

Dual-Kernel Online Reconstruction of Power Maps

Rasoul Nikbakht, Anders Jonsson, Angel Lozano

Universitat Pompeu Fabra (UPF)

08018 Barcelona, Spain.

Email: {rasoul.nikbakht, anders.jonsson, angel.lozano} @upf.edu

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

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

I. INTRODUCTION

A. Motivation

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

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

and vehicles. With that, measurement-driven maps are not only useful for network planning, but further for tasks such as resource allocation or interference management.

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

- *Batch*. These schemes need the measurements to be available before the processing starts, so they have a considerable computational cost and they cannot adapt until a new batch is processed [8], [11]–[13].
- *Online*. These schemes process measurements as they arrive, continuously adapting [9].

B. State of the Art

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

C. Contribution

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

- It is bounded from below depending on the placement, elevation, and other aspects of the base station, as well as the frequency. For instance, the free-space pathloss at 2 GHz already equals 38.5 dB at 1 m, increasing to 60 dB at 12 m.
- It is also bounded from above because, once the signal falls well below the noise floor, the coverage is lost. While again this depends on the bandwidth, as well as the transmit power and other parameters, the in-coverage loss is typically below 160 dB [15].

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

On the basis of the foregoing observation, we posit that a better estimator can be obtained if two distinct *kernel*

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

II. PROBLEM FORMULATION

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

Given a base station, consider the relationship

$$y_n = f(\mathbf{x}_n) + \varepsilon_n \quad (1)$$

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

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

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

A. RKHS Map Formulation

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

$$\langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \rangle = \kappa(\mathbf{x}_1, \mathbf{x}_2) \quad (2)$$

and norm

$$\|\phi(\mathbf{x}_1)\| = \sqrt{\kappa(\mathbf{x}_1, \mathbf{x}_1)} \quad (3)$$

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

The representer theorem [16] states that, for several regularization methods, the optimal function estimate f^* lies in the

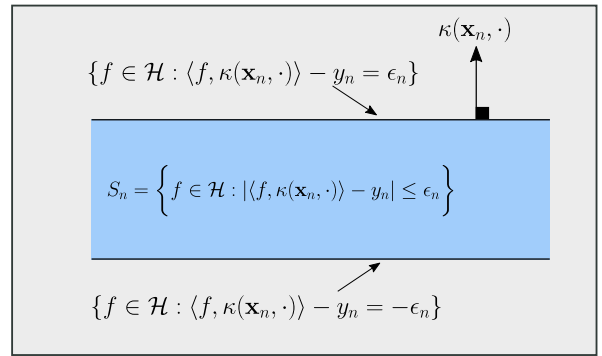


Fig. 1: Set S_n of functions that can approximate y_n with a maximum error of ε_n [17].

finite-dimensional span of the training measurements mapped to the Hilbert space, i.e.,

$$f^* = \sum_n \alpha_n \kappa(\mathbf{x}_n, \cdot). \quad (4)$$

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

$$\begin{aligned} f^*(\mathbf{x}) &= \langle f^*, \phi(\mathbf{x}) \rangle \\ &= \sum_n \alpha_n \kappa(\mathbf{x}_n, \mathbf{x}). \end{aligned} \quad (5)$$

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

Example 1. Consider the radial basis function kernel

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = e^{-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|}{D}}. \quad (6)$$

The optimal function estimate has the form

$$f^*(\mathbf{x}) = \langle f^*, \phi(\mathbf{x}) \rangle \quad (7)$$

$$= \sum_n \alpha_n e^{-\frac{\|\mathbf{x}_n - \mathbf{x}\|}{D}}. \quad (8)$$

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

III. ADAPTIVE PROJECTED SUBGRADIENT METHOD

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

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

$$S_n = \left\{ f \in \mathcal{H} : |\langle f, \kappa(\mathbf{x}_n, \cdot) \rangle - y_n| \leq \epsilon_n \right\} \quad (9)$$

forms a hyperslab, depicted in Fig 1. The intersection

$$S = \bigcap_{n \in \mathbb{N}} S_n \quad (10)$$

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

$$\{\hat{f}_n\}_{n \in \mathbb{N}} \subset \mathcal{H} \quad (11)$$

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

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

$$P_{S_n}(f) = f + \beta_f \kappa(\mathbf{x}_n, \cdot) \quad (12)$$

where

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

Denoting by

$$\mathcal{I}_n = \{n - q + 1, n - q + 2, \dots, n\} \quad (13)$$

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

$$\hat{f}_{n+1} = \hat{f}_n + \mu_n \left(\sum_{i \in \mathcal{I}_n} w_{i,n} P_{S_i}(\hat{f}_n) - \hat{f}_n \right) \quad (14)$$

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

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

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

$$\sum_i w_{i,n} = 1. \quad (15)$$

A schematic interpretation of (14) is presented in Fig 2. At step n , \hat{f}_n is projected onto the q most recent sets and the weighted sum of these projections is computed. The update direction is obtained by subtracting from the projection the previous \hat{f}_n and the step size is μ_n . The parameter q regulates a tradeoff between performance and complexity: a larger q accelerates the convergence at the expense of higher computational cost.

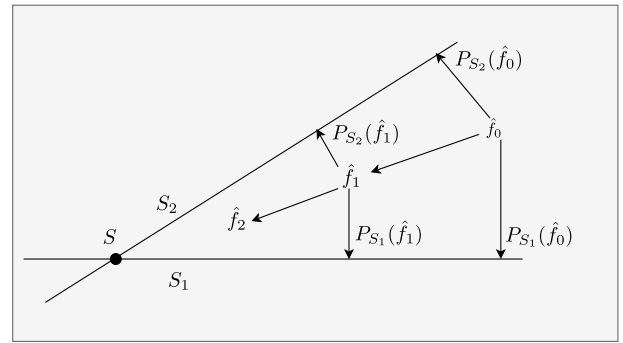


Fig. 2: Sequence of functions f_n that converge to the intersection of two convex sets S_1 and S_2 [17].

IV. DUAL-KERNEL APPROACH

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

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

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

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

Altogether then, we apply the dual-kernel function

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|}{D_{\text{DC}}}\right) + \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|}{D_{\text{var}}}\right) \quad (16)$$

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

V. SPARSIFICATION

The act of determining which measurements conform the dictionary is called sparsification. To sparsify, one can project \hat{f}_n onto a ball of a certain norm and keep only those dictionary elements with a relatively large contribution to $\|\hat{f}_n\|$, thereby

TABLE I: APSM parameters

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

effecting a regularization. The projection onto the ball $[0, \delta]$ can be obtained as [17]

$$\hat{f}_{n+1} = P_{B[0,\delta]} \left(\hat{f}_n + \mu_n \left(\sum_{i \in \mathcal{I}_n} w_{i,n} P_{S_i}(\hat{f}_n) - \hat{f}_n \right) \right) \quad (17)$$

where

$$P_{B[0,\delta]}(f) = \begin{cases} f & \|f\| \leq \delta \\ f/\|f\| & \|f\| > \delta. \end{cases} \quad (18)$$

(This projection can also be regarded as a forgetting factor, whereby older history is progressively removed from memory, fostering adaptivity.) After the projection, the indices of those dictionary elements with considerable contribution can be identified as the set

$$\mathcal{J}_n = \left\{ j \mid \frac{(\alpha_{n,j})^2}{\sum_{\ell \in \mathcal{L}_n} (\alpha_{n,\ell})^2} > \tau \right\} \quad (19)$$

where τ is a threshold, \mathcal{L}_n is the set of indexes of the dictionary elements at step n and $\{\alpha_{n,\ell}\}_{\ell \in \mathcal{L}_n}$ are the corresponding coefficients for the projected locations in the Hilbert space. One can think of (19) as removing those elements in (4) that have a negligible effect on the regression. The threshold τ regulates the tradeoff between complexity and accuracy: by increasing τ , the dictionary becomes sparser and complexity diminishes, but the accuracy drops, and vice versa.

VI. NUMERICAL EVALUATIONS

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

A. Campus Scenario

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

To begin with, let us gauge the effect of adding the DC kernel. Fig. 3 presents the NMSE and the dictionary size, as function of the sparsification threshold τ , with and without the DC kernel. The addition of this kernel reduces the NMSE for a given dictionary size or, equivalently, it reduces the dictionary size for a given NMSE.

Fig. 4 illustrates the convergence of the NMSE with the dictionary size set to 500 and therefore, applying Fig. 3, with $\tau = 1.3 \cdot 10^{-4}$ for dual kernel and $\tau = 4 \cdot 10^{-5}$ for single kernel. The addition of the DC kernel improves the

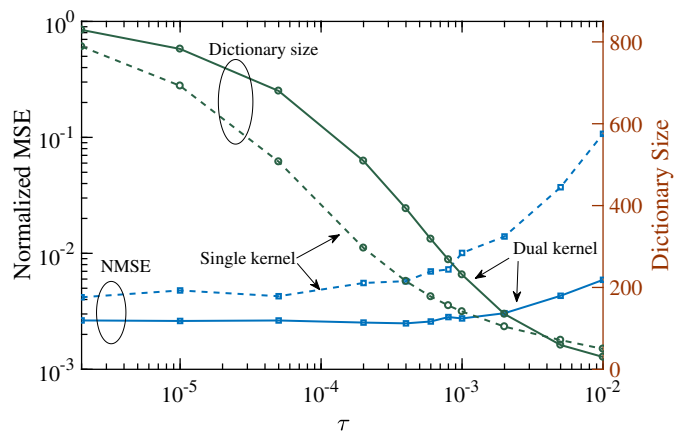


Fig. 3: NMSE (left-hand axis) and dictionary size (right-hand axis) as function of τ , with and without the DC kernel, in the campus scenario.

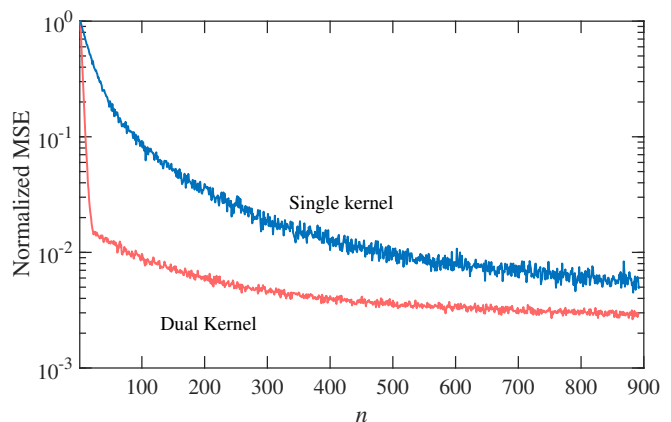


Fig. 4: NMSE as function of n , with and without the DC kernel, in the campus scenario. The dictionary size is 500.

performance markedly, reaching at $n = 250$ the NMSE that requires $n = 890$ without the DC kernel. In turn, the final NMSE shrinks by a factor of 1.7.

Finally, Table II compares the dual-kernel algorithm with some alternatives: the single-kernel algorithm, a multikernel version [9], and a batch SVM baseline. The dual-kernel approach outperforms both online alternatives in terms of NMSE, approaching the batch SVM accuracy. The reduction in NMSE does not come at the expense of an increased variability thereof; rather, the standard deviation of the squared error also abates relative to the single-kernel algorithm. (For the multikernel solution, the variance is not available.)

TABLE II: NMSE and error standard deviation in the campus scenario.

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

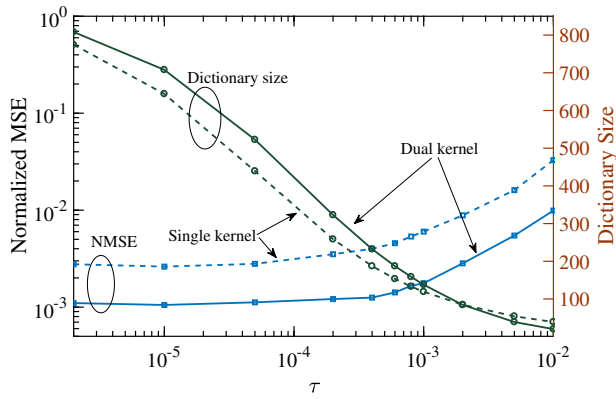


Fig. 5: NMSE (left-hand axis) and dictionary size (right-hand axis) as function of τ , with and without the DC kernel, in the urban scenario.

B. Urban scenario

The second scenario corresponds to ray-tracing data for the central part of Berlin [19]. Data for 193 BSs with a 50-m pixel resolution over an area of 12×12 km is reported in this database. The APSM settings are as per Table I. To make the results comparable with those in the campus scenario, the same number of data points is used, namely 1277, which are randomly selected at each of 1000 passes over which the error is averaged. We set the coverage threshold to 140 dB, a reasonable value for urban settings.

The effect of adding the DC kernel is shown in Fig. 5, where a significant benefit is again observed. Then, in Table III, we compare the dual-kernel approach with its single-kernel brethren and with the batch SVM baseline. The reduction in both NMSE and standard deviation is even more pronounced here than in the campus environment, even if the absolute standard deviations are larger on account of the more challenging nature of an urban setting.

VII. SUMMARY

A clean separation of the DC component from the rest of the large-scale losses, via two suitable kernel functions, benefits the online reconstruction of large-scale loss maps. The accuracy increases with respect to the single-kernel version, but also with respect to a multikernel solution. This confirms that the number of kernels should be chosen carefully to match the structure of the problem, and adding unnecessarily many kernels may end up being detrimental. With two kernels, the performance approaches the baseline represented by more complex batch schemes, where, rather than a limited-size dictionary, all measurements must be stored.

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

TABLE III: NMSE and error standard deviations in the urban scenario.

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

ACKNOWLEDGMENT

Work supported by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502), TEC2015-66228-P, and the European Research Council (grant agreement 694974).

REFERENCES

- [1] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Trans. Veh. Technol.*, vol. 29, no. 3, pp. 317–325, 1980.
- [2] "Urban transmission loss models for mobile radio in the 900 and 1800 MHz bands," European Cooperation in the Field of Scientific and Technical Research EUROCAST 231, Tech. Rep., September 1991.
- [3] V. Erceg, L. J. Greenstein, S. Y. Tjandra, S. R. Parkoff, A. Gupta, B. Kulic, A. A. Julius, and R. Bianchi, "An empirically based path loss model for wireless channels in suburban environments," *IEEE J. Select. Areas Commun.*, vol. 17, no. 7, pp. 1205–1211, Jul 1999.
- [4] C. Phillips, D. Sicker, and D. Grunwald, "Bounding the error of path loss models," in *Proc. 2011 IEEE DySPAN*, May 2011, pp. 71–82.
- [5] C.-F. Yang, B.-C. Wu, and C.-J. Ko, "A ray-tracing method for modeling indoor wave propagation and penetration," *IEEE Trans. Antennas Propagat.*, vol. 46, no. 6, pp. 907–919, 1998.
- [6] K. E. Stocker, B. E. Gschwendtner, and F. M. Landstorfer, "Neural network approach to prediction of terrestrial wave propagation for mobile radio," in *IEE Proceedings H (Microwaves, Antennas and Propagation)*, vol. 140, no. 4, 1993, pp. 315–320.
- [7] E. Ostlin, H.-J. Zepernick, and H. Suzuki, "Macrocell path-loss prediction using artificial neural networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 6, pp. 2735–2747, 2010.
- [8] R. D. Taranto, S. Muppirisetty, R. Raulefs, D. Slock, T. Svensson, and H. Wymeersch, "Location-aware communications for 5g networks: How location information can improve scalability, latency, and robustness of 5g," *IEEE Signal Processing Mag.*, vol. 31, no. 6, pp. 102–112, 2014.
- [9] M. Kasparick, R. L. G. Cavalcante, S. Valentin, S. Staczak, and M. Yukawa, "Kernel-Based Adaptive Online Reconstruction of Coverage Maps With Side Information," *IEEE Trans. Veh. Technol.*, vol. 65, no. 7, pp. 5461–5473, Jul 2016.
- [10] D. Romero, S.-J. Kim, G. B. Giannakis, and R. Lopez-Valcarce, "Learning power spectrum maps from quantized power measurements," *IEEE Trans. Signal Processing*, vol. 65, no. 10, pp. 2547–2560, 2017.
- [11] L. S. Muppirisetty, T. Svensson, and H. Wymeersch, "Spatial wireless channel prediction under location uncertainty," *IEEE Trans. Commun.*, vol. 15, no. 2, pp. 1031–1044, Feb 2016.
- [12] E. Dall'Anese, S. J. Kim, and G. B. Giannakis, "Channel Gain Map Tracking via Distributed Kriging," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 1205–1211, 2011.
- [13] J. Chen, U. Yatmali, and D. Gesbert, "Learning radio maps for UAV-aided wireless networks: A segmented regression approach," in *Proc. IEEE Int. Conf. Commun.* IEEE, May 2017, pp. 1–6.
- [14] M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *IEE Electron. Lett.*, vol. 27, no. 23, pp. 2145–2146, Nov 1991.
- [15] H. Holma and A. Toskala, *LTE for UMTS: Evolution to LTE-advanced*. John Wiley & Sons, 2011.
- [16] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Computational Learning Theory*. Springer Berlin Heidelberg, 2001, pp. 416–426.
- [17] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Processing Mag.*, vol. 28, no. 1, pp. 97–123, Jan 2011.
- [18] C. Phillips and E. W. Anderson, "CRAWDAD dataset cu/cu_wart (v. 2011-10-24)," Downloaded from https://crawdad.org/cu/cu_wart/20111024, Oct. 2011.
- [19] MOMENTUM, "Models and Simulations for Network Planning and Control of UMTS," 2004. [Online]. Available: <http://www.zib.de/momentum/data.php>