from heterogeneous web sources. 2013.

[10] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[11] Marc G. Genton. Classes of kernels for machine learning: A statistics perspective. *Machine Learning Research*, 2:299–312, 2001.

[12] W. Norton Grubb and Paul Ryan. *The roles of evaluation for vocational education and training.* International Labor Office, October 1999.

[13] Steve R. Gunn. Support vector machines for classification and regression. 1998.

[14] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *Machine Learning Research*, 12:2211–2268, 2011.

[15] B. R. Hamilton, R. J. Baxley X. Ma, and S. M. Matechik. Radio frequency tomography in mobile networks. *Proc. IEEE Statist. Signal Process. Workshop*, page 508–511, 2012.

[16] Benjamin R. Hamilton, XiaoliMa, Robert J. Baxley, and Stephen M. Matechik. Propagation modeling for radio frequency tomography in wireless networks. *IEEE Journal of selected topics in signal processing*, 8(1), Feb 2014.

[17] Thomas Hofmann, Bernhard Scholkopf, and Alexander J. Smola. A review of kernel methods in machine learning. 2006.

[18] Thomas Hofmann, Bernhard Scholkopf, and Alexander J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36:1171–1220, 2008.

[19] Hal Daum´e III. Similarity-based multiple kernel learning algorithms for classification of remotely sensed images.

[20] Michael I. Jordan, Francis R. Bach, and Gert R. G. Lanckriet. Multiple kernel learning, conic duality, and the smo algorithm. 2004.

[21] Yuri Kalnishkan. An introduction to kernel methods. Technical report, University of London,Department of Computer Science, May 2009.

[22] Jaz Kandola, John Shawe-Taylor, and Nello Cristianini. Optimizing kernel alignment over combinations of kernels.

[23] Martin Kasparick, Renato L. G. Cavalcante, Stefan Valentina, Sławomir Stanczak, and Masahiro Yukawa. Kernel-based adaptive online reconstruction of coverage maps with side information. *IEEE Transactions on Vehicular Technology*, . 65,,(7), Jul 2016.

[24] Abdullah Konak. Estimating path loss in wireless local area networks using ordinary kriging. 2010.

[25] Risi Kondor. A short introduction to hilbert space methods in machine learning. October 2003.

[26] D.G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Chemistry, Metal and Mining Society of South Africa*, 52:119–139, 1951.

[27] Donghoon Lee and Seung-Jun Kim. Channel gain cartography via low rank and sparsity. 2014.

[28] Shutao Li, Chenglin Liu, and Yaonan Wang. *Pattern Recognition*. Springer International Publishing AG., 2014.

[29] David G. Luenberger. *Optimization By Vector Space Methods*. John Wiley Sons, Inc, 1969.

[30] Ha Quang Minh1, Partha Niyogi1, and Yuan Yao2. Mercer's theorem, feature maps, and smoothing. 2006.

[31] M. Minsky and S. Papert. Perceptrons. *An Introduction to Computational Geometry*. MIT Press, 1969.

[32] L. L. Monte, L. K. Patton, and M. C. Wicks. Direct-path mitigation for underground imaging in rf tomography. *Electromagnetics in Advanced Applications (ICEAA)*, 2010.

[33] N. Patwari and P. Agrawal. Nesh: A joint shadowing model for links in a multi-hop network. *Proc. IEEE ICASSP*, page 2873–2876, 2008.

[34] Neal Patwari and Piyush Agrawal. Effects of correlated shadowing: Connectivity, localization, and rf tomography. May 2008.

[35] Caleb Phillips, Douglas Sicker, and Dirk Grunwald. A survey of wireless path loss prediction and coverage mapping methods. *IEEE Communications Surveys Tutorials*, . 15,(1), 2013.

[36] Natesh S. Pillai, Qiang wu, Feng Liang, Sayan Mukherjee, and Robert L. Wolpert. Characterizing the function space for bayesian kernel models. *Machine Learning Research*, 8:1769–1797, 2007.

[37] JPatrick A. Puhanii. The heckman correction for sample selection and its critique. 2000.

[38] Alain Rakotomamonjy, Francis Bach, Stephane Canu, and Yves Grandvalet. More efficiency in multiple kernel learning. 2007.

[39] Alain Rakotomamonjy, Francis R. Bach, Stephane Canu, and Yves Grandvalet. Simple mkl. *Machine Learning Research*, 9:2491–2521, 2008.

[40] Steven Roman. *Advanced Linear Algebra, Graduate Texts in Mathematics*. Springer-Verlag, 2005.

[41] Daniel Romero, Donghoon Lee, and Georgios B. Giannakis. Blind channel gain cartography. *IEEE Journal of selected topics in signal processing*, 2016.

[42] Behrnard Schölkopf, Koji Tsuda, and Jean-Philippe Vert. *Kernel Methods in Computational Biology*. MIT Press Cambredge, 2004.

[43] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels Support Vector Machines, Regulation, Optimization, and Beyond*. MIT Press, 2001.

[44] Dino Sejdinovic and Arthur Gretton. What is an rkhs? March 2014.

[45] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[46] De shuang huang, Kang Hyun JO, and Ling Wang. *Intelligent Computing Methodologies*. springer, Aug 2014.

[47] César Souza. Kernel functions for machine learning applications, 2010.

[48] Henry Stark and Yongyi Yang. *Vector Space Projections A Numerical Approach to Signal and Image Processing, Neural Nets, and Optics*. John Wiley Sons, Inc, 1998.

[49] Ryan Tibshirani and Larry Wasserman. Nonparametric regression. 2015.

[50] Jean-Philippe Vert, Koji Tsuda, and Bernard Schölkopf. A primer on kernel methods. 1992.

[51] J. Wilson and N. Patwari. Regularization methods for radio tomographic imaging. *Proc. Virginia Tech Wireless Symp.*, Jun 2009.

[52] Joey Wilson and Neal Patwari. See-through-walls motion tracking using variance-based radio tomography networks. 2009.

[53] Joey Wilson and Neal Patwari. Radio tomographic imaging with wireless networks. *IEEE transactions on mobile computing*, 9(5), May 2010.

[54] Sibel Yaman and Jason Pelecanos. Using polynomial kernel support vector machines for speaker verification. *IEEE Signal Processing Letters*, (9), September 2013.

[55] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic networks. 2005.