Cognitive Radio Spectrum Prediction Using Dictionary Learning

Seung-Jun Kim
Dept. of Electrical & Computer Engr.
University of Minnesota
Minneapolis, MN 55455, U.S.A.
Email: seungjun@umn.edu

Georgios B. Giannakis
Dept. of Electrical & Computer Engr.
University of Minnesota
Minneapolis, MN 55455, U.S.A.
Email: georgios@umn.edu

***Abstract*—Spatio-temporal spectrum prediction algorithms for cognitive radios (CRs) are developed using the framework of dictionary learning and compressive sensing. The interference power levels at each CR node locations are predicted using the measurements from a *subset* of CR nodes without a priori knowledge on the primary transmitters. Batch and online alternatives are presented, where the online algorithm features low complexity and memory requirements. Numerical tests verify the performance of the proposed novel methods.**

I. Introduction

The radio frequency (RF) spectrum is a precious resource that must be utilized efficiently. Fixed spectrum allocation, which confers exclusive access rights on spectrum license holders, has resulted in significant under-utilization of the valuable spectral resource, depending on time and locations [1]. The cognitive radio (CR) strategy aims at alleviating this inefficiency by allowing unlicensed secondary users to opportunistically transmit, provided that the transmissions do not disturb the communication of licensed primary users (PUs) [2].

To achieve the necessary protection of PU systems, essential elements of CR systems are spectrum sensing and intelligent resource allocation [3]. The goal of spectrum sensing is to identify unused spectral resources in the frequency, time and space domains. The "spectrum holes" can then be exploited through agile resource allocation.

A simplifying assumption often made for spectrum sensing is that the spectrum occupancy is more or less invariant over the deployment region of the CR systems. Based on this, spectrum sensing is often performed in a collaborative fashion, where the band occupancy by a common set of PU transmitters is detected using observations fused from multiple CRs [4]. This mitigates effectively fading and shadowing, which impede reliable detection of PU presence.

However, the assumption might not hold when PU systems employ a small RF footprint for significant spatial reuse, or the CR network grows in size and gets deployed in a broader geographical region. An instrumental concept in this case is the RF cartography, which provides a map of RF power distribution over space and reveals the spatial variation of spectrum occupancy [5], [6]. Such a construct is useful for optimizing CR network operations, not only in the PHY/MAC, but also in higher layers [3]. Different types of RF maps and algorithms for obtaining them have been proposed [7].

The goal of this work is to acquire the interference power present at each CR node in the network, and also to predict its future levels. A key challenge is that CRs do not have prior information on the number of PU emitters and the corresponding PU-CR channel gains, which are essential for combining the measurements from different sensors. Moreover, CRs might not be able to report their measurements every time, due to energy-saving sleep modes or congested signaling channels. Thus, the network controller must also account for missing observations by performing appropriate interpolation. Finally, the future spectrum state must be inferred based on past measurements.

A multi-dimensional logistic regression model was applied to binary spectrum occupancy observations to predict future occupancy in [8]. Traffic patterns for voice calls were analyzed using time series models so as to aid channel switching decisions of CRs in [9]. Given the model parameters, optimal channel sensing and access decisions were made in the framework of partially observed Markov decision process in [10]. However, spatial selectivity of PU interference was seldom taken into account in [8], [9], [10].

Here, contemporary tools from machine learning and compressive sensing are employed to tackle these challenges. The general dictionary learning framework [11], [12] is adapted here to learn the spatial and temporal patterns of the RF power distribution. To aid in spatial interpolation, the topology information of the CR network is exploited in a semi-supervised learning fashion. Both batch as well as on-line algorithms are developed. The on-line alternative can track the slow variation of the RF power distribution, and features low computational complexity and lax memory requirement.

The rest of this paper is organized as follows. In Sec. II, the system model and the problem statement are presented. Sec. III describes a semi-supervised dictionary learning approach for the interpolation of missing observations in space. Sec. IV extends the proposed algorithms for spatio-temporal

This work was supported by NSF grants ECCS 1002180, ECCS 1202135 and AST 1247885.

prediction. The results from numerical tests are reported in Sec. V, and conclusions are provided in Sec. VI.

II. System Model

Consider a CR network consisting of M nodes, deployed in a geographical area, over which the interference due to the incumbent PU systems vary, albeit smoothly. The CRs form a mesh network by identifying their neighbors, and cooperate for spectrum sensing. Our goal is to acquire the interference level at each CR node location, based on the measurements collected from a *subset* of the CR nodes per time. The missing measurements are due to various practical limitations, such as errors and congestion in the control channel, or the fact that radios are in the sleep mode to save battery.

Suppose that there are K PU transmitters in the area with the k-th PU transmitting at power $p_k(t)$. Let $g_{mk}(t)$ denote the channel gain from the k-th PU to the m-th CR. Then, the interference power level $\pi_m(t)$ perceived at the m-th CR can be modeled as

$$\pi_m(t) = \sum_{k=1}^{K} g_{mk}(t)p_k(t),\ m \in \mathcal{M} := \{1, 2, \ldots, M\} \quad (1)$$

Upon defining vectors $\boldsymbol{\pi}(t) := [\pi_1(t), \ldots, \pi_M(t)]^{\mathcal{T}}$ ($\cdot^{\mathcal{T}}$ denotes transposition) and $\mathbf{p}(t) := [p_1(t), \ldots, p_K(t)]^{\mathcal{T}}$ as well as matrix $\mathbf{G}(t)$ whose (m, k)-entry is $g_{mk}(t)$, the matrix-vector counterpart of (1) can be expressed as

$$\boldsymbol{\pi}(t) = \mathbf{G}(t)\mathbf{p}(t). \quad (2)$$

At each time t, a subset $\mathcal{M}^{obs}(t) \subset \mathcal{M}$ of CRs observe the interference power levels. The measurements from these CRs can be stacked in vector $\mathbf{y}^{obs}(t) \in \mathbb{R}^{|\mathcal{M}^{obs}(t)|}$ given as

$$\mathbf{y}^{obs}(t) = \mathbf{O}(t)\boldsymbol{\pi}(t) + \mathbf{z}(t) \quad (3)$$

where $\mathbf{z}(t) \in \mathbb{R}^{|\mathcal{M}^{obs}(t)|}$ is the measurement noise vector, and $\mathbf{O}(t)$ is a matrix consisting of the m-th row of an $M \times M$ identity matrix, where $m \in \mathcal{M}^{obs}(t)$.

The problem of estimating $\mathbf{G}(t)$ and $\mathbf{p}(t)$, given the past and the current measurements $\mathbf{y}^{obs}(\tau)$ for $\tau = 1, 2, \ldots, t$, when there were no missing observations was tackled using dictionary learning in [13]. Here, the goal is to predict the missing interference levels $\pi_m(t)$ for $m \in \mathcal{M}^{miss}(t) := \mathcal{M} \backslash \mathcal{M}^{obs}(t)$, given $\{\mathbf{y}^{obs}(\tau)\}_{\tau=1}^{t}$.

III. Spatial Spectrum Prediction Using Semi-Supervised Dictionary Learning

Prompted by (2), suppose that $\boldsymbol{\pi}$ can be represented as a linear combination of a small number of bases (*atoms*) taken from a dictionary. Let $\mathbf{D} \in \mathbb{R}^{M \times Q}$ denote a dictionary with Q atoms. Then, the preceding assumption amounts to

$$\boldsymbol{\pi} = \mathbf{D}\mathbf{s} \quad (4)$$

where vector $\mathbf{s} \in \mathbb{R}^Q$ is sparse. Fourier bases or the wavelet bases are some of the dictionaries often used for a variety of natural or man-made signals. Based on this model, the present paper's contribution is to leverage recent advances in compressive sensing and machine learning to predict the unobserved interference levels in space. In the following, a two-stage approach is first considered, in which the dictionary learned in the training phase is used for the desired prediction task in the operational phase. An alternative algorithm capable of performing the dictionary learning and spatial prediction simultaneously will be presented subsequently.

A. Two-Phase Batch Algorithm

Instead of using off-the-shelf bases such as Fourier or the wavelet bases, $\mathbf{D}$ can be directly learned from training data, which can be collected through a sounding procedure. In the presence of missing entries in the data, it is helpful to augment this learning process with additional structural information. In this work, the network topology information, which is typically maintained for various network control tasks such as routing, is leveraged in the framework of semi-supervised learning; see also [12]. A similar approach was successfully applied to network load prediction in [14].

Let $\mathbf{A} \in \{1, 0\}^{M \times M}$ denote the adjacency matrix of the CR network topology. Thus, the (m, m')-th entry $a_{m,m'}$ of $\mathbf{A}$ for $m, m' \in \mathcal{M}$ is 1 if nodes m and m' are neighbors, and 0 otherwise. Then, the Laplacian matrix $\mathbf{L}$ is defined as $\mathbf{L} := \mathrm{diag}(\mathbf{A}\mathbf{1}) - \mathbf{A}$, where $\mathbf{1}$ is the all-one vector, and $\mathrm{diag}(\mathbf{v})$ is a diagonal matrix with the entries of vector $\mathbf{v}$ on its diagonal.

1) Training Phase: In the training phase, given a training set $\{\mathbf{y}_n^{obs}\}_{n=1}^N$, which may contain missing entries, the goal is to form an estimate $\hat{\mathbf{D}}$ of $\mathbf{D}$ such that $\mathbf{y}_n^{obs} \approx \mathbf{O}_n\hat{\mathbf{D}}\mathbf{s}_n$ for $n = 1, 2, \ldots, N$, where coefficients $\mathbf{s}_n$ are sparse, and matrix $\mathbf{O}_n$ discards the missing entries. Specifically, the following optimization problem is solved to obtain $\hat{\mathbf{D}}$.

$$\hat{\mathbf{D}} := \arg \min_{\mathbf{D} \in \mathcal{D}, \{\mathbf{s}_n\}} \sum_{n=1}^{N} f_n(\mathbf{s}_n, \mathbf{D}) \quad (5)$$

where

$$f_n(\mathbf{s}, \mathbf{D}) := \frac{1}{2}||\mathbf{y}_n^{obs} - \mathbf{O}_n\mathbf{D}\mathbf{s}||_2^2 + \lambda_s||\mathbf{s}||_1 + \frac{1}{2}\lambda_L\mathbf{s}^{\mathcal{T}}\mathbf{D}^{\mathcal{T}}\mathbf{L}\mathbf{D}\mathbf{s} \quad (6)$$

$$\mathcal{D} := \{[\mathbf{d}_1, \ldots, \mathbf{d}_Q] \in \mathbb{R}^{M \times Q} : ||\mathbf{d}_q||_2^2 \leq 1,\ q = 1, \ldots, Q\}. \quad (7)$$

Here, the first term in (6) promotes fitness of the reconstruction to the training datum in a least-squares (LS) sense, and the ℓ_1-norm-based regularization term encourages sparsity in $\mathbf{s}$ with $\lambda_s > 0$ playing the role of a tuning parameter. The third term in (6) can be re-written as

$$\mathbf{s}^{\mathcal{T}}\mathbf{D}^{\mathcal{T}}\mathbf{L}\mathbf{D}\mathbf{s} = \sum_{m=1}^{M}\sum_{m'=1}^{M} a_{m,m'}(\pi_m - \pi_{m'})^2 \quad (8)$$

indicating that it encourages the interference levels experienced at neighboring nodes to be similar, with $\lambda_L > 0$ being a tuning parameter.

To appreciate the role of this Laplacian matrix-based regularization, suppose that a CR never reports an observation during the entire training period. Then, without the last term, the corresponding row in $\hat{\mathbf{D}}$ cannot be estimated, making it impossible to predict the interference level at this CR's location. The presence of the Laplacian term allows one to estimate the missing entry relying on neighbors' measurements.

2) Operational Phase: Once $\hat{\mathbf{D}}$ has been obtained as in (5), the operational phase predicts the interference levels. First, a sparse coding step is performed at each time t to estimate the sparse coefficient $\mathbf{s}(t)$ corresponding to the measurement $\mathbf{y}^{obs}(t)$; that is,

$$\hat{\mathbf{s}}(t) := \arg\min_{\mathbf{s}} \frac{1}{2}||\mathbf{y}^{obs}(t) - \mathbf{O}(t)\hat{\mathbf{D}}\mathbf{s}||_2^2 + \lambda_s||\mathbf{s}||_1 + \frac{1}{2}\lambda_L \mathbf{s}^{\mathcal{T}}\hat{\mathbf{D}}^{\mathcal{T}}\mathbf{L}\hat{\mathbf{D}}\mathbf{s}. \quad (9)$$

Then, the desired interference levels in $\boldsymbol{\pi}(t)$ that include the missing entries can be recovered by $\hat{\boldsymbol{\pi}}(t) := \hat{\mathbf{D}}\hat{\mathbf{s}}(t)$.

3) Implementation: Problem (9) is convex and there are various specialized algorithms available for solving the problems of this sort extremely fast. On the other hand, (5) is nonconvex, and it is difficult to obtain globally optimal solutions. However, the problem is convex with respect to $\mathbf{D}$ and $\{\mathbf{s}_n\}$ individually. Thus, to find a locally optimal solution, a block-coordinate descent (BCD) algorithm can be employed, for which convergence is well established [14].

Specifically, at the k-th iteration, updates are done as

$$\{\hat{\mathbf{s}}_n^{(k)}\} := \arg\min_{\{\mathbf{s}_n\}} \sum_{n=1}^{N} f_n(\mathbf{s}_n, \hat{\mathbf{D}}^{(k-1)}) \quad (10)$$

$$\hat{\mathbf{D}}^{(k)} := \arg\min_{\mathbf{D}\in\mathcal{D}} \sum_{n=1}^{N} f_n(\hat{\mathbf{s}}_n^{(k)}, \mathbf{D}) \quad (11)$$

where $\hat{\mathbf{D}}^{(k)}$ and $\{\hat{\mathbf{s}}_n^{(k)}\}$ are the k-th iterates. Note that (10) can be solved separately per $n = 1, 2, \ldots, N$ using the same solver as the one for (9). To solve (11), a BCD algorithm can be once again employed over the columns of $\mathbf{D}$. Define $\tilde{\mathbf{L}}_n := \mathbf{O}_n^{\mathcal{T}}\mathbf{O}_n + \lambda_L\mathbf{L}$, and let $s_{n,q}$ denote the q-th entry of vector $\mathbf{s}_n$. Then, the overall dictionary training algorithm is presented in Table I. A similar scheme for a different application was devised in [14].

Input: training set $\{\mathbf{y}_n^{obs}\}_{n=1}^N$, $\{\mathbf{O}_n\}$, initial dictionary $\mathbf{D}_0$, λ_s, $\mathbf{L}$ and λ_L
Output: $\hat{\mathbf{D}} := [\hat{\mathbf{d}}_1, \hat{\mathbf{d}}_2, \ldots, \hat{\mathbf{d}}_Q]$

Set $\hat{\mathbf{D}} = \mathbf{D}_0$
Repeat
  Perform sparse coding with fixed $\hat{\mathbf{D}}$.
  For $n = 1, 2, \ldots, N$
    $\hat{\mathbf{s}}_n = \arg\min_{\mathbf{s}} f_n(\mathbf{s}, \hat{\mathbf{D}})$
  Next n
  Perform dictionary update with fixed $\{\hat{\mathbf{s}}_n\}$
  Repeat
    For $q = 1, 2, \ldots, Q$
      $\bar{\mathbf{d}}_q = \hat{\mathbf{d}}_q + \left(\sum_{n=1}^N \tilde{\mathbf{L}}_n s_{n,q}^2\right)^{-1} \cdot \left[\sum_{n=1}^N s_{n,q}\left(\mathbf{O}_n^{\mathcal{T}}\mathbf{y}_n^{obs} - \tilde{\mathbf{L}}_n\hat{\mathbf{D}}\mathbf{s}_n\right)\right]$
      $\hat{\mathbf{d}}_q = \bar{\mathbf{d}}_q / \max\{||\bar{\mathbf{d}}_q||_2, 1\}$
    Next q
  Until convergence
Until convergence

TABLE I
A BCD ALGORITHM FOR DICTIONARY TRAINING [14].

B. Online Algorithm

In order to track time-varying statistics of the interference patterns, an online algorithm can be derived, in which the dictionary training and spatial interference prediction are performed jointly at the same time [15]. Compared to the batch training discussed in Sec. III-A, the online algorithm can perform the computation recursively, resulting in significant savings in complexity and memory.

Specifically, the following formulation is adopted, which weights recent observations more heavily.

$$\hat{\mathbf{D}}(t), \{\hat{\mathbf{s}}(t)\} = \arg\min_{\mathbf{D}\in\mathcal{D}, \{\mathbf{s}(t)\}} \sum_{\tau=1}^{t} \beta^{t-\tau}\left(\frac{1}{2}||\mathbf{y}^{obs}(\tau) - \mathbf{O}(\tau)\mathbf{D}\mathbf{s}(\tau)||_2^2 + \lambda_s||\mathbf{s}(\tau)||_1 + \frac{1}{2}\lambda_L\mathbf{s}^{\mathcal{T}}(\tau)\mathbf{D}^{\mathcal{T}}\mathbf{L}\mathbf{D}\mathbf{s}(\tau)\right) \quad (12)$$

where $\beta \in (0, 1]$ is a forgetting factor. Instead of solving problem (12) in a batch fashion for the entire time horizon $\tau = 1, 2, \ldots, t$ whenever a new observation $\mathbf{y}^{obs}(t)$ arrives at each time t, an online approach updates only the "current" coefficient vector $\hat{\mathbf{s}}(t)$, while the past ones $\hat{\mathbf{s}}(t-1), \ldots, \hat{\mathbf{s}}(1)$ are held fixed. Nevertheless, it can be shown under mild conditions that $\hat{\mathbf{D}}$ so obtained converges as $t \to \infty$ to the same $\hat{\mathbf{D}}$ as would be obtained from a batch approach [15].

Although the dictionary update depends on the entire observation history, a recursive computation can avoid storing the past observations and calculations. For this, it is useful to maintain the following quantities:

$$\mathbf{A}(t) := \sum_{\tau=1}^{t} \beta^{t-\tau}\hat{\mathbf{s}}(\tau)\hat{\mathbf{s}}^{\mathcal{T}}(\tau) = \beta\mathbf{A}(t-1) + \hat{\mathbf{s}}(t)\hat{\mathbf{s}}^{\mathcal{T}}(t) \quad (13)$$

$$\mathbf{A}_m(t) := \sum_{\tau=1}^{t} \beta^{t-\tau}\mathbb{1}_{\{m\in\mathcal{M}^{obs}(\tau)\}}\hat{\mathbf{s}}(\tau)\hat{\mathbf{s}}^{\mathcal{T}}(\tau) = \beta\mathbf{A}_m(t-1) + \mathbb{1}_{\{m\in\mathcal{M}^{obs}(t)\}}\hat{\mathbf{s}}(t)\hat{\mathbf{s}}^{\mathcal{T}}(t),\ m \in \mathcal{M} \quad (14)$$

$$\mathbf{B}(t) := \sum_{\tau=1}^{t} \beta^{t-\tau}\mathbf{O}^{\mathcal{T}}(\tau)\mathbf{y}^{obs}(\tau)\hat{\mathbf{s}}^{\mathcal{T}}(\tau) = \beta\mathbf{B}(t-1) + \mathbf{O}^{\mathcal{T}}(t)\mathbf{y}^{obs}(t)\hat{\mathbf{s}}^{\mathcal{T}}(t) \quad (15)$$

Input: online observations $\{\mathbf{y}^{obs}(t)\}$, $\{\mathbf{O}(t)\}$, initial dictionary $\mathbf{D}_0$, λ_s, $\mathbf{L}$, λ_L and $\beta \in (0,1]$
Output: $\{\hat{\boldsymbol{\pi}}(t)\}$

1: Set $\hat{\mathbf{D}}(0) = \mathbf{D}_0$, $\mathbf{A}(0) = \mathbf{0}$, $\mathbf{A}_m(0) = \mathbf{0}$ $\forall m \in \mathcal{M}$, and $\mathbf{B}(0) = \mathbf{0}$.
2: for $t = 1, 2, \ldots$
Perform sparse coding
3: $\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}} \frac{1}{2}||\mathbf{y}^{obs}(t) - \mathbf{O}(t)\hat{\mathbf{D}}(t-1)\mathbf{s}||_2^2 + \lambda_s||\mathbf{s}||_1 + \frac{1}{2}\lambda_L \mathbf{s}^{\mathcal{T}}\hat{\mathbf{D}}^{\mathcal{T}}(t-1)\mathbf{L}\hat{\mathbf{D}}(t-1)\mathbf{s}$
Perform prediction
4: Output $\hat{\boldsymbol{\pi}}(t) = \hat{\mathbf{D}}(t-1)\hat{\mathbf{s}}(t)$
Perform dictionary update
5: $\mathbf{A}(t) = \beta\mathbf{A}(t-1) + \hat{\mathbf{s}}(t)\hat{\mathbf{s}}^{\mathcal{T}}(t)$
6: $\mathbf{A}_m(t) = \beta\mathbf{A}_m(t-1) + \mathbb{1}_{\{m \in \mathcal{M}^{obs}(t)\}}\hat{\mathbf{s}}(t)\hat{\mathbf{s}}^{\mathcal{T}}(t)$ for $\forall m \in \mathcal{M}$
7: $\mathbf{B}(t) = \beta\mathbf{B}(t-1) + \mathbf{O}^{\mathcal{T}}(t)\mathbf{y}^{obs}(t)\hat{\mathbf{s}}^{\mathcal{T}}(t)$
8: Set $[\hat{\mathbf{d}}_1(t), \ldots, \hat{\mathbf{d}}_Q(t)] = \hat{\mathbf{D}}(t-1)$
9: Repeat
10: For $q = 1, 2, \ldots, Q$
11: Update $\hat{\mathbf{d}}_q(t)$ as (18)–(19)
12: Next q
13: Until convergence
14: Set $\hat{\mathbf{D}}(t) = [\hat{\mathbf{d}}_1(t), \ldots, \hat{\mathbf{d}}_Q(t)]$
15: Next t

TABLE II
AN ONLINE ALGORITHM FOR SPECTRUM PREDICTION.

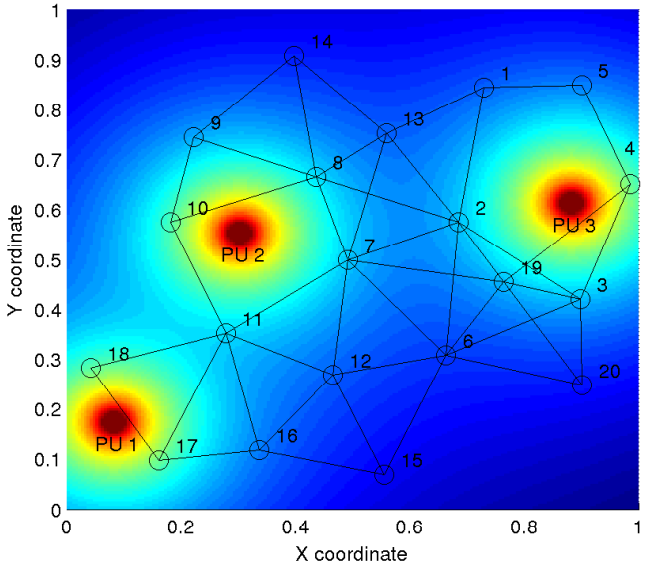

Fig. 1. CR network topology and PU transmitter locations.

where $\mathbb{1}_{\{\cdot\}}$ is an indicator function equal to 1 if the condition inside the braces are satisfied, and 0 otherwise.

Similar to the batch case in Sec. III-A, the dictionary update amounts to solving (12) for $\hat{\mathbf{D}}(t)$ with $\{\hat{\mathbf{s}}(\tau)\}_{\tau=1}^t$ fixed. Let $\hat{s}_j(\tau)$ denote the j-th entry of $\hat{\mathbf{s}}(\tau)$, and $A_{m,jq}(t)$ and $A_{jq}(t)$ the (j,q)-th entry of matrices $\mathbf{A}_m(t)$ and $\mathbf{A}(t)$, respectively. Also, let $\mathbf{b}_j(t)$ represent the j-th column of $\mathbf{B}(t)$. Then, upon defining

$$\boldsymbol{\Phi}_{j,q}(t) := \sum_{\tau=1}^{t} \beta^{t-\tau}\hat{s}_j(\tau)\hat{s}_q(\tau)(\mathbf{O}^{\mathcal{T}}(\tau)\mathbf{O}(\tau) + \lambda_L\mathbf{L}) \quad (16)$$

$$= \text{diag}([A_{1,jq}(t), A_{2,jq}(t), \ldots, A_{M,jq}(t)]) + \lambda_L A_{jq}(t)\mathbf{L} \quad (17)$$

the column-wise BCD leads to the following update for the j-th column of $\hat{\mathbf{D}}(t)$

$$\bar{\mathbf{d}}_j := \boldsymbol{\Phi}_{j,j}(t)^{-1}\left[\mathbf{b}_j(t) - \sum_{q=1, q\neq j}^{Q} \boldsymbol{\Phi}_{j,q}(t)\hat{\mathbf{d}}_q(t)\right] \quad (18)$$

$$\hat{\mathbf{d}}_j(t) = \frac{\bar{\mathbf{d}}_j}{\max\{||\bar{\mathbf{d}}_j||_2, 1\}}. \quad (19)$$

The overall algorithm for online spectrum prediction is given in Table II.

IV. SPATIO-TEMPORAL SPECTRUM PREDICTION

The algorithms developed so far provide imputations for missing measurements of spatial interference distributions, given the (incomplete) measurements of the *current* and the *past* time instants. However, in order to predict the *future* interference status, for which not even partial measurements can be available, one has to incorporate temporal correlation structures into the model, or learn such structures from the data. Spatio-temporal prediction using a Kalman filtering framework has been reported in the context of weather prediction, CR sensing, and network monitoring [16], [7], [17]. In this work, we leverage the data-driven dictionary learning framework to learn temporal dynamics from the data even in an online fashion.

The idea is to simply concatenate into a super-vector the observations over T consecutive intervals, and apply the algorithms developed in Sec. III. That is, define

$$\mathcal{Y}^{obs}(t) := [\mathbf{y}^{obs\,\mathcal{T}}(t), \ldots, \mathbf{y}^{obs\,\mathcal{T}}(t-T+1)]^{\mathcal{T}} \quad (20)$$

$$\mathcal{O}(t) := [\mathbf{O}^{\mathcal{T}}(t), \ldots, \mathbf{O}^{\mathcal{T}}(t-T+1)]^{\mathcal{T}} \quad (21)$$

$$\mathcal{L} := \mathbf{I}_T \otimes \mathbf{L} \quad (22)$$

where $\otimes$ denotes the Kronecker product, which are used in place of $\mathbf{y}^{obs}(t)$, $\mathbf{O}(t)$, and $\mathbf{L}$, respectively, in the algorithm of Table II.

To perform prediction for $\boldsymbol{\pi}(t+1)$, after executing line 14 in Table II, compute sparse coefficient $\hat{\mathbf{s}}^f(t+1)$ for a fictitious observation $\mathcal{Y}^{obs,f}(t+1) := [\mathbf{y}^{obs\,\mathcal{T}}(t), \ldots, \mathbf{y}^{obs\,\mathcal{T}}(t-T+2)]^{\mathcal{T}}$, assuming that the entire $\mathbf{y}^{obs}(t+1)$ is missing; i.e., $\mathcal{O}^f(t+1) := [\mathbf{O}^{\mathcal{T}}(t), \ldots, \mathbf{O}^{\mathcal{T}}(t-T+2)]^{\mathcal{T}}$. Then, $\hat{\boldsymbol{\pi}}(t+1)$ can be obtained as

$$\hat{\boldsymbol{\pi}}(t+1) = \hat{\mathbf{D}}(t)[1{:}M, :]\hat{\mathbf{s}}^f(t+1) \quad (23)$$

where $\hat{\mathbf{D}}(t)[1{:}M, :]$ denotes the first M rows of $\hat{\mathbf{D}}(t)$.

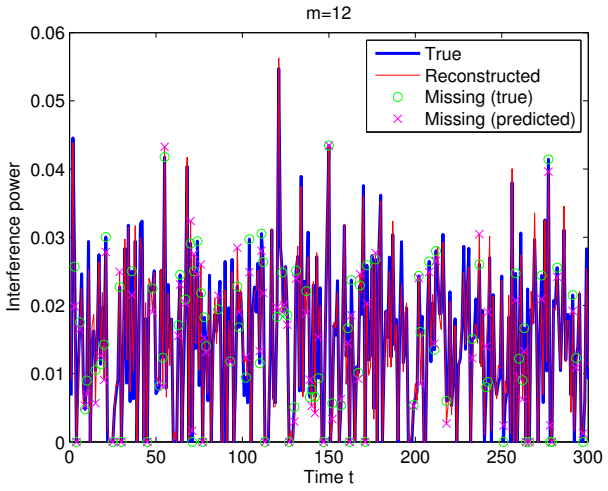

Fig. 2. Interference power using the batch algorithm.

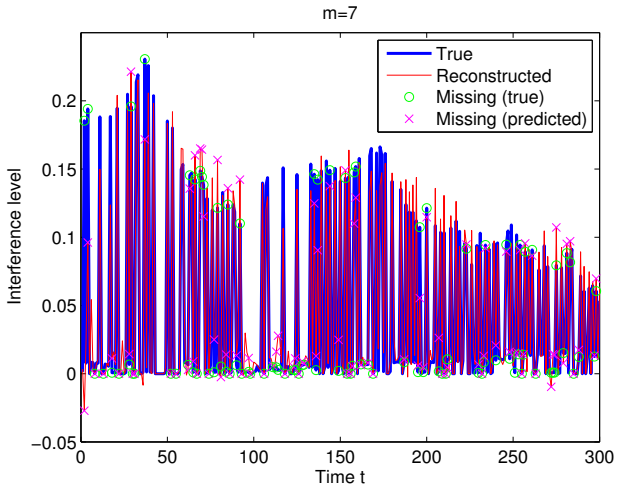

Fig. 3. Interference power using the online algorithm.

V. Numerical Tests

The performance of the proposed algorithms was evaluated via numerical tests. A CR network consisting of $M = 20$ nodes with the topology depicted in Fig. 1 is considered, where the circles denote the CR nodes and the lines represent the connections established among neighbors. The interference power distribution due to $K = 3$ PU transmitters is also depicted in Fig. 1, where the emitter locations are clearly revealed. The pathloss was computed as $\left(\frac{d}{d_0}\right)^{\alpha}$, where d was the distance, $d_0 = 0.01$ and $\alpha = 2.5$. The number of atoms of the dictionary was set to $Q = 50$.

First, the two-phase algorithm was tested. Each of the PUs turned on with a 30% chance, and transmitted at a power level $p_k(t)$ chosen from a uniform distribution with support $[100, 200]$. Each CR made a measurement with a 70% chance. The measurements were corrupted by additive noise generated from a zero-mean Gaussian distribution with variance 10^{-5}, which was then clipped to ensure non-negativity. The measurements were also normalized by the maximum amplitude observed in the training set. No shadowing or small-scale fading was considered, signifying a quasi-static scenario. $N = 300$ samples were used to train the dictionary, and then another 300 samples were supplied for the operational phase. The values of λ_s and λ_L were set to 0.1 and 0.005, respectively. To compensate for the bias inherent in Lasso-type estimators, de-biasing was performed in the operational phase; that is, after performing sparse coding in (9) to obtain $\hat{\mathbf{s}}(t)$, (9) was re-solved *without* the ℓ_1 regularization term *only for the non-zero entries* in $\hat{\mathbf{s}}(t)$. Fig. 2 shows the true interference level and the reconstructed one at CR $m = 12$ in the thick blue and the thin red curves, respectively. The missing (true) levels are denoted by the green circles, whose interpolations are marked by the magenta crosses. It can be seen that the missing entries are accurately recovered through the proposed method.

To test the online algorithm, the Rayleigh fading coefficient

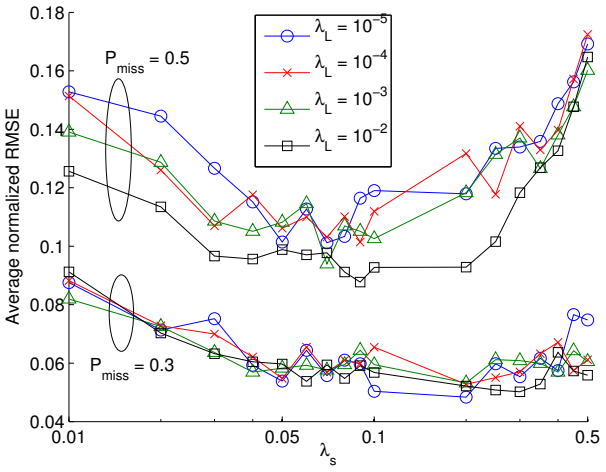

Fig. 4. Average normalized RMSE.

$h_{mk}(t)$ for the channel from PU k to CR m at time t was generated using a first-order autoregressive model

$$h_{mk}(t) = \alpha h_{mk}(t-1) + \sqrt{1-\alpha^2} w_{mk}(t) \qquad (24)$$

where $\alpha = 0.9995$ was used, and $w_{mk}(t)$ was circularly symmetric zero-mean complex Gaussian noise with variance 1. The overall channel gain $g_{mk}(t)$ was formed by multiplying the pathloss with $|h_{mk}(t)|^2$. The forgetting factor was $\beta = 0.95$, and $\lambda_s = 0.25$ and $\lambda_L = 0.005$ were used. The transmit-power of all PUs was fixed to 150 so that the tracking performance could be clearly visible. Fig. 3 shows the interference level for CR 7, where it is evident that the online algorithm tracks the slow variation of interference levels due to channel fading. Also, it is noted that the initial transient for the online learning is quite short.

The normalized root mean square error (RMSE) for the missing observations, averaged over 20 CRs, is depicted in

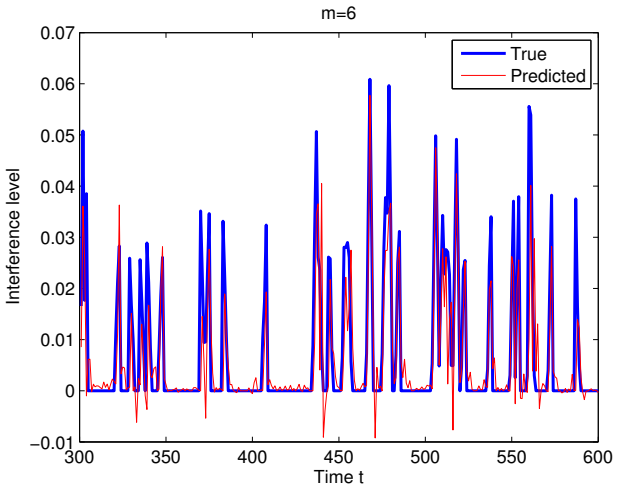

Fig. 5. One-time slot prediction.

Fig. 4 for various values of λ_s and λ_L. The online algorithm was again used. Two sets of curves are presented, corresponding to the chance of missing observations P_{miss} equal to 0.3 and 0.5. The curves reveal that optimal values for λ_s and λ_L exist. It can be deduced that the Laplacian-based regularization becomes more important when a larger fraction of observations are unavailable.

In order to test the temporal prediction, certain traffic patterns were assumed. That is, at each time interval t, PU 1 tossed a coin and transmitted with probability 0.1. If PU 1 did transmit, PU 2 transmitted in the next time slot, followed by PU 3's transmission in the third time slot. Likewise, at each time t, PU 3 started transmission with probability 0.15, followed by PU 2 in the second time slot, and PU 1 in the third. Fig. 5 shows the result of the one-time slot-ahead prediction of the interference power at CR 6 using the online algorithm with $T = 4$. No missing measurements were assumed. It can be seen that the traffic patterns are successfully acquired by dictionary learning to predict future interference levels.

VI. Conclusions

Spectrum prediction algorithms for CR networks have been developed. Using a dictionary learning framework, the algorithms can predict the interference power experienced at each CR node based on the current and the past measurements collected from a *subset* of nodes in the network. Exploiting the fact that the spatial variation of interference is smooth, a regularization term based on the CR network topology was also incorporated. Batch and online algorithms were derived, where the online alternative possessed a tracking capability at lower complexity and memory requirements. Temporal prediction was also discussed. Numerical tests verified the efficacy of our novel schemes. Experiments with real datasets will be considered in a future work.

References

[1] FCC Spectrum Policy Task Force, "Report of the spectrum efficiency working group," Nov. 2002. [Online]. Available: http://www.fcc.gov/sptf/reports.html

[2] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, Sep. 2006.

[3] S.-J. Kim, E. Dall'Anese, J. A. Bazerque, K. Rajawat, and G. B. Giannakis, "Advances in spectrum sensing and cross-layer design for cognitive radio networks," in *E-Reference Signal Processing*. EURASIP, 2013, in press. [Online]. Available: http://www.dtc.umn.edu/s/resources/eurasip2012tut.pdf

[4] E. Visotsky, S. Kuffner, and R. Peterson, "On collaborative detection of TV transmissions in support of dynamic spectrum sharing," in *Proc. of the DySPAN Conf.*, Baltimore, MD, Nov. 2005, pp. 338–345.

[5] M. Wellens, J. Riihijärvi, M. Gordziel, and P. Mähönen, "Spatial statistics of spectrum usage: From measurements to spectrum models," in *Proc. of the ICC Conf.*, Dresden, Germany, Jun. 2009, pp. 1–6.

[6] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. Sig. Proc.*, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.

[7] 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. Proc.*, vol. 5, pp. 24–36, Feb. 2011.

[8] S. Yarkan and H. Arslan, "Binary time series approach to spectrum prediction for cognitive radio," in *Proc. of the IEEE Veh. Technol. Conf.*, Baltimore, MD, Oct. 2007, pp. 1563–1567.

[9] X. Li and S. A. Zekavat, "Traffic pattern prediction and performance investigation for cognitive radio systems," in *Proc. of the IEEE Wireless Commun. and Net. Conf.*, Apr. 2008, pp. 894–899.

[10] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 589–600, Apr. 2007.

[11] I. Tošić and P. Frossard, "Dictionary learning," *IEEE Sig Proc. Mag.*, vol. 28, no. 2, pp. 27–38, Mar. 2011.

[12] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Dec. 2006.

[13] S.-J. Kim, N. Jain, G. Giannakis, and P. Forero, "Joint link learning and cognitive radio sensing," in *Proc. of the 45th Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2011.

[14] P. A. Forero, K. Rajawat, and G. B. Giannakis, "Semi-supervised dictionary learning for network-wide link load prediction," in *Proc. of the 3rd Intl. Workshop on Cognitive Info. Processing (CIP)*, Parador de Baiona, Spain, May 2012.

[15] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, 2010.

[16] C. K. Wikle and N. Cressie, "A dimension-reduced approach to space-time Kalman filtering," *Biometrika*, vol. 86, no. 4, pp. 815–829, 1999.

[17] K. Rajawat, E. Dall'Anese, and G. B. Giannakis, "Dynamic network delay cartography," *submitted to IEEE Trans. Info. Theory*, 2012. [Online]. Available: http://arxiv.org/abs/1204.5507