JOINT LINK LEARNING AND COGNITIVE RADIO SENSING

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Nitin Jain

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Prof. Georgios B. Giannakis

June, 2012

Acknowledgements

First of all, I would like to thank Prof. Georgios Giannakis for his continuous guidance throughout my thesis and providing invaluable knowledge through courses. Beside my advisor, I would also want to thank my collaborator Dr. Seung Jun Kim for his technical inputs, insightful comments and discussion sessions. This thesis would not have been possible without your continuous guidance.

Due thanks to Prof. Daniel Boley and Prof. Mostafa Kaveh for agreeing to serve as members in my thesis committee.

I am grateful to Prof. Tom Luo, Prof. Nihar Jindal and Prof. Yousef Saad for their courses, which helped me acquiring better understanding in my area of interest.

I would also like to express my gratitude towards Yu Zhang, Morteza Mardani, Swayambhoo Jain, Dr. Nikolaos Gatsis and other SPiNCOM members for fruitful discussions and help during my thesis. I also thank them for making my stay pleasant at University of Minnesota. I would also like to thank Prof. P.R.K Rao, professor in my undergraduate school, for developing my interest in Communications and Signal processing. I thank my friends, group *Yo Buddy*, for supporting me continuously.

Finally, I thank my parents, my sister Nidhi Jain and my *mausi* Neetu Jain for unconditional love, support and encouragement throughout my life.

Abstract

In this thesis, novel cooperative spectrum sensing algorithms for cognitive radios (CRs) are developed, which can blindly learn the channel gains between CRs and licensed primary users (PUs), while jointly detecting active PU transmitters at each time instant. A dictionary learning approach is taken to decompose the received signal energy samples per CR into linear combinations of channel gains and PU transmit-powers, up to scaling ambiguity. In addition to a iterative batch baseline algorithm, an efficient online implementation that can track slow variation of channel gains with reduced computational complexity is developed, as well as a distributed alternative, which requires only local message passing among neighbors in CR networks. Two approaches for selecting the sparsity parameter in the batch, online and distributed learning cases are also developed. In order to remove scaling ambiguity from the columns of the channel gain matrix, an assumption that the PU transmit-powers take values from the known set of finite levels is made. Again, dictionary learning approach is used and batch and online algorithms are developed. We have shown through numerical results that recovery of channel gains and PU transmit-powers is possible.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Why cognitive radios?

The cognitive radio (CR) paradigm aims at mitigating the increasingly strenuous spectrum scarcity for wireless communication. The rationale is that although most of the spectrum has been allotted to primary users (PUs) that possess exclusive usage rights, the bands are often significantly underutilized depending on the time and the place of operation [1]. Fig. 1.1 shows the occupancy of frequency during a day time at Southwark in UK. It is clear from the Fig. 1.1 that only $15\% - 20\%$ of overall spectrum is used throughout the day. This wastage of resources stimulated interest in concept of cognitive radio, in which users adapt to the operating environment, including existing spectrum usage [2]. Cognitive radio can be identified as

- A fully cognitive radio also known as Mitola Radio, is a radio that is aware of all its surroundings and adapts all possible wireless parameter i.e. modulation format, multiple access technique, bandwidth, center frequency etc. It requires flexible RF front end capable of swapping frequency bands. Also, baseband processing should be adaptive whose ADC and DACs have enormous dynamic range and can operate over wide range of frequencies. With the advancement in technology, fully cognitive radio will be achieved but currently seems too complicated in practice.
- Spectrum sensing cognitive radio also known as Dynamic spectrum access that adapts the transmission frequency and senses RF spectrum.

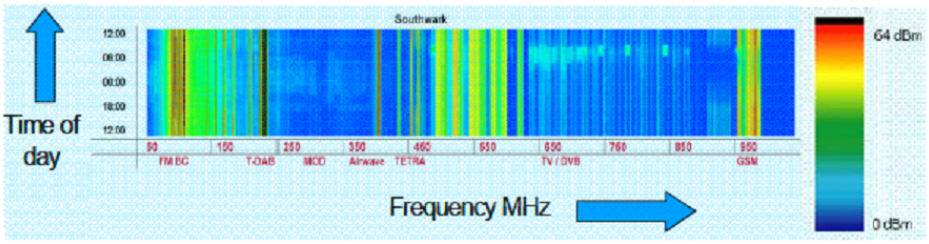

Figure 1.1: Spectrum utilization.
Source:`http://stakeholders.ofcom.org.uk/binaries/research/technology-research/cograd_summary.pdf`

The key tasks are to identify unused spectral resources—often called "spectrum holes" or "white space"—in the time, frequency, and space domains. Then perform dynamic resource allocation for CRs to allocate their transmit parameter such as transmit power, bit-rate, bandwidth etc. based on the available channel state information (CSI) of the PUs and CRs without causing harmful interference to the incumbent PU system. To achieve this goal, cognitive radio consists of cognizant receiver capable of sensing, a agile transmitter which can adapt to the parameters of the operating environment and intelligent dynamic resource allocation which can allocate resources decisively.

1.2 Spectrum sensing

The basic spectrum sensing task is to detect the presence of PU signals in a given frequency band at a given time instant. Such information is useful in spectrum *overlay* scenarios, where CRs are allowed to transmit only on a band completely unoccupied [3]. More sophisticated sensing algorithms can estimate the channel gains, transmit-powers, as well as PU locations. These are valuable in spectrum *underlay* scenarios, where CRs can share bands with PUs as long as the interference inflicted to PUs is kept below a tolerable level.

Spectrum sensing is challenging because prior knowledge on PU signal characteristics may not be available, and PU systems typically do not cooperate for CR sensing. Moreover, fading and shadowing effects make reliable detection difficult. To mitigate these hurdles, multiple CRs can collaborate to perform cooperative spectrum sensing

[4], [5]. The basic idea of co-operative spectrum sensing is to improve sensing performance by exploiting the spatial diversity provided by different propagation times and fading present in the CR observations. Each CR share their sensing information in order to make more reliable combined decision. Co-operative spectrum sensing can also detect hidden terminals and, reduces sensing times and amount of local processing at each CR. The downside is the increased co-operation overhead. A typical assumption is that although CRs are spread over a geographical area to benefit from diversity of reception, they are still close to each other compared to their distances to the PUs such that all CRs observe essentially the same PU activity [6].

1.3 Prior work

More aggressive *spatial* spectrum reuse techniques, where CRs are allowed to transmit as long as Signal-to-interference-plus-noise-ratio (SINR) requirements of PU receivers are satisfied, is possible by departing from such an assumption. RF cartography is a technique that maps out the RF environment in which CR networks operate. Maps of power spectral density provide information as to which regions in space are less "crowded" in terms of interference, thus yielding more transmission opportunities for CRs [7, 8]. However, RF cartography cannot cope with the time varying channel gains.

In [9, 10] Channel gain maps were developed which can detect presence of PU as well as determine the interference caused by CRs to protect PU receivers. Map tracking algorithms using Kriged Kalman filtering were developed in [9] and [10]. The algorithms could interpolate channel gains over the entire space from the measurements made at CR positions. To achieve this, however, the CRs must exchange training sequences to obtain sample measurements of channel gains.

1.4 Our contribution

In the present work, we aim to learn the channel gains *blindly* i.e. without any training sequences or other information from PU, while at the same time detecting the activities of PUs, through cooperation of a network of CRs. Online implementations will permit tracking slowly varying channels and also reduces computational complexity.

Distributed algorithm is also developed, which can track slowly varying channel gains via local message passing [11]. Later with assumption of PU transmit power taking values from finite alphabet, batch and online algorithms are developed to recover channel gains and PU transmit powers exactly. Finally, techniques to select sparsity parameter λ for batch, online and distributed algorithms is developed. This work will be submitted in journal version of [11].

To these ends, a dictionary learning approach is taken. Dictionary learning decomposes a signal into a linear combination of a few atoms from a dictionary, which is jointly learned. Since the seminal work by Olshausen and Field applied this technique to understand human vision [12], it has been used successfully in a number of areas including image denoising, representation of audio data, classification, and clustering [13, 14, 15]. Efficient algorithms have also been developed, ranging from a Lagrange dual-based one [16], to a generalization of the K-means algorithm [17], and an online version [18].

1.5 Organization

The rest of the thesis is organized as follows. In Chapter 2, sec. 2.1 describes the system model and formulate the joint link learning and CR sensing problem. Sec. 2.2 provides the batch and online solutions. Distributed algorithm is developed in sec. 2.3. Sec. 2.4 offers technique for selecting sparsity parameter for batch, online and distributed algorithms. To avoid scaling ambiguity in dictionary, batch and online algorithm with finite alphabet assumption are developed in sec. 2.5.

Chapter 3 presents results of numerical tests with sec. 3.1 and sec. 3.3 presents performance of batch, online and distributed algorithm for both continuous and finite alphabet case respectively. Sec. 3.2 verify the approaches developed for sparsity parameter. Concluding remarks and future work are offered in chapter 4.

Chapter 2

Problem Formulation

2.1 System Model and Problem Statement

Consider a CR network consisting of M nodes that collaborate to detect the presence of active PUs in the deployment area. Assume that there are K PUs with the k-th one transmitting at power level $p_k(t)$ at time t, for $k = 1, 2, \ldots, K$. The m-th CR takes a measurement $\pi_m(t)$ at time t of the RF power spectrum in the narrow frequency band of interest. Let $g_{mk}(t)$ denote the channel gain at time t between the m-th CR and the k-th PU. Due to the lack of explicit support from the PU system on channel estimation, it is often challenging for CRs to acquire the channels accurately. Thus, knowledge of $g_{mk}(t)$ is not assumed; rather, it will be learned from the measurements, while jointly detecting active PU transmitters.

The measurement $\pi_m(t)$ can be modeled as

$$\pi_m(t) = \sum_{k=1}^{K} g_{mk}(t)p_k(t) + e_m(t), \quad m = 1, 2, \ldots, M \tag{2.1}$$

where $e_m(t)$ is the measurement error with power σ^2. Define matrix $\mathbf{G}(t) \in \mathbb{R}_+^{M\times K}$ with $g_{mk}(t)$ as its (m,k)-th element. Define $\boldsymbol{\pi}(t) \triangleq [\pi_1(t), \ldots, \pi_M(t)]^T$ and $\mathbf{p}(t) \triangleq [p_1(t), \ldots, p_K(t)]^T$. Vector $\mathbf{e}(t)$ is defined likewise. Then, (2.1) can be compactly written as

$$\boldsymbol{\pi}(t) = \mathbf{G}(t)\mathbf{p}(t) + \mathbf{e}(t). \tag{2.2}$$

Since oftentimes not all PU transmitters will transmit simultaneously, many of the entries of vector $\mathbf{p}(t)$ will be zero. This is especially true when the CR network does not have prior knowledge as to how many potential PU transmitters are present. In this case, K must be set to an upper-bound to the actual number K_0 of the PUs, leading to an even sparser $\mathbf{p}(t)$.

Supposing that the PU activity changes fast compared to the measurement interval, $\mathbf{p}(t)$ will tend to be statistically independent across time. On the other hand, it is assumed that the channel changes slowly relative to the measurement interval. To be specific, suppose for now that the channel gains do not change over T sampling intervals; that is, $\mathbf{G}(t) = \mathbf{G}$ for $t = 1, 2, \ldots, T$. Then, by accumulating the measurements of all CRs over time $t = 1, 2, \ldots, T$ into matrix $\mathbf{\Pi} \triangleq [\boldsymbol{\pi}(1), \boldsymbol{\pi}(2), \ldots, \boldsymbol{\pi}(T)] \in \mathbb{R}_+^{M \times T}$, and likewise defining $\mathbf{P} \triangleq [\mathbf{p}(1), \ldots, \mathbf{p}(T)] \in \mathbb{R}_+^{K \times T}$ as well as $\mathbf{E}$, one obtains

$$\mathbf{\Pi} = \mathbf{GP} + \mathbf{E}. \tag{2.3}$$

Given $\mathbf{\Pi}$, our goal is to learn the channel matrix $\mathbf{G}$ and determine the sparse matrix $\mathbf{P}$. Since column-wise scaling in $\mathbf{G}$ and the corresponding inverse scaling applied to the rows of $\mathbf{P}$ do not alter the product $\mathbf{GP}$, learning only the relative magnitudes of the channels and the transmit-powers is possible, which is a limitation inherent to any blind algorithm. Also, $\mathbf{GM}^{-1}$ and $\mathbf{MP}$, $\mathbf{M}$ is any square matrix, can be also viewed as a solution. But as a solution $\mathbf{MP}$ should be sparse and have non-negative entries, and also $\mathbf{GM}^{-1}$ should have non-negative entries. These conditions requires $\mathbf{M}$ to be sparse and $\mathbf{M}$, $\mathbf{M}^{-1}$ to have most of entries as non-negative, which rules out possiblity of $\mathbf{M}$ becoming anything except diagonal matrices.

To accomplish this, a joint dictionary learning and sparse regression formulation is considered. First, define set $\mathcal{G}$ as

$$\mathcal{G} \triangleq \{[\mathbf{g}_1, \ldots, \mathbf{g}_K] \geq \mathbf{0} : \|\mathbf{g}_k\|_2 \leq 1 \text{ for } k = 1, \ldots, K\} \tag{2.4}$$

where the nonnegativity constraints are applied element-wise, and $\{\mathbf{g}_k\}$ are column vectors. Then, the optimization problem is posed as

$$\min_{\mathbf{G} \in \mathcal{G}, \mathbf{P} \geq \mathbf{0}, \boldsymbol{\nu} \geq \mathbf{0}} \frac{1}{2} \|\mathbf{\Pi} - \mathbf{GP} - \mathbf{1}^T \otimes \boldsymbol{\nu}\|_F^2 + \lambda \|\mathbf{P}\|_1 \tag{2.5}$$

where the constraint $\mathbf{G} \in \mathcal{G}$ prevents the columns of $\mathbf{G}$ growing arbitrarily large, in addition to enforcing nonnegativity, $\|\mathbf{P}\|_1$ is the sum of the ℓ_1-norms of the columns of

$\mathbf{P}$, which is a penalty term promoting sparsity in $\mathbf{P}$, and $\boldsymbol{\nu}$ compensates for non-negative mean of noise. Since each entry in $\mathbf{P}$ is nonnegative, $\|\mathbf{P}\|_1$ is simply equal to the sum of all the entries in $\mathbf{P}$. Parameter λ tunes the sparsity level of $\mathbf{P}$.

Instead of the batch setup considered in (2.5), an online formulation may be favored when the channel gains are continuously time-varying, albeit still slowly compared to the measurement interval. The online solver will then be able to track the channels. Moreover, the computation may be done recursively as new measurements are obtained over time, thus considerably reducing the computational burden. For this, consider solving at each time t

$$\min_{\substack{\mathbf{G}\in\mathcal{G},\\ \mathbf{p}(t)\geq\mathbf{0},\boldsymbol{\nu}\geq\mathbf{0}}} \sum_{s=1}^{t} \beta^{t-s}\left(\frac{1}{2}\|\boldsymbol{\pi}(s)-\mathbf{G}\mathbf{p}(s)-\boldsymbol{\nu}\|_2^2+\lambda\|\mathbf{p}(s)\|_1\right) \tag{2.6}$$

where the weights β^{t-s} with $\beta\in(0,1]$ are used to gradually forget the past measurements while weighting the recent measurements more heavily.

2.2 Joint Link Learning and Sensing Algorithms

2.2.1 Introduction to Dictionary learning

In recent years, there has been a lot of interest in obtaining the sparse representation of signal. This kind of representation is helpful in compression, feature extraction, denoising etc. It has been established that using a overcomplete dictionary $\mathbf{G}$, which has more number of atoms than the dimension of signal, signal $\boldsymbol{\pi}(t)$ can be represented using few atoms of dictionary $\mathbf{G}$. This can be visualized as solving problem (2.7) known as sparse coding, where dictionary $\mathbf{G}$ is assumed to be known.

$$\mathbf{p}(t)=\arg\min_{\mathbf{p}}\frac{1}{2}\|\boldsymbol{\pi}(t)-\mathbf{G}\mathbf{p}\|_2^2+\lambda\|\mathbf{p}\|_0 \tag{2.7}$$

ℓ_0-norm penalize the number of non-zero entries and forces solution to be sparse. The problem in (2.7) is combinatorially complex, so ℓ_0-norm is replaced by ℓ_1-norm, which makes the problem in (2.8) convex.

$$\mathbf{p}(t)=\arg\min_{\mathbf{p}}\frac{1}{2}\|\boldsymbol{\pi}(t)-\mathbf{G}\mathbf{p}\|_2^2+\lambda\|\mathbf{p}\|_1 \tag{2.8}$$

Penalty parameter of ℓ_1-norm also promotes sparse solution. It is established in compressive sensing literature [19] that under some conditions the solution of (2.7) and (2.8) are same. There are many known methods in literature to solve (2.8) based on co-ordinate descent with soft thresholding.

Dictionary $\mathbf{G}$ in (2.8) can be prespecified transform matrix such as fourier transform, wavelets etc or can be trained using the data. Choosing a prespecified dictionary may provide simple and fast algorithm but not necessary a good sparse representation. This has developed interest in learning dictionary based on multiple observation of data.

Given the multiple observation of data $\mathbf{\Pi} = [\boldsymbol{\pi}(1), \boldsymbol{\pi}(2), \ldots]$, the goal is to estimate the dictionary $\mathbf{G}$ and sparse matrix $\mathbf{P}$ and is given by (2.9)

$$\min_{\mathbf{G},\mathbf{P}} \frac{1}{2}\|\mathbf{\Pi} - \mathbf{G}\mathbf{P}\|_F^2 + \lambda\|\mathbf{P}\|_1 \tag{2.9}$$

Various algorithms have been developed for the joint dictionary learning and sparse coding problem (2.9). The above problem is similar to the optimization problem posed in (2.5). A small but important difference of our setup from the most common ones is the nonnegativity constraints imposed on the entries of $\mathbf{G}$, $\mathbf{P}$ and inclusion of $\boldsymbol{\nu}$. In [16], the ℓ_2-constrained dictionary learning problem was tackled in the dual domain. The K-SVD algorithm was developed in [17], which generalized the K-means clustering algorithm to the dictionary learning formulation. However, it is not straightforward to extend these algorithms to account for the nonnegativity constraints.

Problems (2.5) and (2.6) are nonconvex. Therefore, attaining the global optimum may be difficult. However, when $\mathbf{G}$, $\boldsymbol{\nu}$ is fixed, optimizing over $\mathbf{P}$ involves convex optimization. Likewise, when $\mathbf{P}$ is fixed, the problems for $\mathbf{G}$, $\boldsymbol{\nu}$ become convex. Therefore, alternating minimization is employed to obtain locally optimal solutions. Next, both batch and online algorithms are developed.

2.2.2 Batch Algorithm

In this thesis, the block coordinate descent method also used in [18] is employed for learning $\mathbf{G}$, $\boldsymbol{\nu}$ given $\mathbf{P}$. In Table 2.1, the batch algorithm is presented. $\mathbf{G}_0$ is chosen randomly from $\mathbb{R}_+^{M\times K}$ but not all zero columns. Approach for selecting λ is discussed in sec. 2.4. Lines 3–6 yield the sparse vector $\mathbf{p}(t)$ for each $t = 1, 2, \ldots, T$ given the dictionary from the previous iteration. This can be solved, e.g., using the LARS algorithm [20]. Lines

Input: $\mathbf{\Pi}$, initial dictionary $\mathbf{G}_0$, and λ
Output: $\mathbf{G}$ and $\mathbf{P}$

1: Set $\mathbf{G} \triangleq [\mathbf{g}_1, \ldots, \mathbf{g}_K] = \mathbf{G}_0$, $\mathbf{c} = \mathbf{\Pi 1^T}$ and $\boldsymbol{\nu} = \mathbf{0}$
2: Repeat
3: Perform sparse regression with fixed $\mathbf{G}$, $\boldsymbol{\nu}$.
4: For $t = 1, 2, \ldots, T$
5: $\mathbf{p}(t) = \arg\min_{\mathbf{p} \geq \mathbf{0}} \frac{1}{2}\|\boldsymbol{\pi}(t) - \boldsymbol{\nu} - \mathbf{G}\mathbf{p}\|_2^2 + \lambda\|\mathbf{p}\|_1$
6: Next t
7: Perform dictionary update with fixed $\mathbf{P}$.
8: Compute $\mathbf{A} = \mathbf{P}\mathbf{P}^T$, $\mathbf{B} = \mathbf{\Pi}\mathbf{P}^T$, $\mathbf{d} = \mathbf{P1^T}$
9: Repeat
10: $\tilde{\mathbf{B}} = \mathbf{B} - \boldsymbol{\nu}\mathbf{d}^T$
11: For $k = 1, 2, \ldots, K$
12: $\tilde{\mathbf{g}}_k = \frac{1}{A_{kk}}\left(\tilde{\mathbf{b}}_k - \mathbf{G}\mathbf{a}_k\right) + \mathbf{g}_k$
13: $\mathbf{g}_k = \frac{[\tilde{\mathbf{g}}_k]^+}{\max\{\|[\tilde{\mathbf{g}}_k]^+\|_2, 1\}}$
14: Next k
15: $\boldsymbol{\nu} = \left[\frac{\mathbf{c} - \mathbf{G}\mathbf{d}}{T}\right]^+$
16: until convergence
17: until convergence

Table 2.1: Batch algorithm.

7–16 find the $\mathbf{G}$, $\boldsymbol{\nu}$ that minimizes the quadratic cost $\|\mathbf{\Pi} - \mathbf{G}\mathbf{P} - \mathbf{1^T} \otimes \boldsymbol{\nu}\|_{\mathbf{F}}^{\mathbf{2}}$ subject to $\mathbf{G} \in \mathcal{G}$ and $\boldsymbol{\nu} \geq \mathbf{0}$, given $\mathbf{P}$. Based on the block coordinate descent method [21, Sec. 2.7], the algorithm sequentially updates the individual columns of $\mathbf{G}$ with the other ones and $\boldsymbol{\nu}$ fixed and updates $\boldsymbol{\nu}$ with $\mathbf{G}$ fixed. Detailed derivation is given in appendix A.2. Line 12 solves the unconstrained minimization with respect to the k-th column of $\mathbf{G}$, followed by the projection onto the nonnegative portion of the unit ℓ_2-ball in line 13. Here, $[\cdot]^+$ represents $\max\{0, \cdot\}$. Also, A_{kk} denotes the k-th diagonal element of matrix $\mathbf{A} \triangleq \mathbf{P}\mathbf{P}^T$, and $\mathbf{a}_k$ and $\tilde{\mathbf{b}}_k$ represent the k-th columns of $\mathbf{A}$ and $\tilde{\mathbf{B}}$, respectively.

2.2.3 Online Algorithm

For online link learning and spectrum sensing, (2.6) is solved. With a slight modification to account for the forgetting factor β and the nonnegativity constraint, the algorithm in [18] can be adapted as summarized in Table 2.2.

The key difference of the online algorithm from the batch one is that the sparse

Input: $\boldsymbol{\pi}(t)$ for $t = 1, 2, \ldots$, initial dictionary $\mathbf{G}(0)$, and λ
Output: $\mathbf{G}(t)$ and $\mathbf{p}(t)$ for $t = 1, 2, \ldots$

1: Set $\mathbf{A}(0) = \mathbf{B}(0) = \mathbf{0}$ and $\mathbf{c}(0) = \mathbf{d}(0) = \boldsymbol{\nu}(0) = \mathbf{0}$.
2: For $t = 1, 2, \ldots$
3: Perform sparse regression.
4: $\mathbf{p}(t) = \arg\min_{\mathbf{p}\geq\mathbf{0}} \frac{1}{2}\|\boldsymbol{\pi}(t) - \boldsymbol{\nu}(t-1) - \mathbf{G}(t-1)\mathbf{p}\|_2^2 + \lambda\|\mathbf{p}\|_1$
5: Perform dictionary update.
6: Compute:
7: $\mathbf{A}(t) = \beta\mathbf{A}(t-1) + \mathbf{p}(t)\mathbf{p}(t)^T$
8: $\mathbf{B}(t) = \beta\mathbf{B}(t-1) + \boldsymbol{\pi}(t)\mathbf{p}(t)^T$
9: $\mathbf{c}(t) = \beta\mathbf{c}(t-1) + \boldsymbol{\pi}(t)$
10: $\mathbf{d}(t) = \beta\mathbf{d}(t-1) + \mathbf{p}(t)$
11: Set $\mathbf{G} \triangleq [\mathbf{g}_1, \ldots, \mathbf{g}_K] = \mathbf{G}(t-1)$ and $\boldsymbol{\nu} = \boldsymbol{\nu}(t-1)$(warm start)
12: Repeat
13: $\tilde{\mathbf{B}}(t) = \mathbf{B}(t) - \boldsymbol{\nu}\mathbf{d}(t)^T$
14: For $k = 1, 2, \ldots, K$
15: $\tilde{\mathbf{g}}_k = \frac{1}{A_{kk}(t)}\left[\tilde{\mathbf{b}}_k(t) - \mathbf{G}\mathbf{a}_k(t)\right] + \mathbf{g}_k$
16: $\mathbf{g}_k = \frac{[\tilde{\mathbf{g}}_k]^+}{\max\{\|[\tilde{\mathbf{g}}_k]^+\|_2, 1\}}$
17: Next k
18: $\boldsymbol{\nu} = \frac{1-\beta}{1-\beta^t}\left[\mathbf{c}(t) - \mathbf{G}\mathbf{d}(t)\right]^+$
19: until convergence
20: (Optional: replace all-zero columns in $\mathbf{G}$ to random vectors.)
21: Set $\mathbf{G}(t) = \mathbf{G}$ and $\boldsymbol{\nu}(t) = \boldsymbol{\nu}$
22: Next t

Table 2.2: Online algorithm.

coding problem is solved only for the "current" power vector $\mathbf{p}(t)$; the past solutions $\mathbf{p}(t-1), \ldots, \mathbf{p}(1)$ are unchanged. Still, it was shown in [18] that under mild conditions, $\mathbf{G}(t)$ so obtained converges to the same $\mathbf{G}$ that would have been obtained by updating the entire $\mathbf{p}(1), \ldots, \mathbf{p}(t)$, as $t \to \infty$. Since the past power vectors do not change, matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{c}$ and $\mathbf{d}$ can be updated recursively as shown in lines 6–10 of Table 2.2. Recursive update is helpful as we do not need to store all $\mathbf{p}(t)$ and can avoid big matrix multiplications as in batch algorithm. Sometimes some columns of $\mathbf{G}$ become zeros, degrading performance. Line 20 replaces the degenerate columns with nonzero random columns.

2.3 Distributed Online Algorithm

Input: $\{\pi_m(t)\}_{m=1}^M$ for $t = 1, 2, \ldots$, initial dictionary $\{\bar{\mathbf{g}}_m(0)\}_{m=1}^M$, and λ
Output: $\{\bar{\mathbf{g}}_m(t)\}_{m=1}^M$ and $\mathbf{p}_1(t) = \mathbf{p}_2(t) = \ldots = \mathbf{p}_M(t)$ for $t = 1, 2, \ldots$

Set $\mathbf{A}_m(0) = \mathbf{0}$, $\mathbf{b}_m(0) = \mathbf{d}_m(0) = \mathbf{0}$ and $c_m(0) = 0$ for all m.
For $t = 1, 2, \ldots$
    Perform distributed sparse regression.
        Initialize $\boldsymbol{\zeta}_m^{(0)}$, $\boldsymbol{\kappa}_m^{(0)}$, $\mathbf{p}_m^{(1)}$ and $\mathbf{q}_m^{(1)}$ arbitrarily for all m (e.g., warm start)
        For $i = 1, 2, \ldots,$MAX_ITER
            Perform updates in (2.11)–(2.14) at each CR $m \in \{1, 2, \ldots, M\}$
        Next i
    Perform distributed dictionary update.
        At each CR $m \in \{1, 2, \ldots, M\}$ in parallel
            Compute:
                $\mathbf{A}_m(t) = \beta \mathbf{A}_m(t-1) + \eta \mathbf{I} + \mathbf{p}_m(t)\mathbf{p}_m(t)^T$
                $\mathbf{b}_m(t) = \beta \mathbf{b}_m(t-1) + \mathbf{p}_m(t)\pi_m(t)$
                $c_m(t) = \beta c_m(t-1) + \pi_m(t)$
                $\mathbf{d}(t) = \beta \mathbf{d}(t-1) + \mathbf{p}(t)$
            Set $\bar{\mathbf{g}}_m \triangleq [g_{m1}, \ldots, g_{mK}] = \bar{\mathbf{g}}_m(t-1)$ and $\nu_m = \nu_m(t-1)$ (warm start).
            Repeat
                For $k = 1, 2, \ldots, K$
                    $\tilde{\mathbf{b}}_m(t) = \mathbf{b}_m(t) - \nu_m \mathbf{d}(t)^T$
                    $g_{mk} = \left[\frac{1}{A_{m,kk}(t)}\left(\tilde{b}_{mk}(t) - \bar{\mathbf{g}}_m \mathbf{a}_k(t)\right) + g_{mk}\right]^+$
                Next k
                $\nu_m = \frac{1-\beta}{1-\beta^t}\left[c_m(t) - \bar{\mathbf{g}}_m \mathbf{d}(t)\right]^+$
            until convergence
            (Optional: if the k-th entry of $\mathbf{p}_m(t)$ has been zero for consecutive MAX_UNUSED times, replace it with a random number, for $k = 1, \ldots, K$)
Next t

Table 2.3: Distributed online algorithm.

The algorithms put forth in the preceding section require a fusion center that collects the measurements from CR nodes to perform dictionary learning in a centralized fashion. In practice, a distributed implementation may be desired, e.g., to avoid the single point of failure, ensure scalability of the algorithm, and reduce the feedback overhead.

Thus, consensus-based optimization over a multi-hop network is employed to obtain a *distributed* online dictionary learning algorithm [22].

One hurdle in deriving a distributed algorithm to solve (2.6) is the constraint $\mathbf{G} \in \mathcal{G}$. Since this constraint couples all CRs, a naive implementation requires centralized processing. To bypass this issue, the following alternative formulation is considered; see also [23]

$$\min_{\substack{\mathbf{G}\geq\mathbf{0},\mathbf{p}(t)\geq\mathbf{0},\\ \boldsymbol{\nu}\geq\mathbf{0}}} \sum_{s=1}^{t}\beta^{t-s}\left[\frac{1}{2}\left(\|\boldsymbol{\pi}(s)-\mathbf{G}\mathbf{p}(s)-\boldsymbol{\nu}\|_2^2+\eta\|\mathbf{G}\|_F^2\right)+\lambda\|\mathbf{p}(s)\|_1\right] \tag{2.10}$$

where the extra ℓ_2 regularization term on $\mathbf{G}$ prevents the entries of $\mathbf{G}$ from growing without bound.

Similar to the centralized algorithms, the distributed one works by alternating two steps: distributed sparse regression and distributed dictionary update. The goal of the distributed sparse regression step is to solve the optimization problem in line 4 of Table 2.2 through in-network processing. That is, it is desired that each CR $m \in \{1, 2, \ldots, M\}$ takes its own measurement $\pi_m(t)$ at time t, and arrives at global consensus on the optimal $\mathbf{p}(t)$ in a distributed manner, using only the messages exchanged with its one-hop neighbors $\mathcal{N}_m \subset \{1, 2, \ldots, M\}$. This problem has been tackled in [24] and [9]. In particular, the algorithm in [24] allows parallel update of the individual entries in $\mathbf{p}(t)$. We only make minor modifications to take care of nonnegativity of $\mathbf{p}(t)$.

It is emphasized that only the m-th row of $\mathbf{G}(t-1)$ is necessary to perform the updates at each CR $m \in \{1, \ldots, M\}$. Let $\bar{\mathbf{g}}_m(t-1)$ denote the m-th row of $\mathbf{G}(t-1)$. Then, the following iterative update is performed at CR m, with i denoting the iteration index:

$$\boldsymbol{\zeta}_m^{(i)} = \boldsymbol{\zeta}_m^{(i-1)} + \rho \sum_{m'\in\mathcal{N}_m}\left(\mathbf{p}_m^{(i)} - \mathbf{p}_{m'}^{(i)}\right) \tag{2.11}$$

$$\boldsymbol{\kappa}_m^{(i)} = \boldsymbol{\kappa}_m^{(i-1)} + \rho\left(\mathbf{p}_m^{(i)} - \mathbf{q}_m^{(i)}\right) \tag{2.12}$$

$$\mathbf{p}_m^{(i+1)} = \frac{1}{\rho\left(2\left|\mathcal{N}_m\right|+1\right)}\left[\rho\mathbf{q}_m^{(i)} - \boldsymbol{\zeta}_m^{(i)} - \boldsymbol{\kappa}_m^{(i)} + \rho\sum_{m'\in\mathcal{N}_m}\left(\mathbf{p}_m^{(i)} + \mathbf{p}_{m'}^{(i)}\right) - \frac{\lambda}{M}\right]^+ \tag{2.13}$$

$$\begin{aligned}\mathbf{q}_m^{(i+1)} &= \rho^{-1}\left[\mathbf{I} - \frac{\bar{\mathbf{g}}_m(t-1)^T\bar{\mathbf{g}}_m(t-1)}{\rho+\|\bar{\mathbf{g}}_m(t-1)\|_2^2}\right] \\ &\quad\cdot\left(\bar{\mathbf{g}}_m(t-1)^T(\pi_m(t)-\nu_m(t)) + \rho\mathbf{p}_m^{(i+1)} + \boldsymbol{\kappa}_m^{(i)}\right).\end{aligned} \tag{2.14}$$

Here, $|\cdot|$ is the cardinality of the set, $\rho>0$ a step size parameter, and $\boldsymbol{\zeta}_m$, $\boldsymbol{\kappa}_m$ and $\mathbf{q}_m$ are auxiliary variables. It can be seen that CR m just needs to collect locally from its neighbors $m'\in\mathcal{N}_m$ the current estimate of $\mathbf{p}_{m'}$ at each iteration. From the theory of alternating direction method of multipliers (ADMoM) on which the algorithm is based, it can be shown that the algorithm converges globally as i tends to infinity, to the consensus estimate $\mathbf{p}_1^{(i)} = \mathbf{p}_2^{(i)} = \ldots = \mathbf{p}_M^{(i)}$, which coincides with the solution $\mathbf{p}(t)$ to the problem in line 4 of Table 2.2, provided that the network is connected; i.e., there exist (multi-hop) paths from any node to any other node.

Based on the shared $\{\mathbf{p}(s)\}_{s=1}^t$ accumulated up to time t, the dictionary $\mathbf{G}$ can be updated in a distributed manner. This is straightforward since the cost in (2.10) as well as the constraint $\mathbf{G}\geq\mathbf{0}$ are separable in the individual rows of $\mathbf{G}$. Thus, each CR m needs to update $\bar{\mathbf{g}}_m$ by solving a nonnegativity-constrained least-squares problem. Obviously, this can be done in parallel. For simplicity, the coordinate descent method is used again in the proposed algorithm. The $\mathbf{G}(t)$ as $t\to\infty$ obtained here also converges to the $\mathbf{G}$ obtained in batch. The proof can be followed on same lines as in [18] with the only difference in Lemma 1. Here, $\mathbf{A}_{t+1}-\mathbf{A}_t = \eta\mathbf{I}+\mathbf{p}_m(t)\mathbf{p}_m(t)^T$ with $\beta=1$, which is still bounded and doesn't grow with t. Hence, $\mathbf{G}_{t+1}-\mathbf{G}_t = \mathcal{O}(\frac{1}{t})$. The overall algorithm is presented in Table 2.3, where $A_{m,kk}(t)$ denotes the (k,k)-th entry of matrix $\mathbf{A}_m(t)$, and $\tilde{b}_{mk}(t)$ the k-th entry of vector $\tilde{\mathbf{b}}_m(t)$. Line 23 replaces degenerate columns in $\mathbf{G}$ with random columns in a distributed manner.

2.4 Selection of sparsity parameter λ

The batch, online and distributed algorithm introduced earlier assume the knowledge of sparsity parameter λ. Two techniques for selecting best $\lambda\in\{\lambda_1,\cdots,\lambda_L\}$ ordered in

increasing order, are discussed in this section. Both techniques involve running batch algorithm for all λ with polishing in sparse regression step. The modified sparse regression step for $\mathbf{p}(t)$ is given by

$$\begin{aligned}
\tilde{\mathbf{p}} &= \arg\min_{\mathbf{p}\geq\mathbf{0}} \frac{1}{2}\|\boldsymbol{\pi}(t) - \boldsymbol{\nu}(t-1) - \mathbf{G}(t-1)\mathbf{p}\|_2^2 + \lambda\|\mathbf{p}\|_1. \\
\mathcal{I} &\triangleq \{i : \tilde{p}_i \geq 0\}. \\
\mathbf{p}_{\mathcal{I}}(t) &= \arg\min_{\mathbf{p}_{\mathcal{I}}\geq\mathbf{0}} \frac{1}{2}\|\boldsymbol{\pi}(t) - \boldsymbol{\nu}(t-1) - \mathbf{G}_{\mathcal{I}}(t-1)\mathbf{p}_{\mathcal{I}}\|_2^2. \\
\mathbf{p}_{\bar{\mathcal{I}}}(t) &= \mathbf{0}. \\
\mathbf{p}(t) &= \left[\mathbf{p}_{\mathcal{I}}(t)^T \mathbf{p}_{\bar{\mathcal{I}}}(t)^T\right]^T.
\end{aligned} \tag{2.15}$$

2.4.1 Approach 1

Select λ_i that produces steep jump in $e(\lambda) \triangleq \|\boldsymbol{\Pi} - \mathbf{GP} - \mathbf{1}^T \otimes \boldsymbol{\nu}\|_F^2$ when plotted against λ. As the value of λ increases, the lesser number of atoms in dictionary are used. After certain λ, number of atoms will not be sufficient to describe the model, so $e(\lambda)$ should increase abruptly.

2.4.2 Approach 2

Another technique assumes that average number of active PUs say ρ_{true} is known, which can be obtained from statistics of cellular system operator. Then, average number of detected PUs say ρ_d is compared with ρ_{true} and the λ which minimizes $\|\rho_d - \rho_{true}\|$ is picked. This could be easily extended to the online case by defining $\rho_{od}(t)$ as average number of detected PUs till time t and step size μ. Then instead of fixed λ in sparse regression step, $\lambda(t)$ could be used and can be updated as $\lambda(t) = \lambda(t-1) + \mu * (\rho_{od}(t) - \rho_{true})$. Similar update is applicable for distributed case as ρ_{true} will be available at each CRs, and $\rho_{od}^m(t)$ defined as average number of detected PUs at CR m is equal to $\rho_{od}(t)$ because $\mathbf{p}_m^{(i+1)}$ converges to $\mathbf{p}(t)$ for all m. Then $\lambda^m(t)$ at CR m is updated as $\lambda^m(t) = \lambda^m(t-1) + \mu * (\rho_{od}^m(t) - \rho_{true})$.

2.5 Resolution of scaling ambiguities

The channel gain matrix $\mathbf{G}$ in preceding algorithms are learned up to scaling ambiguity. In this section, we will resolve the issue of scaling ambiguity in columns of $\mathbf{G}$ by assuming that $\mathbf{p}_i(t)$ take values from known finite alphabet $\mathcal{P}$. Assumption of knowing finite alphabet is reasonable as technologies like Bluetooth use fixed known power levels. since $\mathbf{p}_i(t) \in \mathcal{P}$, columns of $\mathbf{G}$ cannot grow arbitrarily large so the constraints $\|\mathbf{g}_k\|_2 \leq 1$ for $k = 1, \ldots, K$ will be dropped. Let us define

$$\mathcal{Z} \triangleq \{[\mathbf{g}_1, \ldots, \mathbf{g}_K] \geq \mathbf{0} \text{ for } k = 1, \ldots, K\} \tag{2.16}$$

Now, the modified optimization problem is given by

$$\min_{\mathbf{G}\in\mathcal{Z}, \mathbf{P}\in\mathcal{P}^{K\times T}, \boldsymbol{\nu}\geq\mathbf{0}} \frac{1}{2}\|\mathbf{\Pi} - \mathbf{GP} - \mathbf{1}^T \otimes \boldsymbol{\nu}\|_F^2 + \lambda\|\mathbf{P}\|_1 \tag{2.17}$$

As similar to previous case, online formulation for solving at time t is given by

$$\min_{\mathbf{G}\in\mathcal{Z}, \mathbf{p}(t)\in\mathcal{P}^K, \boldsymbol{\nu}\geq\mathbf{0}} \sum_{s=1}^{t} \beta^{t-s} \left(\frac{1}{2}\|\boldsymbol{\pi}(s) - \mathbf{Gp}(s) - \boldsymbol{\nu}\|_2^2 + \lambda\|\mathbf{p}(s)\|_1 \right) \tag{2.18}$$

Modified problem in (2.17) is non-convex, therefore global optimum solution cannot be guaranteed. However, when $\mathbf{P}$ is fixed, the problems for $\mathbf{G}$, $\boldsymbol{\nu}$ become convex. Likewise, when $\mathbf{G}$, $\boldsymbol{\nu}$ is fixed, optimizing over $\mathbf{P}$ involves solving problem of multi user detection (MUD). Therefore, alternating minimization is employed to obtain locally optimal solutions. Next, both batch and online algorithms are developed.

The batch algorithm for solving (2.17) is presented in Table 2.4. As opposed to previous case, dictionary is updated first in order to roughly capture the norm of columns of channel gain matrix $\mathbf{G}$. Once dictionary $\mathbf{G}$ is updated, $\mathbf{p}(t)$ for each $t = 1, 2, \cdots, T$ is obtained by doing an exhaustive search over $\mathcal{P}^K$. Methods are developed in [25]-[26] with reduced average complexity for solving problem in line 16 but these methods require QR decomposition of $\mathbf{G}$ at every iteration, which is costly. Line 18 remove similar rows in $\mathbf{P}$ by retaining one row and setting other to zero. This operation combine columns of $\mathbf{G}$ and make $\mathbf{P}$ more sparse, therefore reduces the cost of over all objective function given by 2.5. Here, min_norm_$\Pi \triangleq min_{\{t\}_1^T} \|\boldsymbol{\pi}_t\|_2$.

Online algorithm is developed in Table 2.5 for solving (2.18). Again, dictionary $\mathbf{G}(t)$ obtained from online algorithm converges to the same $\mathbf{G}$ that would have been obtained in batch algorithm.

Input: $\mathbf{\Pi}$, initial matrix $\mathbf{P}_0$, min_norm_Π and λ
Output: $\mathbf{G}$ and $\mathbf{P}$

Set $\mathbf{P} \triangleq [\mathbf{p}(1), \ldots, \mathbf{p}(T)] = \mathbf{P}_0$, $\mathbf{c} = \mathbf{\Pi 1^T}$ and $\boldsymbol{\nu} = \mathbf{0}$
Repeat
    Perform dictionary update with fixed $\mathbf{P}$.
        Compute $\mathbf{A} = \mathbf{P}\mathbf{P}^T$, $\mathbf{B} = \mathbf{\Pi}\mathbf{P}^T$, $\mathbf{d} = \mathbf{P 1^T}$
        Repeat
            $\tilde{\mathbf{B}} = \mathbf{B} - \boldsymbol{\nu}\mathbf{d}^T$
            For $k = 1, 2, \ldots, K$
                $\tilde{\mathbf{g}}_k = \frac{1}{A_{kk}} \left(\tilde{\mathbf{b}}_k - \mathbf{G}\mathbf{a}_k \right) + \mathbf{g}_k$
                $\mathbf{g}_k = [\tilde{\mathbf{g}}_k]^+$
            Next k
                $\boldsymbol{\nu} = \left[\frac{\mathbf{c} - \mathbf{G}\mathbf{d}}{T} \right]^+$
        until convergence
Replace all-zero columns in G to random vectors with norm equals to min_norm_Π.
    Perform Multi user detection with fixed $\mathbf{G}$, $\boldsymbol{\nu}$.
        For $t = 1, 2, \ldots, T$
            $\mathbf{p}(t) = \arg \min_{\mathbf{p} \in \mathcal{P}^K} \frac{1}{2} \|\boldsymbol{\pi}(t) - \boldsymbol{\nu} - \mathbf{G}\mathbf{p}\|_2^2 + \lambda \|\mathbf{p}\|_1$
        Next t
Remove similar rows i.e. If rows of P are same then retain one of them and set other to zero.
until convergence

Table 2.4: Batch algorithm under finite-alphabet constraints.

Input: $\boldsymbol{\pi}(t)$ for $t = 1, 2, \ldots$, initial vector $\mathbf{p}(0)$, min_norm_Π and λ
Output: $\mathbf{G}(t)$ and $\mathbf{p}(t)$ for $t = 1, 2, \ldots$

Set $\mathbf{A}(0) = \mathbf{B}(0) = \mathbf{G}(0) = \mathbf{0}$, $\mathbf{c}(0) = \mathbf{d}(0) = \boldsymbol{\nu}(0) = \mathbf{0}$ and $\boldsymbol{\pi}(0) = \boldsymbol{\pi}(1)$
For $t = 1, 2, \ldots$
    Perform dictionary update.
        Compute:
            $\mathbf{A}(t) = \beta\mathbf{A}(t-1) + \mathbf{p}(t-1)\mathbf{p}(t-1)^T$
            $\mathbf{B}(t) = \beta\mathbf{B}(t-1) + \boldsymbol{\pi}(t-1)\mathbf{p}(t-1)^T$
            $\mathbf{c}(t) = \beta\mathbf{c}(t-1) + \boldsymbol{\pi}(t-1)$
            $\mathbf{d}(t) = \beta\mathbf{d}(t-1) + \mathbf{p}(t-1)$
        Set $\mathbf{G} \triangleq [\mathbf{g}_1, \ldots, \mathbf{g}_K] = \mathbf{G}(t-1)$ and $\boldsymbol{\nu} = \boldsymbol{\nu}(t-1)$(warm start)
        Repeat
            $\tilde{\mathbf{B}}(t) = \mathbf{B}(t) - \boldsymbol{\nu}\mathbf{d}(t)^T$
            For $k = 1, 2, \ldots, K$
                $\tilde{\mathbf{g}}_k = \frac{1}{A_{kk}(t)}\left[\tilde{\mathbf{b}}_k(t) - \mathbf{G}\mathbf{a}_k(t)\right] + \mathbf{g}_k$
                $\mathbf{g}_k = [\tilde{\mathbf{g}}_k]^+$
            Next k
            $\boldsymbol{\nu} = \frac{1-\beta}{1-\beta^t}\left[\mathbf{c}(t) - \mathbf{G}\mathbf{d}(t)\right]^+$
        until convergence
        Replace all-zero columns in $\mathbf{G}$
        to random vectors with norm equals to min_norm_Π.
        Set $\mathbf{G}(t) = \mathbf{G}$ and $\boldsymbol{\nu}(t) = \boldsymbol{\nu}$
    Perform sparse regression.
        $\mathbf{p}(t) = \arg\min_{\mathbf{p}\in\mathcal{P}^K\times 1} \frac{1}{2}\|\boldsymbol{\pi}(t) - \boldsymbol{\nu}(t) - \mathbf{G}(t)\mathbf{p}\|_2^2 + \lambda\|\mathbf{p}\|_1$
Next t

Table 2.5: Online algorithm under finite-alphabet constraints.

Chapter 3

Numerical Tests

To verify the proposed algorithms, various numerical tests are performed. A network of $M = 10$ CR nodes is used to detect $K_0 = 3$ PUs. The locations of the CRs and the PUs are depicted in Fig. 3.1 using circles and triangles, respectively, while single-hop connectivity is represented by the edges of the graph. The number of atoms in the dictionary is set to $K = 5$. The channels g_{mk} are given by $g_{mk} = (d_{mk}/d_0)^{-\gamma}|h_{mk}|^2$ for $m = 1, \ldots, M$ and $k = 1, \ldots, K_0$, where d_{mk} is the distance between CR m and PU k, d_0 is the reference distance set to 0.1, γ is the path loss exponent set to 2.5, and h_{mk} are Rayleigh fading coefficients following an autoregressive model

$$h_{mk}(t+1) = \alpha h_{mk}(t) + \sqrt{1-\alpha^2} z_{mk}(t) \tag{3.1}$$

with h_{mk} and z_{mk} drawn independently from the standard complex Gaussian distribution. When a time-varying channel is generated, $\alpha = 0.9999$ is used, while for fixed channels, $\alpha = 1$. The PUs are active with probability 0.3, and upon being active, their transmission powers are drawn independently from the uniform distribution with support $[0.1, 1]$.

3.1 Batch, Online and distributed algorithm

Fig. 3.2 shows the true PU transmit-powers (top panel) and the estimated PU powers (bottom panel). The channels were time-varying and the (centralized) online algorithm with $\beta = 0.95$ was employed. Different choices of β affects the convergence rate and can

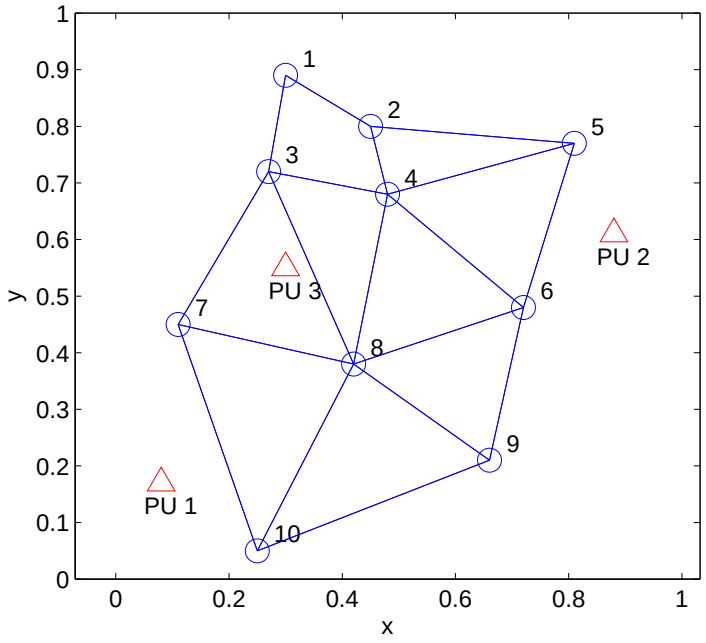

Figure 3.1: CR network topology.

also be chosen as $\beta_t = (1 - 1/t)^\theta$, where θ can be chosen to improve the convergence rate [18]. To identify the PUs detected, correlation coefficients between the true and the estimated channels were computed and thresholded at 0.9. When more than one atoms of the dictionary had correlation higher than the threshold, the corresponding entries in $\mathbf{p}(t)$ were summed. It can be seen from the figure that the relative magnitudes of PU powers have been correctly estimated. Note that since the channels are normalized, the distance-based path loss effect is captured in the estimated PU powers; cf. Fig. 3.1. Moreover, the network sensing algorithm can detect the correct set of active PUs from the mixture observations, by checking whether the estimated PU powers are positive. Fig. 3.3 depicts the corresponding receiver operating characteristics (ROCs) for detecting the individual PUs.

To assess the quality of channel estimation, the normalized mean-square error (MSE) performance of the batch, online and distributed algorithms as the noise power is varied is presented in Fig. 3.4 for the case where the channels are fixed. For the distributed case, $\rho = 1$ is chosen experimentally to achieve good convergence rate and $\eta = 0.01$ is

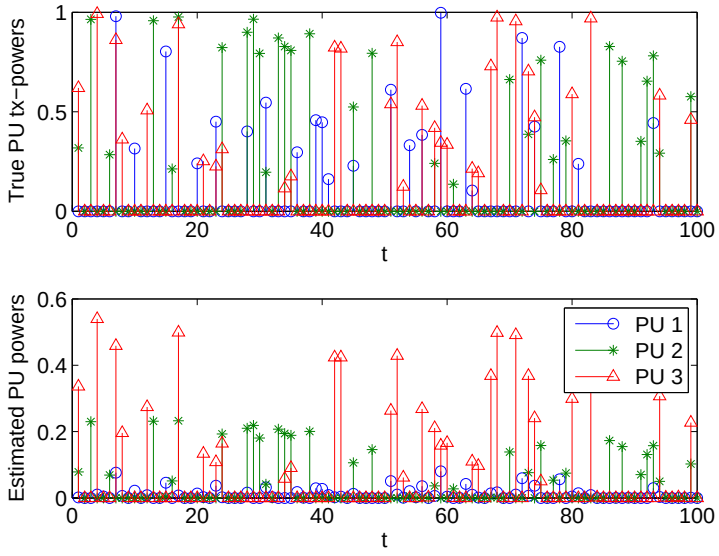

Figure 3.2: True and estimated PU transmit-powers.

used. It can be seen that the batch algorithm outperforms the online versions especially at low SNR. On the other hand, the performance of the centralized and the distributed online algorithms is indistinguishable. Similar observation can be made from the MSE-versus-time curves of Fig. 3.5 for time-varying channels, where the noise power is 10^{-3}.

3.2 Tuning of the sparsity parameter λ

To verify the approach 1 for selecting sparsity parameter λ, variation of LSE$\triangleq \|\Pi - \mathbf{GP}\|_F^2$ and MSE over λ is presented in Fig. 3.6. Here, MSE, LSE and average number of PU are computed over 52 different λ logarithmically spaced in the interval $[e^{(-10)} \quad 1]$. It is clear from Fig. 3.6 that λ which provides the steep jump matches with the λ which provides minimum MSE say λ^*. Variation of average number of detected PUs and MSE over λ is presented in Fig. 3.7 to verify approach 2. Again, the λ at which average number of detected PU is same as average number of actual PUs matches with λ^*.

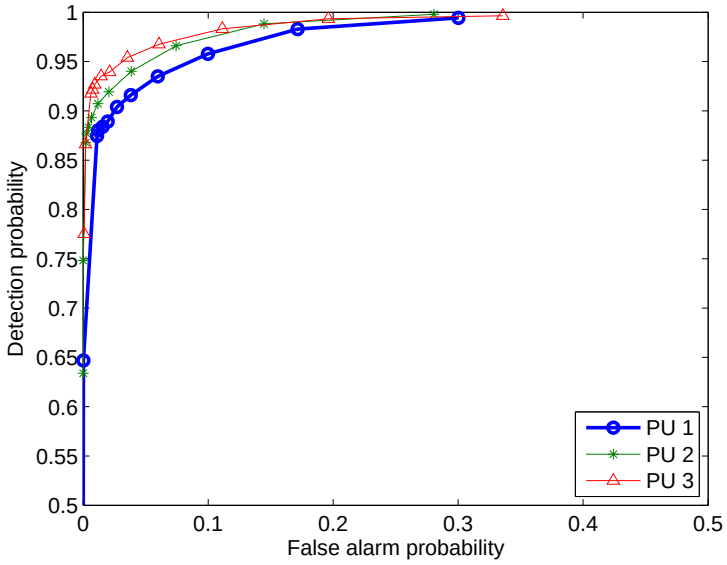

Figure 3.3: ROC curves.

3.3 Finite-alphabet constraints

The true PU transmit-powers (top panel) and the PU powers (bottom panel) estimated by the (centralized) online algorithm for the fixed channel are shown in Fig. 3.8. PU transmit powers were drawn from known finite alphabet $\mathcal{P} = \{0, 1, 2\}$ with probability distibution of $\{0.7, 0.15, 0.15\}$. Initial matrix $\mathbf{P}_0$ for batch algorithm is densely populated with entries from $\mathcal{P} = \{0, 1, 2\}$. It is clear from Fig. 3.8 that PU transmit powers are recovered exactly.

Fig. 3.9 presents the normalized MSE performance of the batch and online algorithms as the SNR is varied for the case where the channels are fixed and PU transmit powers were drawn from $\mathcal{P} = \{0, 1\}$ with probability of active being 0.3. With prior information of finite alphabet, one might think that MSE at different scaled SNR in Fig. 3.9 should be less than MSE at different scaled SNR in Fig. 3.4, but there is an additional error in Fig. 3.9 from mismatching of norm of channel gains. Despite of that, MSE in Fig. 3.9 is smaller to MSE in Fig. 3.4.

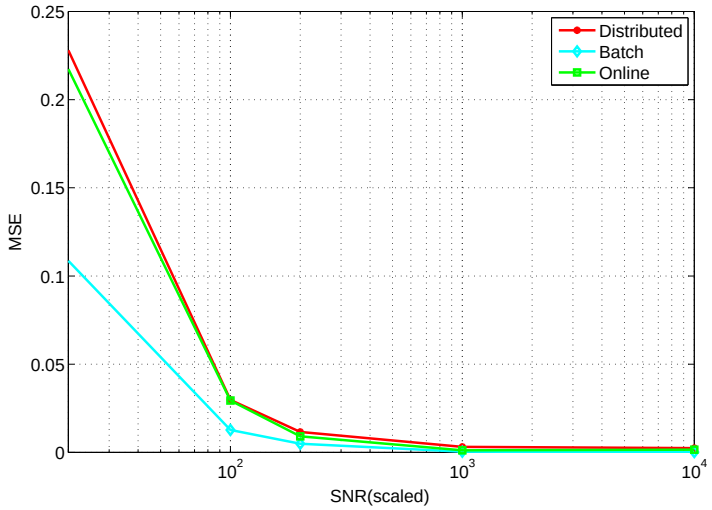

Figure 3.4: MSE vs. scaled SNR.

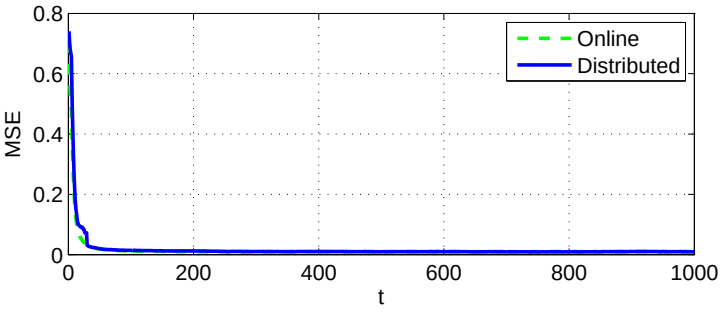

Figure 3.5: MSE vs. time.

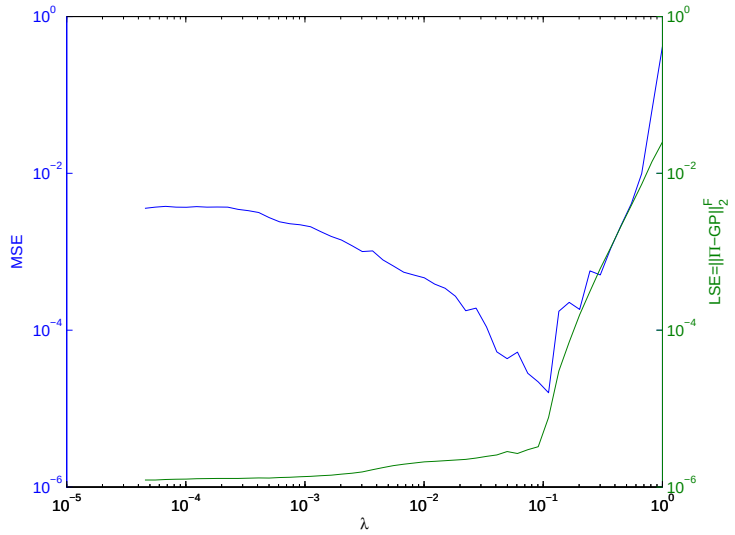

Figure 3.6: MSE and LSE vs. λ.

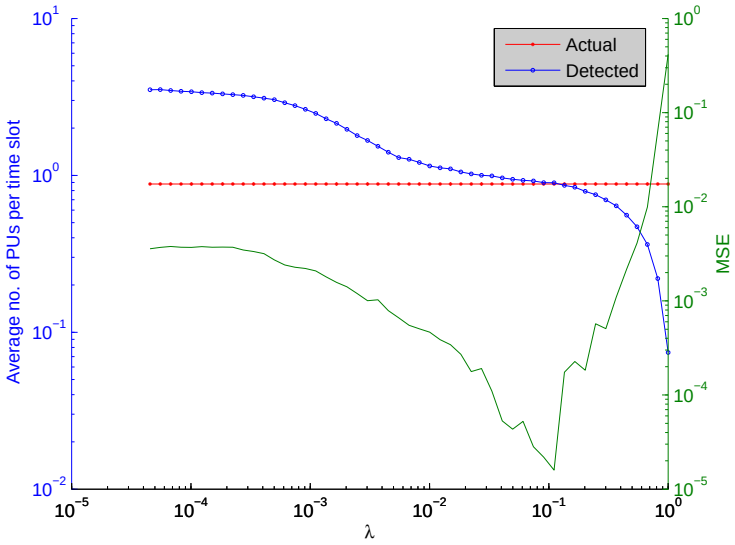

Figure 3.7: MSE and average no. of PU vs. λ.

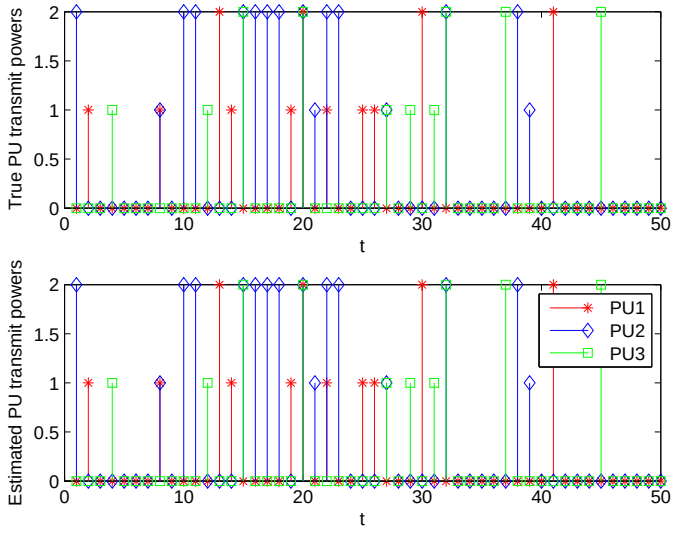

Figure 3.8: True and estimated PU transmit powers under finite-alphabet constraints.

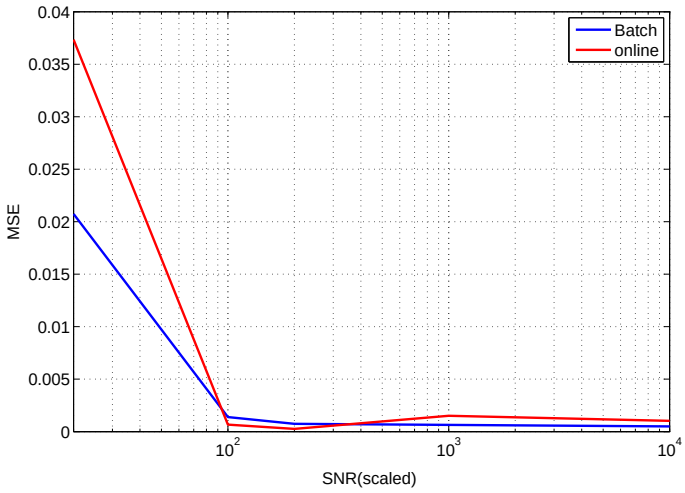

Figure 3.9: MSE vs scaled SNR under finite-alphabet constraints.

Chapter 4

Conclusions

Cooperative spectrum sensing algorithms have been proposed that can detect the active set of PUs at each time instant while jointly acquiring the channels between CRs and PUs. Employing a dictionary learning approach, the algorithms factorized the received energy sample matrix into a dictionary with nonnegative entries that represented normalized channel gains, and a sparse nonnegative matrix signifying PU transmit-powers. Batch, online, as well as distributed online algorithms were developed. Techniques for selecting sparsity parameter λ is also developed for batch, online and distributed case. The numerical tests demonstrated that the algorithms can estimate individual PU powers and track slowly varying channels, up to scaling ambiguity. Also, numerical results verifying both approaches for selecting λ is reported.

Later, scaling ambiguity is removed by assumption that transmit-powers taking values from finite alphabet. Again, batch and online algorithms are developed. Numerical results reporting the almost recovery of channel gain matrix and transmit powers are presented.

Algorithms for spectrum sensing is developed in this thesis which is only the half job done for CRs. Integration of dynamic resource allocation for CR with spectrum sensing is considered for future work.

Theoretical proof for convergence of solution of online algorithm to solution of batch algorithm in case of finite alphabet has not been established and is left for future work.

This thesis considers that all CRs transmit their information to fusion center in batch and online case or share information with their neighbors in distributed case

for spectrum sensing. In practice, some of the CRs may not be willing to share their information or are inactive to save their battery power. So, spectrum sensing with only few CRs randomly sharing their information could be also considered for future work. Also, since nearby CRs sense correlated PU activities an extra laplacian penalty term could be added in objective function [27] to improve learning.

References

[1] FCC. Spectrum policy task force report. ET Docket no. 02-135, Nov. 2002. [online] `http://hraunfoss.fcc.gov/edocspublic/attachmatch/DOC-228542A1.pdf`.

[2] S. Haykin. Cognitive radio: brain-empowered wireless communications. *IEEE J. Sel. Areas Commun.*, 23(2):201–220, Feb. 2005.

[3] Q. Zhao and B. M. Sadler. A survey of dynamic spectrum access. *IEEE Sig. Proc. Mag.*, 24(3):79–89, May 2007.

[4] A. Ghasemi and E. S. Sousa. Spectrum sensing in cognitive radio networks: the cooperation-processing tradeoff. *Wireless Commun. Mobile Comput.*, 7(9):1049–1060, Nov. 2007.

[5] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan. Cooperative spectrum sensing in cognitive radio networks: A survey. *Physical Communication*, 4(1):40–62, Mar. 2011.

[6] J. Unnikrishnan and V. V. Veeravalli. Cooperative sensing for primary detection in cognitive radio. *IEEE J. Sel. Topics Sig. Proc.*, 2(1):18–27, Feb. 2008.

[7] A. Alaya-Feki, S. Ben Jemaa, B. Sayrac, P. Houze, and E. Moulines. Informed spectrum usage in cognitive radio networks: interference cartography. In *Proc. of IEEE 19th Intl. Symp. Personal, Indoor and Mobile Radio Commun.*, pages 1–5, Cannes, France, Sep. 2008.

[8] J. A. Bazerque, G. Mateos, and G. B. Giannakis. Group-Lasso on splines for spectrum cartography. *IEEE Trans. Sig. Proc.*, 59(10):4648–4663, Oct. 2011.

[9] Seung-Jun Kim, E. Dall'Anese, and G. B. Giannakis. Cooperative spectrum sensing for cognitive radios using Kriged Kalman filtering. *IEEE J. Sel. Topics Sig. Proc.*, 5(1):24–36, Feb. 2011.

[10] E. Dall'Anese, S.-J. Kim, and G. B. Giannakis. Channel gain map tracking via distributed Kriging. *IEEE Trans. Veh. Technol.*, 60(3):1205–1211, Mar. 2011.

[11] S.-J. Kim, N. Jain, G. B. Giannakis, and P.A. Forero. Joint link learning and cognitive radio sensing. In *Proc. of the Asilomar Conf.*, pages 1415–1419, Pacific Grove, CA, Dec. 2011.

[12] B. A. Olshausen and D. J. Field. Sparse coding with an covercomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, Dec. 1997.

[13] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Proc.*, 15(12):3736–3745, Dec. 2006.

[14] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies. Sparse representations in audio and music: from coding to source separation. *Proc. of IEEE*, 98(6):995–1005, Dec. 2009.

[15] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *Proc. of the 24th Intl. Conf. on Machine Learning*, pages 759–766, Corvallis, OR, Jun. 2007.

[16] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 801–808. MIT Press, Cambridge, MA, 2007.

[17] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Sig. Proc.*, 54(11):4311–4322, Nov. 2006.

[18] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *J. Machine Learning Res.*, 11:19–60, Mar. 2010.

[19] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE. Trans. of Information Theory*, 51(12):4203–4215, Dec. 2005.

[20] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Stat.*, 32(2):407–499, 2004.

[21] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.

[22] I. D. Schizas, A. Ribeiro, and G. B. Giannakis. Consensus in *ad hoc* WSNs with noisy links—Part I: Distributed estimation of deterministic signals. *IEEE Trans. Sig. Proc.*, 1(56):350–364, Jan. 2008.

[23] V. P. Pauca, J. Piper, and R. J. Plemmons. Nonnegative matrix factorization for spectral data analysis. *Linear Algebra and its Appl.*, 416(1):29–47, Jul. 2006.

[24] G. Mateos, J. A. Bazerque, and G. B. Giannakis. Distributed sparse linear regression. *IEEE Trans. Sig. Proc.*, 58(10):5262–5276, Oct. 2010.

[25] H. Zhu and G. B. Giannakis. Exploiting sparse user activity in multiuser detection. *IEEE. Trans. on Comm.*, 59(2):454–465, Feb. 2011.

[26] Z.Tian, G. Leus, and V. Lottici. Detection of sparse signals under finite alphabet constraints. In *Proc. of the ICASSP Conf.*, pages 2349–2352, Taipei, Apr. 2011.

[27] M. Zheng, J. Bu, C. Chen, C.Wang, L.Zhang, G.Qiu, and D.Cai. Graph regularized sparse coding for image representation. *IEEE. Trans. on Image Processing*, 20(5):1327–1336, May. 2011.

Appendix A

A.1 Glossary

- **CR** – Cognitive Radio.
- **PU** – Licensed primary user.
- **SINR**– Signal-to-interference-plus-noise-ratio.
- **MUD** – Multi user detection
- **SNR** – Signal to noise ratio
- **MSE** – Mean square error
- **ROC** – Receiver operating characteristic
- **LSE** – Least square error
- **CSI** – Channel state information

A.2 Derivation of dictionary update step

Consider minimization over $\mathbf{G}$ and $\boldsymbol{\nu}$ in dictionary update step

$$\mathbf{G} = \arg \min_{\mathbf{G}\in\mathcal{G},\boldsymbol{\nu}\geq\mathbf{0}} \sum_{t=1}^{T} \left(\frac{1}{2}\|\boldsymbol{\pi}_t - \boldsymbol{\nu} - \mathbf{G}\mathbf{p}_t\|_2^2 + \lambda \mathbf{1^T p_t} \right) \tag{A.1}$$

Since the constraints are separable, the above constrained problem can be solved using block co-ordinate descent by fixing all but $\mathbf{g}_i$ and can be rewritten in $\mathbf{g}_i$ as

$$\mathbf{g}_i = \arg \min_{\mathbf{g}_i\succeq\mathbf{0},\mathbf{g}_i^T\mathbf{g}_i\leq 1} \sum_{t=1}^{T} \left(\frac{1}{2}\|\mathbf{f}_t^i - \mathbf{g}_i p_{i,t}\|_2^2 \right) \tag{A.2}$$

where, $\mathbf{f}_t^i = \boldsymbol{\pi}_t - \boldsymbol{\nu} - \mathbf{G}\mathbf{p}_t + \mathbf{g}_i p_{i,t}$.
Using the lagrange multipliers ($\delta \geq 0, \boldsymbol{\mu} \geq \mathbf{0}$) (A.2) can be converted to an unconstrained problem given by

$$\mathbf{g}_i = \arg\min_{\mathbf{g}_i} \sum_{t=1}^{T} \left(\frac{1}{2}\|\mathbf{f}_t^i - \mathbf{g}_i p_{i,t}\|_2^2 \right) + \delta\|\mathbf{g}_i\|_2^2 - \boldsymbol{\mu}^T\mathbf{g}_i$$

and the closed form solution for (A.2) is given by

$$\mathbf{g}_i^* = \frac{\sum_{t=1}^{T} \mathbf{f}_t^i p_{i,t} + \boldsymbol{\mu}^*}{\sum_{t=1}^{T} p_{i,t}^2 + \delta\|\mathbf{g}_i^*\|} \tag{A.3}$$

As $\mathbf{A} = \mathbf{P}\mathbf{P}^T$ and $\tilde{\mathbf{B}} = (\boldsymbol{\Pi} - \mathbf{1}^T \otimes \boldsymbol{\nu})\mathbf{P}^T$, then $\mathbf{a}_i = \sum_{t=1}^{T} \mathbf{p}_t p_{i,t}$ and $\tilde{\mathbf{b}}_i = \sum_{t=1}^{T} (\boldsymbol{\pi}_t - \boldsymbol{\nu}) p_{i,t}$.
After substituting above and $\mathbf{f}_t^i$ in numerator of (A.2), it can be rewritten as

$$\mathbf{g}_i^* = \frac{\tilde{\mathbf{b}}_i - \mathbf{G}\mathbf{a}_i + \mathbf{g}_i A_{i,i}}{A_{i,i} + \delta\|\mathbf{g}_i^*\|}$$

As defined in batch algorithm, $\tilde{\mathbf{g}}_i = \frac{\tilde{\mathbf{b}}_i - \mathbf{G}\mathbf{a}_i + \mathbf{g}_i A_{i,i}}{A_{i,i}}$, so numerator is simply $A_{ii}\tilde{\mathbf{g}}_i + \boldsymbol{\mu}^*$.
Considering two possible cases for δ,
Case1: $\delta^* = 0$ and $\|\mathbf{g}_i^*\| < 1$, then $\mathbf{g}_i^* = [\tilde{\mathbf{g}}_i]^+$
Case2: $\delta^* > 0$ and $\|\mathbf{g}_i^*\| = 1$, then $\delta^* = \|[\tilde{\mathbf{g}}_i]^+\|A_{ii} - A_{ii}$ implies $\mathbf{g}_i^* = \frac{[\tilde{\mathbf{g}}^i]^+}{\|[\tilde{\mathbf{g}}^i]^+\|}$
Combining both cases, $\mathbf{g}_i^* = \frac{[\tilde{\mathbf{g}}^i]^+}{\max\{\|[\tilde{\mathbf{g}}^i]^+\|, 1\}}$

Now consider minimizing (A.1) over $\boldsymbol{\nu}$ in dictionary update step can be rewritten as

$$\boldsymbol{\nu} = \arg\min_{\boldsymbol{\nu} \geq \mathbf{0}} \sum_{t=1}^{T} \left(\frac{1}{2} \|\mathbf{r}_t - \boldsymbol{\nu}\|_2^2 \right) \tag{A.4}$$

where, $\mathbf{r}_t = \boldsymbol{\pi}_t - \mathbf{G}\mathbf{p}_t$.

By adding a lagrange multiplier ($\boldsymbol{\gamma} \geq \mathbf{0}$), the above is equivalent to

$$\boldsymbol{\nu} = \arg\min_{\boldsymbol{\nu}} \sum_{t=1}^{T} \left(\frac{1}{2} \|\mathbf{r}_t - \boldsymbol{\nu}\|_2^2 \right) - \boldsymbol{\gamma}^T \boldsymbol{\nu} \tag{A.5}$$

and the closed form solution for above is given by

$$\boldsymbol{\nu}^* = \left[\frac{\sum_{t=1}^{T} \mathbf{r}_t}{T} \right]^+ . \tag{A.6}$$

$$= \left[\frac{\sum_{t=1}^{T} \boldsymbol{\pi}_t - \mathbf{G} \sum_{t=1}^{T} \mathbf{p}_t}{T} \right]^+ . \tag{A.7}$$

$$= \left[\frac{\mathbf{c} - \mathbf{G}\mathbf{d}}{T} \right]^+ . \tag{A.8}$$