Channel-Gain Cartography via Mixture of Experts

Luis M. Lopez-Ramos, *Member, IEEE,* Yves Teganya, *Student Member, IEEE,*
Baltasar Beferull-Lozano, *Senior Member, IEEE,* and Seung-Jun Kim, *Senior Member, IEEE*

***Abstract*—In order to estimate the channel gain (CG) between the locations of an arbitrary transceiver pair across a geographic area of interest, CG maps can be constructed from spatially distributed sensor measurements. Most approaches to build such spectrum maps are *location-based*, meaning that the input variable to the estimating function is a pair of spatial locations. The performance of such maps depends critically on the ability of the sensors to determine their positions, which may be drastically impaired if the positioning pilot signals are affected by multi-path channels. An alternative *location-free* approach was recently proposed for spectrum power maps, where the input variable to the maps consists of features extracted from positioning signals, instead of location estimates. The location-based and the location-free approaches have complementary merits. In this work, apart from adapting the location-free features for the CG maps, a method that can combine both approaches is proposed in a mixture-of-experts framework.**

I. Introduction

Information regarding the channel gain (CG) between a pair of wireless transceivers is critical in a plethora of resource allocation (RA) algorithms. In the context of device-to-device (D2D) communication and cognitive radios, judiciously designed RA algorithms can boost the network performance metrics significantly [1]. For instance, consider a cellular system where regular cellular connections coexist with D2D communication. A RA scheme can be implemented for channel assignment and/or power allocation across the pairs of D2D devices, with the goal of maximizing the total aggregated throughput or other relevant metrics. Such a RA scheme will typically require estimates of the CGs between the cellular users and the D2D users, and between the transmitter and the receiver in each of the D2D links, in order to quantify the expected interference caused from sharing the communication channels. Given the difficulty of continuously measuring the CG between arbitrary pairs of devices, the approach based on the CG map is very useful [2], [3]. The RA performance depends heavily on the accuracy of the CG estimates; therefore, the accuracy of the map is critical. Moreover, in the case of mobile networks, the CGs for the spatial locations where the transceivers are expected to be in the future time slots are also necessary. Thus, it is important to have CG estimates not only where the transceivers are currently located, but also in the arbitrary locations around them.

This work was supported by grant(s) FRIPRO TOPPFORSK WISECART grant 250910/F20 and SFI Offshore Mechatronics 237896/E30, from the Research Council of Norway, as well as by US NSF grant 1547347.

The first three authors are with the WISENET Center, Dept. of ICT, University of Agder, Jon Lilletunsvei 3, Grimstad, 4879 Norway. E-mails:{luismiguel.lopez, yves.teganya, baltasar.beferull}@uia.no. Seung-Jun Kim is with the Dept. of Comput. Sci. Electr. Engr., University of Maryland, Baltimore County, Baltimore, MD 21250, USA. E-mail: sjkim@umbc.edu.

In its simplest form, a CG map can be defined as a function that maps a pair of spatial locations to an estimate of the CG between them. The CG maps were estimated by recovering the so-called spatial loss field [4], where the gain was modeled as a distance-based loss plus a weighted integral of the spatial loss field. This model accounts for the loss due to the absorption from obstacles, but can be inaccurate in the multi-path propagation environments where the signals may get severely attenuated, or amplified by multiple reflections.

The data samples used to train the CG function are taken from a set of pairs of sensing nodes spread across the area to be covered by the map. Thus, the existing methods typically rely on accurate sensor location estimates [2], [3]. However, under practical conditions (for instance, when the localization is achieved using positioning pilots sent by base stations), the sensing nodes might not be able to determine their locations accurately. Specifically, the time (difference) of arrival (ToA/TDoA) measurements may experience severe bias due to non-line-of-sight (NLoS) conditions [5], with large localization error as a consequence. The training data is thus noisy in the input variable, which then translates into errors in the map estimates. The error due to the NLoS propagation in strong multi-path environments may even destroy many location estimates, rendering the CG map uninformative.

Recently, in the simpler case of an interference power map, which is a function of a single location, this issue was mitigated by extracting features from the pilot signals directly, and building the maps using kernel ridge regression (KRR) [6]. Such features have nature similar (but not exactly equal) to the positioning features such as the ToA or TDoA. Following [6], here the (standard) procedure of estimating the location of the sensors from the available pilot signals, and training the map as a function of a vector of spatial coordinates, is referred to as the "location-based" (LocB) cartography. On the other hand, the (novel) method of directly learning the map estimate as a function of the pilot signal features is referred to as the "location-free" (LocF) cartography.

While many works explored the use of advanced regression techniques (such as those based on deep learning [7], [8] or advanced kernel methods [9]) to improve the accuracy of spectrum maps, most of these works build LocB maps. When the estimation of the sensor locations is accurate, the maps produced by these methods are also accurate. Consequently, these methods rely on pilot signals from base stations only when the LoS component is strong, but not in strong multi-path environments. On the other hand, the results in [6] show a promising gain in the power map accuracy especially in the scenarios with significant multi-path effects. Note, however,

that the dimensionality of the feature space considered in [6] grows with the number of base stations in the covered area, and some locations may not receive signals from all base stations; this can lead to having data points with missing features, generating the need for dimensionality reduction and data completion techniques. In opposition, LocB methods consider as their feature space the set of possible locations, which has constant dimensionality (2 or 3).

A. Main Contributions

The main idea of this work is to build a map estimation algorithm that can combine both the LocB and the LocF methods in a way that exploits the knowledge about location uncertainty. Notice that the more precise the location information of a node is, the more reliable the LocB map estimate is expected to be. An uncertainty measure regarding the location estimate acts, intuitively, as a measure of reliability of the LocB method (or expert) relative to the LocF method (or expert).

It is demonstrated in this work that such an uncertainty estimate can be exploited to build more accurate CG maps. To this end, the proposed approach estimates the CG from: a) the LocF features (e.g. center of mass (CoM) of the channel impulse response as in [6]), b) the estimated location of transmitter and receiver, and c) the location uncertainty information. We postulate that the complementary benefits of the LocF and LocB estimators can be exploited efficiently by learning a LocB estimator and a LocF estimator, and combining their outputs with a gating function that incorporates the localization uncertainty. With higher uncertainty in the location estimation, the final estimate relies more on a LocF estimator; while otherwise, a LocB estimator has more weight on the final estimate. The weights of the LocB and LocF estimates are given by the gating function, which is optimized jointly with the experts or estimators.

The aforementioned way of estimating the CG is also attractive because of its simplicity, incorporating the divide-and-conquer philosophy of the mixture-of-experts (MoE) methods [10]. This is suitable for our development because we seek an approach that enables estimating CGs at arbitrary locations, in addition to the sensor positions (although the accuracy of such estimates will be lower as compared to the accuracy obtained at the locations with sensor nodes). We expect that in real scenarios, some training points will have only one type of feature available. For the other training points with both types of features (LocB and LocF), the learning will be enhanced.

Moreover, this work is (to the best of our knowledge) the first to apply the LocF approach to the estimation of CG maps.

The rest of the paper is structured as follows. In Sec. II, our learning model is put forth in a mixture-of-experts (MoEs) framework. In Sec. III, the CG cartography problem is formulated. In Sec. IV a solution is derived, and the choice of hyper-parameters is later discussed (Sec. IV-A). The results of numerical experiments are presented in Sec. V, and Sec. VI concludes the paper.

II. Modeling

Consider a transmitter located at $\boldsymbol{x}_t \in \mathcal{R}$, and a receiver located at $\boldsymbol{x}_r \in \mathcal{R}$, where $\mathcal{R}$ is the region of interest (typically a subset of $\mathbb{R}^2$ or $\mathbb{R}^3$). The CG between them is denoted by $C_{t,r} \in \mathbb{R}$. The main goal of CG cartography is to learn a function that can give a point estimate $\hat{c}_{t,r}$ of $C_{t,r}$, given the information gathered at each of the two terminals. Specifically, let $\hat{\boldsymbol{x}}_t$ denote an estimate of the transmitter's location, and $e_t \in \mathbb{R}_+$ an uncertainty measure regarding $\hat{\boldsymbol{x}}_t$. Let $\boldsymbol{\phi}_t \in \mathbb{R}^M$ denote the vector containing the LocF features extracted from the pilot signals the same sensing node has received, where M is the total number of features extracted (more information about how $\boldsymbol{\phi}_t$ is obtained can be found in Sec. II-B). A representation of the available information at the transmitter comes from stacking the aforementioned variables in the vector $\boldsymbol{\psi}_t := [\boldsymbol{\phi}_t^\top, \ \hat{\boldsymbol{x}}_t^\top, \ e_t]^\top$ (analogously for the information at the receiver, $\boldsymbol{\psi}_r$). The function we seek to learn is expressed in the form

$$\hat{c}_{t,r} = f(\boldsymbol{\psi}_t, \boldsymbol{\psi}_r).$$

With the twofold goal of estimating CG at arbitrary pairs of transmitter-receiver locations, and also where the sensing nodes do not have an accurate location estimate, our goal is to develop an approach such that for a query where $\boldsymbol{\phi}_t$ and $\boldsymbol{\phi}_r$ are available and $(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_r)$ have large uncertainty (or even are missing), one can leverage the LocF technique [6]; and whenever $(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_r)$ are accurate and the features in $\boldsymbol{\phi}$ are noisy, it should work similarly to LocB approaches such as [2], [11]. And between these two extreme situations, the idea discussed here is to combine both LocF and LocB estimates, exploiting the knowledge of uncertainty in the location information.

A popular approach to combine the estimates into f is MoEs [10]. It is common practice to use a convex combination of the output of each of the experts, mainly because the combination coefficients can be interpreted as conditional probabilities of the events defining which of the experts has the best estimate for a given data point. This can be expressed without loss of generality by defining the gating function $\tilde{g}(\cdot)$:

$$f(\boldsymbol{\psi}_t, \boldsymbol{\psi}_r) = \tilde{g}(\boldsymbol{\psi}_t, \boldsymbol{\psi}_r) f_l(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_r) + \big(1 - \tilde{g}(\boldsymbol{\psi}_t, \boldsymbol{\psi}_r)\big) f_p(\boldsymbol{\phi}_t, \boldsymbol{\phi}_r). \tag{1}$$

Such a gating mechanism is widespread in the ML literature not only because it is used in MoE, but also because of its presence in recurrent neural networks [12]. We postulate that incorporating the location uncertainty measure in the input of this gating function will result into an improved performance of the MoE-based CG map. This approach is justified because the LocB estimator is expected to perform better than the LocF one when the location estimate is sufficiently good. What "sufficiently good" means is something we expect the model will learn from the data. Moreover, the mixture allows each expert to focus its resources in learning its own part of the CG map in those areas where it is expected to perform better. The empirical results in Sec. V support this idea.

A. Simple MoE Model: Gating as a Function of the Localization Uncertainty Measures

The main idea in the model in (1) is to restrict each of the two experts in the mixture to have as input either location estimates, or LocF features. In order to keep the model complexity at the minimum, we propose to restrict the gating function to take as input only the uncertainty in estimating the locations $\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_r$. This yields the simplest possible model that combines the aforementioned experts, and takes the location uncertainty into account.

With $e_{\boldsymbol{x},t}$ and $e_{\boldsymbol{x},r}$ respectively denoting the uncertainty measures referring to estimating $\hat{\boldsymbol{x}}_t$ and $\hat{\boldsymbol{x}}_r$, we will design a gating function

$$g : \mathbb{R}^2_+ \to [0,\, 1] \tag{2}$$

that takes the localization error vector $\boldsymbol{e} := [e_{\boldsymbol{x},t},\ e_{\boldsymbol{x},r}]^\top$ as an input for any transmitter-receiver pair (t, r). A more sophisticated model could also incorporate the uncertainty associated with ϕ, but it is not clear if the gain in performance would be significant. The MoE model can be written as:

$$f(\boldsymbol{\psi}_t, \boldsymbol{\psi}_r) = g(\boldsymbol{e}) f_l(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_r) + (1 - g(\boldsymbol{e}))\, f_p(\boldsymbol{\phi}_t, \boldsymbol{\phi}_r) \tag{3}$$

For this model, it is clear that $g(\boldsymbol{e})$ should give less emphasis on $f_l(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_r)$ when either $e_{\boldsymbol{x},t}$ or $e_{\boldsymbol{x},r}$ is large.

Successful learning of the hybrid (MoE) model $f(\boldsymbol{\psi})$ entails some advantages: The information carried by the location uncertainty measure allows to give as much weight as is needed to the LocB and LocF estimates. Whenever the location estimates are deemed reliable, the MoE gives more weight to the location-based estimate, which mitigates the relative difficulty of the location-free estimation to generalize due to the higher dimensionality of the positioning features . To see this, consider the case where two different queries are performed for the same Tx-Rx pair, but the positioning (pilot) signals are received from different location sources. While a pure localization-free approach might fail to generalize, the CG can still be estimated successfully if the localization algorithm identifies the location correctly. One can even evaluate the CG map for an arbitrary pair of locations when there is no sensing node at either one or both locations. In such a case, the estimate is simply given by $f_l(\boldsymbol{x}_t, \boldsymbol{x}_r)$.

The gating function should be component-wise non-increasing, i.e.,

$$g(\boldsymbol{e}) \leq g(\boldsymbol{e}')\ \forall\ (\boldsymbol{e}, \boldsymbol{e}') \text{ such that } \boldsymbol{e} \succeq \boldsymbol{e}'$$

where the notation $\boldsymbol{a} \succeq \boldsymbol{b}$ denotes for $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^N$ that $[\boldsymbol{a}]_i \geq [\boldsymbol{b}]_i\ \forall i \in [1, N]$. Under the assumption of symmetric channels, $g(\boldsymbol{e})$ should also be symmetric, i.e.

$$g([e_{\boldsymbol{x},t},\ e_{\boldsymbol{x},r}]^\top) = g([e_{\boldsymbol{x},r},\ e_{\boldsymbol{x},t}]^\top);\ \forall\ \boldsymbol{e} \in \mathbb{R}^2_+.$$

For large $\|\boldsymbol{e}\|$, meaning an unknown location, the weight given by the gating function to the LocB estimator should vanish: $\lim_{\|\boldsymbol{e}\| \to \infty} g(\boldsymbol{e}) = 0$. However, it is not necessary to force $g(\boldsymbol{0}) = 1$, as the LocB map is imperfect and a certain contribution from the LocF one may give better performance, even if some locations are deemed perfectly known.

B. Positioning Signal-Based Features

The LocF features extracted from the pilot signals that will be used in this work are the centers of mass (CoMs) described in [6]. Let $CoM_{m,n}$ denote the center of mass of the cross-correlation between the pilot received at the n-th sensing node from the m-th positioning signal source, and the pilot received from the reference source (which is arbitrary and the same for all sensing nodes). The feature vector is then $\boldsymbol{\phi}_n := [CoM_{1,n}, \ldots, CoM_{M,n}]$. The reason of the choice of this kind of features is that they evolve smoothly over space and are robust to the pilot distortions caused by multipath. Therefore, the function f_p can be easily learnt with such features.

When the region to map is large, it is likely that some of the base stations that send pilot signals are so far away from a given location that some features become missing, either in some of the training points, or in query points. While a few missing TDoAs is not a big issue for a localization algorithm as long as enough sources are visible, the way the LocF map is designed requires all entries in the query feature vector to carry values. The technique for imputing such missing features in [6] can be seamlessly applied in the application in this work. In a nuthshell, this technique is based in the assumption that the LocF features lie in a low-dimensionality subspace. Such a subspace is learnt from the training data using a low-rank matrix completion technique.

C. Location Estimates and Uncertainty measures

Location estimates can be obtained in practice by extracting TDoA measurements from the pilot signals [13], and solving the localization problem e.g. along the lines of [5]. It is assumed that the location estimator also gives a scalar measure of the uncertainty of the location estimate. This measure can be, e.g. the spectral radius of a covariance matrix, or the diameter of an uncertainty region for a given level of confidence. The procedure for obtaining such an uncertainty measure is left out of the scope of the present paper. Our experiments will rely on synthetically-generated uncertainty measures.

III. Problem Formulation

With N_p denoting the number of training transmitter-receiver pairs, let $t(n)$ and $r(n)$ respectively denote the indices of the transmitter and the receiver of the n-th pair; and let $\tilde{c}_n$ denote the measured CG between them (i.e., a noisy observation of $C_{t(n),r(n)}$). Adopting a regularized least-squares criterion, the CG map training can be expressed as:

$$\underset{f \in \mathcal{F}}{\text{minimize}} \quad \frac{1}{N_p} \sum_{n=1}^{N_p} \left(\tilde{c}_n - f(\boldsymbol{\psi}_{t(n)}, \boldsymbol{\psi}_{r(n)})\right)^2 + \lambda \tilde{\Omega}(f) \tag{4}$$

One valid approach is to define a neural network (NN) architecture, let $\mathcal{F}$ denote the set of all functions that NN can express, and define $\tilde{\Omega}$ as a regularizer that depends on the neural weights. However, the number of training samples for such an NN to achieve good generalization may be far beyond the number of samples available in a realistic practical case.

We aim at learning the function f in a structured way, by using the MoE described in section II. We expect the number of samples needed for good generalization to be much smaller than that with a generic model.

The joint optimization of the experts and the gating function is written as the regularized functional estimation problem (5) at the bottom of the page, where $\mathcal{G}$ denotes the set of instances of $g(\cdot)$ that have the properties discussed at the end of section II-A, and $\mathcal{F}_p$ and $\mathcal{F}_l$ are model-specific spaces or sets of functions: e.g. reproducing kernel Hilbert spaces (RKHS) for a given kernel, or functions implemented by an NN. The terms between parenthesis multiplying λ_p and λ_l are intended for balancing the contribution of the regularization terms for any value of $g(\boldsymbol{e}_n)$. If these terms were absent, many algorithmic attempts to solve this problem would very likely fall into one of the two trivial solutions, namely: $g(\boldsymbol{e}) = 0 \ \forall \boldsymbol{e}$ or $g(\boldsymbol{e}) = 1 \ \forall \boldsymbol{e}$, which respectively imply $f(\boldsymbol{\psi}_n) = f_p(\boldsymbol{\phi}_{t(n)}, \boldsymbol{\phi}_{r(n)})$, or $f(\boldsymbol{\psi}_n) = f_l(\hat{\boldsymbol{x}}_{t(n)}, \hat{\boldsymbol{x}}_{r(n)})$. The problem of estimating the coefficients of a set of kernel machines whose outputs are combined using a given gating function is presented and discussed in [14]. Differently, the joint optimization of the experts and the gating function is done in a novel way here, exploiting the problem structure to yield a low-complexity algorithm. Upon scaling up the objective by the constant N_p, and defining

$$\boldsymbol{f}_l := [f_l(\hat{\boldsymbol{x}}_{r(1)}, \hat{\boldsymbol{x}}_{r(1)}), \ldots, f_l(\hat{\boldsymbol{x}}_{t(N_p)}, \hat{\boldsymbol{x}}_{r(N_p)})]^\top, \tag{6a}$$

$$\boldsymbol{f}_p := [f_p(\boldsymbol{\phi}_{t(1)}, \boldsymbol{\phi}_{r(1)}), \ldots, f_p(\boldsymbol{\phi}_{t(N_p)}, \boldsymbol{\phi}_{r(N_p)})]^\top, \tag{6b}$$

$$\boldsymbol{g} := \left[g(\boldsymbol{e}_1), g(\boldsymbol{e}_2), \ldots g(\boldsymbol{e}_{N_p})\right]^\top, \tag{6c}$$

the problem (5) can be rewritten equivalently as

$$\begin{aligned} \underset{f_l \in \mathcal{F}_l, f_p \in \mathcal{F}_p, g \in \mathcal{G}}{\text{minimize}} \quad & \|\tilde{\boldsymbol{c}} - \boldsymbol{g} \odot \boldsymbol{f}_l - (\boldsymbol{1} - \boldsymbol{g}) \odot \boldsymbol{f}_p\|^2 \\ & + \lambda_l \boldsymbol{1}^\top \boldsymbol{g}\, \Omega(f_l) + \lambda_p \boldsymbol{1}^\top (\boldsymbol{1} - \boldsymbol{g}) \Omega(f_p), \end{aligned} \tag{7}$$

where $\odot$ denotes element-wise vector (Hadamard) product.

IV. Optimization

At this point, the functions f_p, f_l, g can be learnt using several different approaches. If such functions are expressed parametrically, one can compute (automatically via back propagation) the gradient of the cost function in (7) , and run a gradient-based minimization algorithm to seek a local minimum (as is common practice for NNs).

An alternative approach is to solve the problem in (7) using block-coordinate minimization (BCM):

$$f_l^{(k+1)} := \arg\min_{f_l \in \mathcal{F}_l} \left\| \tilde{\boldsymbol{c}} - (\boldsymbol{1} - \boldsymbol{g}^{(k)}) \odot \boldsymbol{f}_p^{(k)} - \boldsymbol{g}^{(k)} \odot \boldsymbol{f}_l \right\|^2 + \lambda_l 1^\top \boldsymbol{g}\, \Omega(f_l) \tag{8a}$$

$$f_p^{(k+1)} := \arg\min_{f_p \in \mathcal{F}_p} \left\| \tilde{\boldsymbol{c}} - (\boldsymbol{1} - \boldsymbol{g}^{(k)}) \odot \boldsymbol{f}_p - \boldsymbol{g}^{(k)} \odot \boldsymbol{f}_l^{(k+1)} \right\|^2 + \lambda_p \boldsymbol{1}^\top (\boldsymbol{1} - \boldsymbol{g}) \Omega(f_p) \tag{8b}$$

$$g^{(k+1)} := \arg\min_{g \in \mathcal{G}} \left\| \tilde{\boldsymbol{c}} - \boldsymbol{f}_p^{(k+1)} - (\boldsymbol{f}_p^{(k+1)} - \boldsymbol{f}_l^{(k+1)}) \odot \boldsymbol{g} \right\| + \left(\lambda_l \Omega(\boldsymbol{f}_l^{(k+1)}) - \lambda_p \Omega(\boldsymbol{f}_p^{(k+1)}) \right) \boldsymbol{1}^\top \boldsymbol{g} \tag{8c}$$

BCM converges monotonically to a local minimum of (7).

Whenever $\mathcal{F}_l$ is an RKHS (denoted by $\mathcal{H}_l$) with associated kernel $\kappa_l(\cdot, \cdot)$, and the regularizer Ω is the associated RKHS norm $\|\cdot\|_{\mathcal{H}_l}^2$, the subproblem (8a) is a standard kernel ridge regression (KRR) problem; the same applies to (8b). According to the Representer Theorem [15], there exist minimizers for (8a) and (8b) with the following forms, respectively:

$$f_l(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_r) = \sum_{n=1}^{N_p} \alpha_{l,n} \kappa_l([\hat{\boldsymbol{x}}_t^\top,\ \hat{\boldsymbol{x}}_r^\top]^\top, [\hat{\boldsymbol{x}}_{t(n)}^\top,\ \hat{\boldsymbol{x}}_{r(n)}^\top]^\top) \tag{9a}$$

$$f_p(\boldsymbol{\phi}_t, \boldsymbol{\phi}_r) = \sum_{n=1}^{N_p} \alpha_{p,n} \kappa_p([\boldsymbol{\phi}_t^\top,\ \boldsymbol{\phi}_r^\top]^\top, [\boldsymbol{\phi}_{t(n)}^\top,\ \boldsymbol{\phi}_{r(n)}^\top]^\top). \tag{9b}$$

and if we define the kernel matrix $\boldsymbol{K}_l$ such that $[\boldsymbol{K}_l]_{ij} = \kappa_l([\hat{\boldsymbol{x}}_{t(i)}^\top,\ \hat{\boldsymbol{x}}_{r(i)}^\top]^\top, [\hat{\boldsymbol{x}}_{t(j)}^\top,\ \hat{\boldsymbol{x}}_{r(j)}^\top]^\top)$; define $\boldsymbol{K}_p$ analogously, and $\boldsymbol{D}^{(k)} \triangleq \mathrm{Diag}(\boldsymbol{g}^{(k)})$; solving (8a-8b) boils down to:

$$\boldsymbol{\alpha}_l^{(k+1)} := \arg\min_{\boldsymbol{\alpha}_l} \left\| \tilde{\boldsymbol{c}} - (\boldsymbol{I} - \boldsymbol{D}^{(k)}) \boldsymbol{K}_p \boldsymbol{\alpha}_p^{(k)} - \boldsymbol{D}^{(k)} \boldsymbol{K}_l \boldsymbol{\alpha}_l \right\|^2 + \lambda_l \boldsymbol{1}^\top \boldsymbol{g}\, \boldsymbol{\alpha}_l^\top \boldsymbol{K}_l \boldsymbol{\alpha}_l \tag{10a}$$

$$\boldsymbol{\alpha}_p^{(k+1)} := \arg\min_{\boldsymbol{\alpha}_p} \left\| \tilde{\boldsymbol{c}} - (\boldsymbol{I} - \boldsymbol{D}^{(k)}) \boldsymbol{K}_p \boldsymbol{\alpha}_p - \boldsymbol{D}^{(k)} \boldsymbol{K}_l \boldsymbol{\alpha}_l^{(k+1)} \right\|^2 + \lambda_p \boldsymbol{1}^\top (\boldsymbol{1} - \boldsymbol{g})\, \boldsymbol{\alpha}_p^\top \boldsymbol{K}_p \boldsymbol{\alpha}_p \tag{10b}$$

and it holds that $\boldsymbol{f}_p = \boldsymbol{K}_p \boldsymbol{\alpha}_p$, and $\boldsymbol{f}_l = \boldsymbol{K}_l \boldsymbol{\alpha}_l$.

When both $\mathcal{F}_l$ *and* $\mathcal{F}_p$ are RKHSs, (10) can be substituted with a joint optimization whose closed form is (11) (shown at the top of next page). It turns out that a related model is proposed and discussed in [14], but the gating function there is a generic (softmax) function, which would make the optimization in (8c) nonconvex. An alternative approach is proposed in this paper, based on exploiting the structure of the problem at hand to design a low-complexity solver.

Recall that in the RKHS case, (8a-8b) become convex problems. If the optimization over $g(\cdot)$ is formulated as a convex problem, one can expect much more efficient learning.

$$\begin{aligned} \underset{f_p \in \mathcal{F}_p, f_l \in \mathcal{F}_l, g \in \mathcal{G}}{\text{minimize}} \quad & \tfrac{1}{N_p} \textstyle\sum_{n=1}^{N_p} \left(\tilde{c}_n - g(\boldsymbol{e}_n) f_l(\hat{\boldsymbol{x}}_{t(n)}, \hat{\boldsymbol{x}}_{r(n)}) - (1 - g(\boldsymbol{e}_n)) f_p(\boldsymbol{\phi}_{t(n)}, \boldsymbol{\phi}_{r(n)}) \right)^2 \\ & + \left(\tfrac{1}{N_p} \textstyle\sum_{n=1}^{N_p} g(\boldsymbol{e}_n) \right) \lambda_l \Omega(f_l) + \left(\tfrac{1}{N_p} \textstyle\sum_{n=1}^{N_p} 1 - g(\boldsymbol{e}_n) \right) \lambda_p \Omega(f_p), \end{aligned} \tag{5}$$

$$\begin{bmatrix}\boldsymbol{\alpha}_p^{(k+1)}\\ \boldsymbol{\alpha}_l^{(k+1)}\end{bmatrix} = \begin{bmatrix}(\boldsymbol{I}_{N_p}-\boldsymbol{D}^{(k)})^2\boldsymbol{K}_p+\lambda_p\mathbf{1}^\top(\mathbf{1}-\boldsymbol{g})\boldsymbol{I}_{N_p} & (\boldsymbol{I}_{N_p}-\boldsymbol{D}^{(k)})\boldsymbol{D}^{(k)}\boldsymbol{K}_l\\ \boldsymbol{D}^{(k)}(\boldsymbol{I}_{N_p}-\boldsymbol{D}^{(k)})\boldsymbol{K}_p & (\boldsymbol{D}^{(k)})^2\boldsymbol{K}_l+\lambda_l\mathbf{1}^\top\boldsymbol{g}\boldsymbol{I}_{N_p}\end{bmatrix}^{-1}\begin{bmatrix}(\boldsymbol{I}_{N_p}-\boldsymbol{D}^{(k)})\tilde{\boldsymbol{c}}\\ \boldsymbol{D}^{(k)}\tilde{\boldsymbol{c}}\end{bmatrix} \tag{11}$$

In fact, one can directly incorporate the properties of $g(\cdot)$ described in Sec. II-A in the definition of $\mathcal{G}$, so that (8c) becomes:

$$\begin{aligned}\underset{\boldsymbol{g}\in\mathbb{R}^{N_p}}{\text{minimize}} \quad & \left\|\tilde{\boldsymbol{c}}-\boldsymbol{f}_p^{(k+1)}-\text{Diag}(\boldsymbol{f}_l^{(k+1)}-\boldsymbol{f}_p^{(k+1)})\boldsymbol{g}\right\|^2\\ & +\left(\lambda_l\Omega(\boldsymbol{f}_l^{(k+1)})-\lambda_p\Omega(\boldsymbol{f}_p^{(k+1)})\right)\mathbf{1}^\top\boldsymbol{g} \quad (12a)\\ \text{s. to:} \quad & \mathbf{0}\preceq\boldsymbol{g}\preceq\mathbf{1} \quad (12b)\\ & [g]_i\le[g]_j\ \forall\ (i,j)\ \text{ s.t. } \boldsymbol{e}_i\succeq\boldsymbol{e}_j \quad (12c)\end{aligned}$$

which is a standard convex quadratic problem with affine constraints. Regarding symmetry, it can be enforced easily (not only on the gating function but also on f_p and f_l) by augmenting the training set, i.e., for each sample $(\tilde{c}_n, \boldsymbol{\psi}_{t,n}, \boldsymbol{\psi}_{r,n})$, adding its counterpart $(\tilde{c}_n, \boldsymbol{\psi}_{r,n}, \boldsymbol{\psi}_{t,n})$ to the training set. Once $\boldsymbol{g}$ is found, any gating function $g(\cdot)$ in agreement with (6c) will be optimal for the training set for fixed $\boldsymbol{f}_p, \boldsymbol{f}_l$. Once the overall procedure has converged, an instance of $g(\cdot)$ can be recovered easily by interpolating the values in $\boldsymbol{g}$ with any interpolation technique (e.g., KNN).

Remark. The collection of constraints in (12c) is written with as many constraints as partial order relations in the set $\{\boldsymbol{e}_n\}_{n=1}^{N_p}$, for clarity. The number of constraints grows superlinarly with N_p. To avoid excess of complexity, the number of constraints can be reduced by building a directed acyclic graph (DAG) with the latter order relations, and computing its transitive reduction. This results into a DAG encoding the minimal set of constraints (that implies all others by the transitive property), yielding an equivalent problem with many fewer constraints.

A. Hyperparameter selection

If a Gaussian/RBF kernel is used, the kernel functions (κ_l, κ_p) have width parameters σ_l and σ_p. The proposed estimator has then the following hyperparameters: $\lambda_l, \lambda_p, \sigma_l, \sigma_p$. It may be challenging to adjust all these hyperparameters by grid-search and cross-validation (CV), for two reasons: a) the dimensionality of the search space is 4, as opposed to the search space for LocB or LocF which is 2; b) the computation required to train the MoE is much higher than that for each of the experts separately, because of the iterative loop and the relatively slow convergence of BCM. A simplified procedure is proposed, based on selecting the hyperparameters which are CV-optimal for the LocB and LocF estimators separately, and then reusing the same hyperparameters for the MoE. The procedure is tabulated as Alg. 1.

The computational cost of Alg. 1 depends on: a) the number of elements in the grids where λ_p and λ_l are searched; b) the number of training samples N_p, and c) the number of iterations required for the for loop in step 5 to converge. The dominating step with a practical configuration is step 6, whose complexity is $\mathcal{O}(N_p^3)$ due to the matrix inversion in (11).

Algorithm 1 Hyper-parameter selection and MoE training

Input: Training data $\{\boldsymbol{\psi}_{t,n}, \boldsymbol{\psi}_{r,n}, \tilde{c}_n\}_{n=1}^N$
Output: CG estimating function $f(\boldsymbol{\psi}_t, \boldsymbol{\psi}_r)$

1: Select hyperparameters (λ_p, σ_p) for the LocF CG estimator via CV and grid search
2: Select hyperparameters $(\lambda_l, \sigma_l)$for the LocB CG estimator via CV and grid search
3: Initialize $g(\boldsymbol{e}) = 1/2\,\forall \boldsymbol{e}$ by defining $\boldsymbol{g} = \mathbf{1}/2$
4: Set $(\lambda_l, \lambda_p, \sigma_l, \sigma_p)$ as hyperparameters for the MoE
5: **for** $k = 1, 2, \ldots$ **do** (until convergence)
6: Joint KRR coefficients optimization via (11)
7: Optimize gating function via (12)
8: **Return** $f(\boldsymbol{\psi}_t, \boldsymbol{\psi}_r)$ via (3)

V. Experiments

A wireless propagation environment is simulated using an adapted version of the ray-tracing software in [16]. The original code considers a set of walls and several sources to generate a power map accounting for direct, first, and second order reflected paths. The original source code has been modified to generate CGs between any two points in the area where the set of walls lie.

A set of positioning sources (e.g., base stations) are also simulated, and their pilot signals are transmitted through the aforementioned environment, so the received pilots are affected by the same multipath and attenuation that creates the CGs. The features to be used by the LocF estimators are obtained as the CoM of the cross-correlation between each pair of localization source pilots [cf. Sec. II-B]. The location estimates $\hat{\boldsymbol{x}}$ to be used by the LocB estimator are generated syntheticallyby adding random noise $\sim\mathcal{N}(\mathbf{0}, \sigma_x^2\boldsymbol{I})$ to the true locations of the simulated nodes. The location uncertainties e_x are also synthetically generated by adding random noise $\sim\mathcal{N}(0, \sigma_e^2)$ to the Euclidian distance between the true location and its estimate. For these experiments, $\sigma_x := 7$ m, and $\sigma_e := 0.3$ m (so that the uncertainty is significant and its measure is consistent with the deviation of the estimate from the true location). Training and testing data are generated by spreading the sensing UE terminals in the area uniformly at random, and generating for each pair $(t(n), r(n))$ the channel gain observation $\tilde{c}_n := C_{t(n),r(n)} + \epsilon_n$, with $\epsilon_n \sim \mathcal{N}(0, \sigma_c^2)$. Channel gains are expressed in dB, and $\sigma_c = 2dB$.

A first experiment is run to visualize the estimators resulting from the proposed algorithm. Steps 1 and 2 produce a LocF and a LocB CG estimators, which are not part of any mixture. Once the joint optimization of $\boldsymbol{f}_l, \boldsymbol{f}_p, \boldsymbol{g}$ is done (steps 5-7),

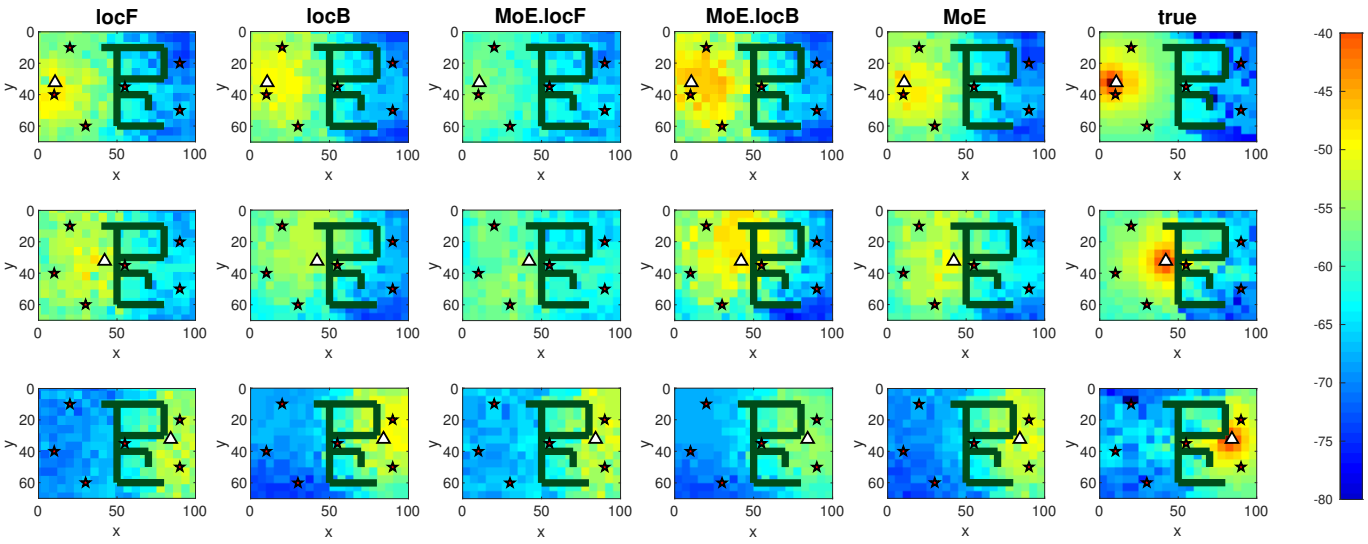

Fig. 1: Colormaps for several slices of the CG map produced by each estimator in experiment 1. Dark lines represent walls, and stars represent positioning pilot signal sources. Each pixel's color indicates the estimated CG in dB between a transmitter located at the triangle and a receiver located at the pixel center. Each column corresponds to a different estimator, where "MoE.locF" ("MoE.locB") denotes the LocFree (LocBased) expert of the mixture; $N_p = 2000$ samples.

not only the final estimator $f(\cdot,\cdot)$ is available, but also $f_l(\cdot,\cdot)$ and $f_p(\cdot,\cdot)$ as a by-product[1], which are different from the estimators trained in steps 1 and 2. Fig. 1 shows a subset of the estimated CGs for each of these 5 estimators, and also shows the true gains for comparison. It can be observed that, in the first two rows, MoE.locF underestimates the CG in several areas, whereas MoE.locB tends to overestimate them. Interestingly, in the third row one can observe the converse situation.

To analyze the difference in performance between the proposed mixture estimator MoE, and the simple estimators LocB and LocF, a second experiment is run. The goal is to compare the normalized mean square error (NMSE) incurred by each of the aforementioned estimators for different number of training samples, shown in Fig. 2. The NMSE is defined as

$$\text{NMSE} = \mathbb{E}\{|f(\boldsymbol{\psi}_t, \boldsymbol{\psi}_r) - C_{t,r}|^2\}/\text{var}\{C_{t,r}\}$$

where the expectation and variance are taken over locations uniformly distributed across the region of interest. The main feature to remark in Fig. 2 is that, above a certain number of training samples (800 for this experiment), the MoE estimate (which combines MoE.locB and MoE.locF) achieves a better performance than the (simple) LocF or LocB estimators. This suggests that a training set with too few samples does not carry enough information to successfully learn the three functions involved in MoE.

The increase of the NMSE incurred by MoE.locB when the number of samples becomes higher is also remarkable. A possible explanation for this behaviour is that the MoE.locB

[1]The functions f, f_l and f_p are respectively labeled in Fig. 1 as MoE, MoE.locB and MoE.locF.

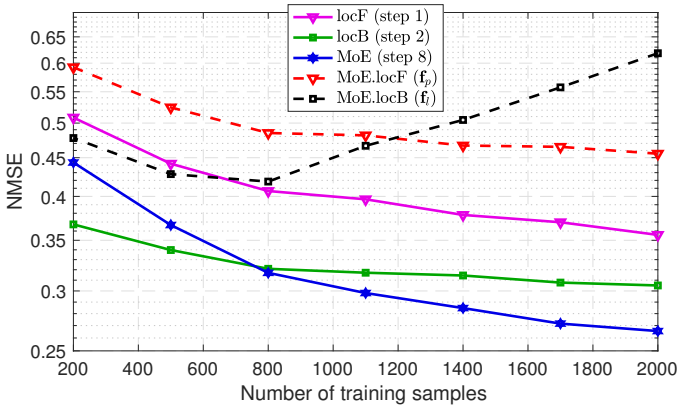

Fig. 2: Comparison of test NMSE for several different learning instances with a different number of training samples. Hyperparameters determined by Alg. 1. Results averaged over 30 Monte Carlo realizations.

spatial function becomes more complex (rougher) in an attempt to make the MoE fit the data better. Increasingly complex estimators usually lead to overfitting but, according to Fig. 2, MoE does not overfit. This suggests that the gating function is successfully filtering out the abrupt changes in MoE.locB.

ACKNOWLEDGEMENT

The authors want to acknowledge Assoc. Prof. Daniel Romero for his participation in discussions during an early stage of this work.

VI. Conclusions

A mixture-of-experts (MoE) model has been proposed to map CG between the locations of any transceiver pair in the area of interest. The location-free (LocF) and location-based (LocB) approaches are combined using a gating function that incorporates the uncertainty associated with the location estimate. The proposed algorithmic approach learns the MoE.locF and MoE.locB components and the gating function using a block-coordinate minimization approach. Experiments with simulated data confirm the ability of the proposed approach to perform with lower error than the simple LocF or LocB estimators. These results motivate future work extending the experimental setup with more realistic and diverse propagation scenarios.

References

[1] M. Elnourani, M. Hamid, D. Romero, and B. Beferull-Lozano, "Underlay device-to-device communications on multiple channels," in *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Process.*, Calgary, Canada, Apr. 2018, pp. 3684–3688.

[2] S.-J. Kim, E. Dall'Anese, and G. B. Giannakis, "Cooperative spectrum sensing for cognitive radios using Kriged Kalman filtering," *IEEE J. Sel. Topics Sig. Process.*, vol. 5, no. 1, pp. 24–36, Jun. 2010.

[3] D. Lee, D. Berberidis, and G. B. Giannakis, "Adaptive Bayesian radio tomography," *IEEE Trans. Sig. Process.*, vol. 67, no. 8, pp. 1964–1977, Mar. 2019.

[4] P. Agrawal and N. Patwari, "Correlated link shadow fading in multi-hop wireless networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 9, pp. 4024–4036, Aug. 2009.

[5] G. Wang, A. M.-C. So, and Y. Li, "Robust convex approximation methods for TDOA-based localization under NLOS conditions," *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3281–3296, 2016.

[6] Y. Teganya, D. Romero, L. M. Lopez-Ramos, and B. Beferull-Lozano, "Location-free spectrum cartography," *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 4013–4026, Aug. 2019.

[7] X. Han, L. Xue, F. Shao, and Y. Xu, "A power spectrum maps estimation algorithm based on generative adversarial networks for underlay cognitive radio networks," *Sensors*, vol. 20, no. 1, pp. 311, 2020.

[8] H. Ye, G. Y. Li, B.-H. F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional gan," in *Proc. IEEE Global Commun. Conf.*, 2018, pp. 1–5.

[9] S. M. Aldossari and K.-C. Chen, "Machine learning for wireless communication channel modeling: An overview," *Wireless Personal Commun.*, vol. 106, no. 1, pp. 41–70, 2019.

[10] S. Masoudnia and R. Ebrahimpour, "Mixture of experts: A literature survey," *Artif. Intellig. Review*, vol. 42, no. 2, pp. 275–293, 2014.

[11] E. Dall'Anese, S.-J. Kim, and G. B. Giannakis, "Channel gain map tracking via distributed kriging," *IEEE Trans. Vehic. Tech.*, vol. 60, no. 3, pp. 1205–1211, 2011.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[13] N. E. Gemayel, S. Koslowski, F. K. Jondral, and J. Tschan, "A low cost TDOA localization system: Setup, challenges and results," in *Proc. Workshop Pos. Navigation Commun.*, Dresden, Germany, Mar. 2013, pp. 1–4.

[14] J. Santarcangelo and X.-P. Zhang, "Kernel-based mixture of experts models for linear regression," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2015, pp. 1526–1529.

[15] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proc. Comput. Learning Theory*, Amsterdam, The Netherlands, Jul. 2001, pp. 416–426.

[16] S. Hosseinzadeh, H. Larijani, and K. Curtis, "An enhanced modified multi wall propagation model," in *Proc. IEEE Global Internet of Things Summit*, Geneva, Switzerland, Jun. 2017, pp. 1–4.