PRIMARY USER BEHAVIOR ESTIMATION AND CHANNEL ASSIGNMENT FOR DYNAMIC SPECTRUM ACCESS IN ENERGY-CONSTRAINED COGNITIVE RADIO SENSOR NETWORKS

By

XIAOYUAN LI

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2013
© 2013 Xiaoyuan Li
To my family
ACKNOWLEDGMENTS

First and foremost, I want to express my sincerest gratitude to Dr. Janise McNair, for her supportive advice, patience and kindness on my graduate studies. Dr McNair is a respectable, responsible and resourceful professor and she has provided me with valuable advice in the writing of this thesis. Her keen and vigorous academic observation enlightens me not only in this thesis but also in my future career.

I will extend my thanks to my PhD committee members. I would like to thank Dr. Liuqing Yang for all the pleasant talk with her, Dr. Dapeng Wu for his help and encouragement, Dr. Shigang Chen for his precious suggestion on my research work and Dr. Lei Zhang for his kindness. I would also like to express my appreciations to them for reviewing my manuscript and giving comments.

I would also like to show my grateful thanks to all my labmates, who made my PhD life colorful and much easier. I will give my special appreciation to Dexiang Wang, who provided many useful help and suggestion on my PhD research. I will also give my thanks to Xiang Mao, for his help at the last phase of my dissertation. Last but not the least, my thanks would go to my beloved family, my husband and my parents, whose love and support is the most important thing to me and help me moving forward.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>4</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>7</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>8</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>10</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1 INSTRUCTION</td>
<td>12</td>
</tr>
<tr>
<td>1.1 Cognitive Radio Sensor Networks</td>
<td>12</td>
</tr>
<tr>
<td>1.2 Energy Challenges</td>
<td>13</td>
</tr>
<tr>
<td>1.3 Issues and Contributions</td>
<td>14</td>
</tr>
<tr>
<td>1.3.1 Primary User Behavior Estimation</td>
<td>14</td>
</tr>
<tr>
<td>1.3.2 Residual Energy Aware Channel Assignment</td>
<td>15</td>
</tr>
<tr>
<td>1.3.3 Dynamic Spectrum Access with Packet Size Adaptation</td>
<td>15</td>
</tr>
<tr>
<td>1.4 Organization</td>
<td>15</td>
</tr>
<tr>
<td>2 SYSTEM MODELS</td>
<td>17</td>
</tr>
<tr>
<td>2.1 Frame Structure and Network model</td>
<td>17</td>
</tr>
<tr>
<td>2.1.1 Spectrum Sensing Model</td>
<td>17</td>
</tr>
<tr>
<td>2.1.2 Channel Assignment Model</td>
<td>18</td>
</tr>
<tr>
<td>2.2 Primary User Behavior Model</td>
<td>20</td>
</tr>
<tr>
<td>2.3 Energy Consumption Model</td>
<td>21</td>
</tr>
<tr>
<td>3 PRIMARY USER BEHAVIOR ESTIMATION WITH PERFECT SENSING</td>
<td>23</td>
</tr>
<tr>
<td>3.1 Overview of PU Estimation Challenges</td>
<td>23</td>
</tr>
<tr>
<td>3.2 Related Work</td>
<td>25</td>
</tr>
<tr>
<td>3.3 The Distribution of Maximum Likelihood Estimator</td>
<td>26</td>
</tr>
<tr>
<td>3.3.1 Derivation of the Probability Mass Function (PMF)</td>
<td>27</td>
</tr>
<tr>
<td>3.3.2 Using Normal Distribution to Approximate the Real Estimation Distribution</td>
<td>31</td>
</tr>
<tr>
<td>3.4 The Analysis of the Estimation Accuracy</td>
<td>34</td>
</tr>
<tr>
<td>3.4.1 Confidence Interval</td>
<td>34</td>
</tr>
<tr>
<td>3.4.2 The Required Length of the Sample Sequence</td>
<td>36</td>
</tr>
<tr>
<td>3.5 The Adaptation of the Sample Sequence Length</td>
<td>37</td>
</tr>
<tr>
<td>3.6 Simulation Results</td>
<td>40</td>
</tr>
<tr>
<td>3.6.1 Estimation Accuracy of Exact PMF</td>
<td>40</td>
</tr>
<tr>
<td>3.6.2 Estimation Accuracy of Adaptation Algorithm</td>
<td>43</td>
</tr>
<tr>
<td>3.7 Summary</td>
<td>47</td>
</tr>
</tbody>
</table>
# PRIMARY USER BEHAVIOR ESTIMATION WITH IMPERFECT SENSING

## 4.1 Overview of Imperfect Sensing

## 4.2 Related Work

## 4.3 The Hidden Markov Model of PU behavior

### 4.3.1 Model of Imperfect Sensing

### 4.3.2 Structure of Hidden Markov Model

## 4.4 Estimation of Transition Probabilities Using HMM

### 4.4.1 Probability of a given observed sequence

### 4.4.2 Estimation of HMM Model Parameters Using Baum-Welch Algorithm

## 4.5 The Estimation Accuracy Analysis of HMM

### 4.5.1 Confidence Level of the Estimation

### 4.5.2 Selection of Initial Parameters

## 4.6 Numerical Results

## 4.7 Summary

# RESIDUAL ENERGY AWARE CHANNEL ASSIGNMENT SCHEMES

## 5.1 Spectrum Sharing in Cognitive Radio Sensor Networks (CRSNs)

## 5.2 Related Work

## 5.3 R-Coefficient

## 5.4 Channel Assignment

### 5.4.1 Random Pairing

### 5.4.2 Greedy Channel Search

### 5.4.3 Optimization-based Channel Assignment

## 5.5 Simulation Results

## 5.6 Summary

# DYNAMIC SPECTRUM ACCESS WITH PACKET SIZE ADAPTATION AND RESIDUAL ENERGY BALANCING

## 6.1 Dynamic Spectrum Access and Energy Consumption

## 6.2 Related Work

## 6.3 Packet Size Adaptation

## 6.4 Residual Energy Balancing Channel Assignment

## 6.5 Simulation Results

### 6.5.1 Performance of Packet Size Adaptation

### 6.5.2 Residual Energy Balancing Channel Assignment

### 6.5.3 Impacts of estimation accuracy

## 6.6 Summary

# CONCLUSIONS

REFERENCES

BIOGRAPHICAL SKETCH
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-1  HMM Symbols</td>
<td>56</td>
</tr>
<tr>
<td>5-1  Simulation parameters of residual energy aware channel assignment</td>
<td>82</td>
</tr>
<tr>
<td>6-1  Simulation parameters of dynamic spectrum access with packet size adaption</td>
<td>96</td>
</tr>
<tr>
<td>6-2  The impact of estimation accuracy on other metrics</td>
<td>100</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Time-slotted structure of a frame</td>
<td>17</td>
</tr>
<tr>
<td>2-2</td>
<td>A cluster-based CRSN with PU</td>
<td>18</td>
</tr>
<tr>
<td>2-3</td>
<td>Time-slotted structure of a frame</td>
<td>19</td>
</tr>
<tr>
<td>2-4</td>
<td>A two-state Markov model of PU behavior</td>
<td>20</td>
</tr>
<tr>
<td>3-1</td>
<td>Probability distribution of $\hat{p}$</td>
<td>32</td>
</tr>
<tr>
<td>3-2</td>
<td>The comparison of binomial and norm distribution</td>
<td>33</td>
</tr>
<tr>
<td>3-3</td>
<td>The comparison of exact and norm distribution ($p = 0.5, q = 0.5$)</td>
<td>35</td>
</tr>
<tr>
<td>3-4</td>
<td>The impact of transition probabilities on the required sample sequence length</td>
<td>38</td>
</tr>
<tr>
<td>3-5</td>
<td>The flow chart of the proposed learning algorithm</td>
<td>39</td>
</tr>
<tr>
<td>3-6</td>
<td>Impact of transition probabilities on estimation accuracy of $\hat{p}$</td>
<td>41</td>
</tr>
<tr>
<td>3-7</td>
<td>Impact of number of samples on estimation accuracy ($p = q = 0.1$)</td>
<td>42</td>
</tr>
<tr>
<td>3-8</td>
<td>Impact of false alarm probability on estimation accuracy ($p = q = 0.1$)</td>
<td>43</td>
</tr>
<tr>
<td>3-9</td>
<td>Estimates of PU behavior over frames</td>
<td>44</td>
</tr>
<tr>
<td>3-10</td>
<td>The comparison of theoretical and estimated results</td>
<td>44</td>
</tr>
<tr>
<td>3-11</td>
<td>The confidence level of the proposed algorithm</td>
<td>45</td>
</tr>
<tr>
<td>3-12</td>
<td>The comparison of the confidence level between fixed and proposed algorithm</td>
<td>46</td>
</tr>
<tr>
<td>3-13</td>
<td>The comparison of the relative error between fixed and proposed algorithm</td>
<td>47</td>
</tr>
<tr>
<td>4-1</td>
<td>PU behavior model with imperfect sensing</td>
<td>51</td>
</tr>
<tr>
<td>4-2</td>
<td>Two-state HMM model</td>
<td>54</td>
</tr>
<tr>
<td>4-3</td>
<td>The probability distribution of perfect and imperfect sensing</td>
<td>64</td>
</tr>
<tr>
<td>4-4</td>
<td>The comparison of confidence level between perfect and imperfect sensing</td>
<td>67</td>
</tr>
<tr>
<td>4-5</td>
<td>The impact of transition probabilities on the required sequence length for imperfect sensing</td>
<td>68</td>
</tr>
<tr>
<td>4-6</td>
<td>The impact of transition probabilities on the required sequence length for perfect sensing</td>
<td>69</td>
</tr>
<tr>
<td>4-7</td>
<td>The impact of transition probabilities on the required sequence length ($p=q$)</td>
<td>69</td>
</tr>
</tbody>
</table>
Abstract of Dissertation Presented to the Graduate School of the University of Florida in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

PRIMARY USER BEHAVIOR ESTIMATION AND CHANNEL ASSIGNMENT FOR DYNAMIC SPECTRUM ACCESS IN ENERGY-CONSTRAINED COGNITIVE RADIO SENSOR NETWORKS

By
Xiaoyuan Li

August 2013

Chair: Janise Y. McNair
Major: Electrical and Computer Engineering

Cognitive radio technology improves spectrum utilization by allowing secondary users (SUs) to access the licensed spectrum bands in an opportunistic manner as long as it does not interfere with the activity of the primary users (PUs). This technology may also be used for wireless sensor networks (WSNs) to solve the problem of spectrum scarcity and bursty traffic. With the knowledge of PU behavior, sensors can transmit packets on the channels which are currently not occupied and vacate the bands by the detection of PU signals. In this dissertation, the spectrum sensing and spectrum access problems are investigated in a cognitive radio sensor network (CRSN), in which a cognitive radio is installed in each sensor and it can be tuned to any available channel.

Modeling and estimating the PU behavior is critical to implement dynamic spectrum access. For perfect sensing without sensing errors, we investigate the estimation accuracy of the PU behavior based on the Markov model. The performance of Maximum Likelihood (ML) estimation is evaluated by its distribution. To meet the requirement of estimation accuracy while reducing the unnecessary sensing time, we propose a learning algorithm to dynamically estimate the required length of the sample sequence. For the imperfect sensing with sensing errors, a two-state HMM is employed to model PU behavior with imperfect sensing. Baum-Welch algorithm is used to estimate the
transition probabilities. The estimation accuracy is compared with that of perfect sensing.

Due to the inherent power and resource constraints of sensor networks, energy efficiency is the primary concern for the network design. We investigate the residual energy aware channel assignment problem in a cluster-based multi-channel CRSN. An $R$-coefficient is developed to estimate the predicted residual energy using sensor information (current residual energy and expected energy consumption) and channel conditions (PU behavior). An Optimization-based channel assignment scheme which maximizes the total residual energy of the network is proposed to reduce energy consumption and prolong the network lifetime.

We also consider another important concern for proposing an appropriate opportunistic spectrum access scheme, the total energy consumption needed to successfully transmit a certain amount of information bits. It helps sensors to transmit as much information as possible during their lifetime. We dynamically choose the optimal packet size to minimize energy-per-bit (the ratio of the total energy consumption to the amount of successfully transmitted information bits), which adapts to the time-varying channel states depending on both the behavior of primary users and the activity of sensors. Moreover, we increase the network lifetime by balancing residual energy among sensors.
CHAPTER 1
INSTRUCTION

In this chapter, the background knowledge of a cognitive radio sensor networks (CRSN) is provided, which includes its definition, motivation and design challenges. Then the organization of this dissertation follows.

1.1 Cognitive Radio Sensor Networks

The rapid growth of wireless services leads to the demand for a solution to spectrum scarcity and under-utilization of the licensed bands. Therefore, the Federal Communications Commission (FCC) allows unlicensed users to access the temporarily unused spectrum in an opportunistic manner [1]. Cognitive radio technology enables dynamic spectrum access for the secondary user (SU) by sensing the usage information of the spectrum from the radio environment. The SU with cognitive radio can access the best available spectrum among the licensed bands as long as it does not cause any interference to the primary users (PUs) with a specific license. In this way, the spectrum efficiency of the network will be improved.

Cognitive radio technology may also be used in wireless sensor networks (WSNs). Existing WSNs are traditionally characterized by fixed spectrum allocation over crowded bands [2]. Spectrum scarcity is highlighted because of the ever increasing demand for various wireless networks, such as WiFi and Bluetooth. The event-driven nature often generates bursty traffic, which increases the probability of collision and packet loss. Traditional WSNs lack the ability of adjusting its radio configuration to the dynamic operating environment [3]. Cognitive radio allows opportunistic spectrum access to multiple available channels, which gives potential advantages to WSNs by increasing the communication reliability and improving the energy efficiency. Recent papers, such as [2, 4–14], propose the promising applications of Cognitive Radio Sensor Networks (CRSNs). Similar to traditional WSNs, a CRSN consists of a large number of low-cost low-power sensors with a limited battery energy. In the CRSN, each sensor
is equipped with a cognitive radio, which enables the adaptation of its operating parameters. A sensor selects the most appropriate channel once an available band is identified and vacates the band when a PU’s transmission is detected. The integration of cognitive radio capabilities provides many advantages to WSNs. For example, there are potentially more bandwidth available for sensors. Moreover, bit error rate may be decreased due to the ability to access the best available channel.

1.2 Energy Challenges

There are many challenges for designing communication and networking protocols for a CRSN, such as additional communication and processing requirements with cognitive radio capabilities, transmission power control to avoid interference with PUs, multi-hop opportunistic communications among densely deployed sensors. Above all, the most important one is the inherent energy constraint of the low-capability sensors with unchargeable batteries.

First, since sensor must collect spectrum usage information to opportunistically access the licensed channel which is currently not occupied by PUs, they have to sense spectrums to find spectrum opportunities prior to transmission. Unreliable identification of spectrum opportunities may result in packet loss and retransmission, which is a waste of the energy. However, additional energy consumption is imposed by spectrum sensing and the exchange of sensing results. A dedicated channel is designated to exchange control data such as spectrum sensing result and neighbor information. Since a network-wide common channel may not be possible, a local common control channel is needed within a given locality.

Second, sensors have to analyze the sensing results and make a decision about the best available channel and the corresponding operating parameters. Moreover, since multiple sensors may try to access the same spectrum, a spectrum sharing mechanism is needed to coordinate multiple simultaneous transmissions, which includes both the management of coexistence with PUs and resource allocation among sensors. The
medium access control (MAC) protocols in traditional cognitive radio network focus on QoS performance such as throughput and delays, which do not match the inherent resource constraint of sensors. Spectrum access in CRSNs has to be coordinated to increase both spectrum utilization and energy efficiency.

1.3 Issues and Contributions

In this dissertation, the spectrum sensing and spectrum access problems are investigated in a CRSN, in which a cognitive radio is installed in each sensor. The radio can be tuned to any available channel. Sensors could access the best available channel which is temporarily unused by any PU and stops the transmission immediately after the PUs signals are detected.

1.3.1 Primary User Behavior Estimation

Since the spectrum access of sensors should not cause any interference to PUs, a precise model and estimation of the PU behavior is important to enable efficient dynamic spectrum access. In this dissertation, PU behavior is modeled as a Markov chain and its transition probabilities are estimated using maximum likelihood (ML) estimation. We investigate the estimation accuracy of the PU behavior and the relationship between the accuracy and the processing overhead.

The main contributions of this work are as follows:

1. An expression for the probability mass function (PMF) of the ML estimator is derived to evaluate the accuracy of the estimated transition probabilities. To the best of our knowledge, this is the first work to analyze the performance of the ML estimator by deriving its PMF.

2. We show that the distribution of the ML estimator approximately follows the normal distribution. The estimation accuracy is therefore analyzed by the confidence interval defined on the normal distribution.

3. A learning algorithm which iteratively refines the estimation results is developed for an accurate estimation of the PU behavior. The length of the sample sequence required for a given confidence level is dynamically determined to adapt to the changing PU behavior. It achieves the requirement of estimation accuracy while reducing unnecessary sensing time.
4. A two-state Hidden Markov model (HMM) is utilized to model PU behavior and estimate PU state transition probabilities with sensing error. The relationship among the length of the observed sequence, the real state transition probabilities, the false alarm and miss detection probabilities, the selection of initial parameters and the accuracy of the estimated values is validated.

1.3.2 Residual Energy Aware Channel Assignment

Spectrum access of sensors has to be coordinated to increase spectrum utilization while avoiding interference to PUs. Moreover, collisions among sensors should also be reduced. Therefore, the channel assignment problem should be investigated from the aspect of energy consumption and network lifetime.

The main contributions of this work are as follows:

1. An $R$-coefficient determined by sensor energy information and PU behavior is proposed to represent the predicted residual energy.

2. An Optimization-based channel assignment scheme is proposed which maximizes the total residual energy of the network. It leads to better performance in terms of energy consumption and network lifetime.

1.3.3 Dynamic Spectrum Access with Packet Size Adaptation

The effective energy, which is the energy consumption for the successfully transmitted data is considered for the dynamic spectrum access. It helps sensors to transmit as much information as possible during their lifetime.

The main contributions of this work are as follows:

1. A packet size adaptation scheme for data packets is proposed to improve energy efficiency by minimizing the network energy-per-bit (EPB), which is defined as the ratio of the total energy consumption to the amount of successfully transmitted information bits of the whole network.

2. After the packet size is determined, the channel assignment of CRSN is investigated with the objective of residual energy balancing.

1.4 Organization

The rest of the dissertation is organized as follows.
In Chapter 2, the system models used in this dissertation are described, which includes the network model, the time-slotted frame structure, the PU behavior model and the energy consumption model.

In Chapter 3, the estimation accuracy of PU behavior is studied. To meet the requirement of estimation accuracy while reducing the unnecessary sensing time, we propose a learning algorithm which refines the estimation results iteratively.

In Chapter 4, a two-state HMM is employed to model PU behavior with imperfect sensing. Baum-Welch algorithm is used to estimate the transition probabilities. The estimation accuracy is compared with that of perfect sensing.

In Chapter 5, the channel assignment problem is investigated with the goal of total network residual energy maximization.

In Chapter 6, the dynamic spectrum access scheme with packet size adaptation and residual energy balancing is proposed to improve energy efficiency and prolong the network lifetime.

In Chapter 7, we conclude the current work and describe the work remaining for this dissertation.
In this chapter, we describe the system models used in each work. The chapter is organized as follows. Section 2.1 introduces the network model and the time-slotted frame structure. Section 2.1.1 describes the time structure used for spectrum sensing. Section 2.1.2 describes the cluster-based network model and the time structure for channel assignment. Section 2.2 introduces the Markov model of PU behavior. The energy consumption model is introduced in Section 2.3.

2.1 Frame Structure and Network model

2.1.1 Spectrum Sensing Model

The system is time-slotted with $K$ slots in a frame. The length of a frame is assumed to be short enough so that the PU behavior remains unchanged within the duration of a frame. Each frame consists of a channel sensing phase, which takes the first $N$ slots, and a channel access phase, which can take the remaining $K-N$ slots. The slotted structure is shown in Figure 2-1.

![Figure 2-1. Time-slotted structure of a frame](image)

In the channel sensing phase, channel occupancy according to a PU is monitored to form a sample sequence of length $N$. The PU behavior is estimated based on the sequence. The SU transmits data packets in the channel access phase in light of the estimation results.
There is an inherent tradeoff in the frame structure between the number of time slots for channel sensing and channel access. An increase in the channel sensing time improves the estimation accuracy. However, it also decreases the data transmission time in the channel access phase. In addition, the energy consumption for channel sensing and the memory cost for the storage of sensing results should be minimized. Therefore, the length of the sample sequence should be carefully selected to improve the overall performance.

2.1.2 Channel Assignment Model

Figure 2-2. A cluster-based CRSN with PU

The PU network and the CRSN are generally unrelated in terms of communication. They coexist in the same area as shown in Figure 2-2. PUs are either static or mobile nodes with high transmission power. Sensors are assumed to be static or move infrequently inside the range of a cluster. The resource and capability constraints limit the spectrum sensing capability of sensors. Moreover, the network-wide common control channel which plays an essential role in general cognitive radio networks may not be feasible in a low-power large-area CRSN [2]. Therefore, we propose a
cluster-based multi-channel CRSN, in which each cluster has a cluster head (CH) and a fixed local common control channel. The CH is a special energy-rich sensor node with high cognitive radio capabilities for spectrum sensing and channel assignment among its cluster members (CMs). The local common control channel is introduced to exchange control information for channel assignment and network maintenance.

We assume there are $M$ different data channels and one common control channel in each cluster. At each time, a data channel can only be assigned to one sensor and a sensor can only transmit on one data channel. CMs send data packets to CH on data channels. CH collects data from its members and sends the processed data to the base station via the cluster head backbone. In this work, we only consider channel assignment for intra-cluster communication and energy consumption during data transmission from CMs to CH. Inter-cluster performance will be discussed in our future work. Therefore, the concept network in this work means the range of a cluster.

![Time-slotted structure of a frame](image)

**Figure 2-3.** Time-slotted structure of a frame

The system is time-slotted with $K+1$ time slots in a frame. Each frame consists of a channel assignment phase, which takes the first slot, and a data transmission phase, which can take the remaining $K$ slots. The slotted structure is shown in Figure 2-3. CH monitors PU activity on each channel periodically and estimates spectrum availability based on PU statistics. In each time slot, CMs will be in one of the three states, listen, transmit or sleep. In channel assignment phase, CM sensors that need to transmit wake up and inform CH by sending a low bit-rate assign_request message via the common
control channel. Then they turn to listen state in this phase for assign_reply messages from CH. These sensors are called active sensors and the probability of CM sensors being active is denoted as $P_{\text{active}}$. The assign_request message includes the residual energy and the location information of the CM for calculation of predicted residual energy, which will be explained later.

After receiving all the assign_request messages, CH carries out channel assignment, which will be discussed in Chapter 5 and Chapter 6. Then CH broadcasts the assign_reply messages with channel assignment results. If a CM does not get assigned to any channel, it will turn to sleep state to save power. CM who gets assigned tunes its radio to the designated channel and starts to transmit the event data to CH in data transmission phase. However, it will stop the transmission immediately after detecting PU signals on the same channel and the ongoing transmitted packet will be dropped. When it stops or finishes transmission, it turns to sleep and wakes up when another transmission is required.

## 2.2 Primary User Behavior Model

The PU behavior is modeled in a two-state Markov chain, where the presence and absence of PU signals are represented as busy and idle states, respectively, as shown in Figure 2-4.

![Figure 2-4. A two-state Markov model of PU behavior](image)

Let $p$ denote the probability that the channel state changes from idle to busy. Symmetrically, let $q$ denote the probability that the channel state changes from busy to...
The probability that the channel is idle and busy can be obtained by deriving the steady state probabilities for the model shown in Figure 2-4.

\[
\begin{align*}
P_{\text{idle}} &= \frac{q}{p+q} \\
\frac{q}{p+q} \\
P_{\text{busy}} &= \frac{p}{p+q}
\end{align*}
\]  

(2-1)

The PU behavior is therefore determined by the transition probabilities. The SU makes the decision on channel selection by searching for the best available channel based on these parameters. The details will be discussed in the following chapters.

2.3 Energy Consumption Model

In this dissertation, since CH is assumed to be rich in energy, we only consider CMs’ energy consumption for both control message exchange and data transmission. There are \( N \) sensors deployed in a cluster and each sensor carries a non-rechargeable battery with the same initial energy \( E_{\text{in}} \). The communication channel can be considered as a channel following a simple path loss model, where fading and multipath effects are ignored [15]. The energy consumed in data transmission is \( E_{\text{cir}} + \varepsilon d^\alpha \), where \( E_{\text{cir}} \) is circuit energy consumption and \( \varepsilon \) is the amplifier energy required at the receiver, both of which are measured per bit. \( d \) is the distance between CM and CH and \( \alpha \) is the path loss coefficient depending on the path characteristics. In this work, a free space model is considered for signal degradation, in which \( \alpha \) is equal to 2.

Therefore, if sensor \( i \) continuously transmits for \( l \) slots, the total energy consumption is calculated as follows.

\[
E_{tr}^i(l) = (E_{\text{cir}} + \varepsilon d_i^2) \times B \times T \times l
\]

(2-2)

Where \( B \) is the transmission rate in \( \text{bit/s} \) and \( T \) is the length of a slot period in \( \text{second} \). \( E_{\text{cir}} \) is in \( \text{nJ/bit} \). \( \varepsilon \) is in \( \text{pJ/bit/m}^2 \).
Since CMs receive the broadcast messages from CH, energy consumed for the reception also needs to be considered and it is denoted by

\[ E_{i}^{\nu}(l) = E_{cir} \times B \times T \times l \]  \hspace{1cm} (2-3)
CHAPTER 3
PRIMARY USER BEHAVIOR ESTIMATION WITH PERFECT SENSING

In this chapter, we investigate the estimation accuracy of the PU behavior based on the Markov model. Maximum Likelihood (ML) estimation is employed to estimate the transition probabilities of the Markov model based on the sample sequence of PU idle/busy states. An approximate distribution of the ML estimator is derived to evaluate the estimation accuracy specified by the confidence level. To meet the requirement of estimation accuracy while reducing the unnecessary sensing time, we propose a learning algorithm which refines the estimation results iteratively. It dynamically estimates the required length of the sample sequence which is adaptive to the changing PU behavior.

This chapter is organized as follows. Section 3.1 introduces an overview of the PU estimation problems. Section 3.2 introduces the recent studies related to PU behavior. Section 3.3 introduces the ML estimator, followed by the derivation of its PMF and the approximation of its normal distribution. The analysis of the estimation accuracy and the required length of the sample sequence are discussed in Section 3.4. The estimation algorithm with adaptive length of the sample sequence is proposed in Section 3.5. Section 3.6 provides the numerical results. Section 3.7 concludes this chapter.

3.1 Overview of PU Estimation Challenges

A key functionality of cognitive radio devices is to sense the radio environment before they access the licensed spectrum. The spectrum sensing result is used to understand how the PUs use the spectrum. Therefore, accurate spectrum sensing is important for SUs to avoid any interference to PUs. Unreliable identification of spectrum availability would result in collisions, packet losses and unnecessary delays, which degrades the overall performance. A precise model and estimation of the PU behavior is thus needed for prediction of the future channel states to help improve spectrum utilization.
The PU behavior has been assumed to follow the Markov model in recent studies \cite{16-22}. The channel occupancy of PU at any time slot is considered as a state, which can be either busy or idle. The Markov model provides the information for the prediction of future states based on the current observations. If the PU behavior is known, the SU could make the appropriate decision on channel access and proactively vacate the channel even before detecting any signal from the PU. However, there are several challenges imposed by this methodology.

First, in a cognitive radio network, an SU may not know the PU behavior in advance. It keeps sensing the channel over consecutive time periods and stores all the channel states to form a sample sequence. Then the transition probabilities of the Markov model for the PU behavior are estimated based on the sample sequence. Without knowing the model parameters, the SU may cause harmful interference to PUs and the performance of the SU itself may also be greatly affected.

Second, the sample sequence should be long enough to achieve certain precision of the estimation. However, both the energy wasted for performing spectrum sensing and the memory used for storing the samples are expected to be kept as low as possible for the SU. Moreover, the SU cannot transmit data packets when it senses the channel. The time wasted on the channel sensing should be reduced to improve channel utilization.

Third, the PU behavior may vary over time due to the changing PU traffic density \cite{22, 23}. Thus, the PU behavior estimate has to be updated accordingly. Moreover, the required number of states in the sample sequence needed for an accurate estimation of the model may also differ greatly. The SU should perform the channel estimation using an online algorithm with the varied length of the sample sequence to reduce unnecessary sensing time.

Due to the above concerns, a precise estimator of the PU behavior is needed to enable efficient dynamic spectrum access of SUs. Maximum likelihood (ML) estimation
is commonly used to estimate the state transition probabilities of the Markov model. The accuracy of the ML estimation has to be enforced for the SU to obtain a proper knowledge of the PU behavior before accessing the spectrum. In this work, the exact distribution of the ML estimator is derived to analyze the relationship among the length of the sample sequence, the state transition probabilities and the accuracy of the estimates. We show that the distribution of the ML estimator approximately follows the normal distribution. The estimation accuracy is therefore analyzed by the confidence interval defined on the normal distribution. The required length of the sample sequence can therefore be determined for any given accuracy requirement. A learning algorithm which iteratively refines the estimation results is developed for an accurate estimation of the PU behavior. The length of the sample sequence is dynamically determined to adapt to the changing PU behavior.

### 3.2 Related Work

In most of the recent studies on cognitive radio networks, the channel occupancy of PUs has been considered as a two-state Markov model [16–22]. In [16, 20, 21], the transition probabilities are assumed to be known to the SU. However, in real applications, it is very difficult for an SU to obtain these parameters in advance. It has to estimate the PU behavior based on the current observations. In [19], the channel usage pattern of PUs is assumed to be static, which is also not practical in a changing radio environment. The SU has to obtain new samples and re-estimate the transition probabilities according to variations of the PU behavior. ML estimation [24] is used for estimating the transition probabilities of the Markov chain in [17] and [18]. The transition probabilities are determined by maximizing the probability of the current observations. However, the performance analysis of the ML estimation is not considered and how to decide the required number of samples is not mentioned in [17] and [18]. To the best of our knowledge, our work is the first work to estimate the length of the sample sequence required for any given accuracy requirement of the ML estimator using its distribution.
3.3 The Distribution of Maximum Likelihood Estimator

The two-state Markov chain is used to model the PU behavior in this work, as described in Chapter 2, Section 2.2. In real applications, the transition probabilities in Equation (2–1) may not be known a priori and they need to be estimated based on the observations of the PU behavior. The states of the \( N \) most recent slots on the channel form a sample sequence denoted by \( S = \{s_1, s_2, \cdots, s_N\} \). It is a binary sequence with 0 and 1 representing idle and busy state, respectively. Using ML estimation [24], the estimators of transition probabilities \( p \) and \( q \) are derived as follows:

\[
\begin{align*}
\hat{p} &= \frac{n_{01}}{n_0} \\
\hat{q} &= \frac{n_{10}}{n_1},
\end{align*}
\]

(3–1)

where \( \hat{p} \) and \( \hat{q} \) denotes the estimated value of \( p \) and \( q \), respectively. \( n_{ij} \ (i, j \in \{0, 1\}) \) represents the number of state transitions from state \( i \) to state \( j \). \( n_i \ (i \in \{0, 1\}) \) denotes the number of all transitions from state \( i \).

Note that \( n_{01} \) and \( n_0 \) should satisfy:

\[
n_{01} \leq \min(n_0, N-n_0),
\]

(3–2)

where \( N \) is the number of states in the sample sequence.

**Proof**: Based on the definitions of \( n_{01} \) and \( n_0 \), \( n_{01} \leq n_0 \). The number of all transitions from state 1 is \( n_1 = n_{10} + n_{11} \). \( n_0 + n_1 \) is equal to the number of total transitions \( N-1 \).

Assume \( n_{01} > N - n_0 \), we have

\[
n_{01} + n_0 > N \\
\iff n_{01} + n_0 > n_0 + n_1 + 1 \]

(3–3)

\[
\iff n_{01} > n_1 + 1
\]
The number of the occurrences of state 1 is equal to \( n \) if the last state is state 0 or \( n + 1 \) otherwise. Because each \( 0 \rightarrow 1 \) transition involves a state 1, \( n_{01} \) cannot be greater than \( n + 1 \). This contradicts the constraint in Equation (3–3). Similarly, \( n_{10} \leq \min(n_1, N - n_1) \).

### 3.3.1 Derivation of the Probability Mass Function (PMF)

The PMF of the ML estimator is derived to evaluate its performance in this section. Denote \( Pr(\hat{p} = x) \) as the probability that \( p \) takes value \( x \) and it is calculated as follows. Denote \( Pr(\hat{q} = x) \) as the probability that \( q \) takes on value \( x \) and it could be derived similarly.

The observed sequence with \( N \) state samples is \( S = \{s_1, s_2, \cdots, s_N\} \) and hence the total number of all possible sequences is \( 2^N \). It is computationally overwhelming to enumerate each sequence and calculate its corresponding transition probability estimate. Instead, we search for an analytical way to calculate \( Pr(\hat{p} = x) \) by grouping all the sequences that lead to the same \( x \), which is the ratio of \( n_{01} \) to \( n_0 \). That means \( x \) is the same for all sequences within the corresponding group. Then, the PMF can be obtained by summing all the individual occurrence probability values of the sequences within that group. For example, \( Pr(\hat{p} = 0.5) = \sum_{k=1}^{\min(n_0, n-N)} Pr(\hat{p} = \frac{k}{2k}) \) where \( Pr(\hat{p} = \frac{k}{2k}) \) represents the probability of occurrence of all sequences with \( n_{01} = k \) and \( n_0 = 2k \).

The probability \( Pr(\hat{p} = x) \) with the given \( n_{01} \) and \( n_0 \) is derived as follows. Denote a parameter set by \( \chi = (s_1, s_N, n_0, n_{01}) \), in which \( s_1 \) represents the first state of the observed sequence, \( s_N \) represents the last state, \( n_0 \) and \( n_{01} \) denotes the number of transitions starting with 0 and the number of transitions from 0 to 1, respectively. Define \( Q(\chi) \) as a function of the parameter set \( \chi \), which represents the individual occurrence probability of the sequences with a given \( \chi \). \( N_{\text{total}}(\chi) \) represents the total number of sequences with the same \( \chi \). \( T(\chi) \) represents the total occurrence probability of the sequences with the given \( \chi \) and it is calculated by:

\[
T(\chi) = Q(\chi) N_{\text{total}}(\chi),
\]  

(3–4)
In the above equation, $Q(\chi)$ with $\chi = (1, 0, n_0, n_{01})$ is calculated by:

\[
Q(\chi) = P_{busy} p_{n_{01}} q_{n_0}^{n_0 + 1} \left(1 - p\right)^{n_0 - n_{01}} (1 - q)^{N - n_0 - n_{01} - 2}.
\] (3–5)

Note that in this case, the relationship of $N, n_0, n_{01}$ must satisfy the following constraints:

\[
n_{01} \leq \min(n_0, N - n_0 - 2).
\] (3–6)

The proof is skipped as it is similar to the proof of Equation (3–2).

Next we will describe the calculation of the number of sequences with the same individual occurrence probability $Q(\chi)$. Because the sequence ends with state 0, the total number of states 0 is $n_0 + 1$. Denote the number of states 0 before the last state 1 by $m$. Therefore there are $n_0 + 1 - m(m \leq n_0)$ consecutive zeros at the end. An example of the state sequence is shown as follows:

\[
S_j = \{1 \cdots 0 \cdots 0 \cdots 0 \cdots 0 \cdots 0 \cdots 0 \cdots 1 \cdots 0 \cdots 00\}.
\] (3–7)

The number of groups $\overbrace{0 \cdots 01}$ is $n_{01}$ and the number of zeros in each $\overbrace{0 \cdots 01}$ is in the range of $[1, m - (n_{01} - 1)]$.

The $m$ zeros should be allocated to $n_{01} \overbrace{0 \cdots 01}$ groups and the number of allocations is calculated by

\[
N_{alloc}(\chi) = \sum_{l=1}^{m-(n_{01}-1)} \binom{m-l}{n_{01}-2} = \binom{m-1}{n_{01}-1}.
\] (3–8)
Then, for each allocation, the number of placements of \( \overline{0 \cdots 01} \) in the state sequence is

\[
N_{\text{comb}}(\chi) = \binom{N - n_0 - 2}{n_{01}}.
\] (3–9)

Note that according to the constraint in Equation (3–6), \( n_{01} \) is guaranteed to be less than or equal to \( N - n_0 - 2 \).

Since \( m \in [n_{01}, n_0] \), the total number of the sequences with the same occurrence probability \( Q(\chi) \) is calculated by

\[
N_{\text{total}}(\chi) = \sum_{m=n_{01}}^{n_0} N_{\text{alloc}}(\chi) N_{\text{comb}}(\chi)
= \sum_{m=n_{01}}^{n_0} \binom{m-1}{n_{01}-1} \binom{N - n_0 - 2}{n_{01}}
= \binom{n_0}{n_{01}} \binom{N - n_0 - 2}{n_{01}}.
\] (3–10)

Therefore, \( T(\chi) \) with \( \chi = (1, 0, n_0, n_{01}) \) can be obtained by Equation (3–4).

\( T(0, 0, n_0, n_{01}), T(0, 1, n_0, n_{01}), T(1, 1, n_0, n_{01}) \) could be derived in a similar way.

Note that if \( n_{01} = N - n_0 - 1 \), the state sequence with \( (s_1, s_N) = (1, 0) \) does not exist. Moreover, if \( n_{01} = N - n_0 \), the sequence with \( (s_1, s_N) = (0, 0) \), \( (s_1, s_N) = (1, 0) \) and \( (s_1, s_N) = (1, 1) \) does not exist. \( n_{01} \) can not be greater than \( N - n_0 \) according to the constraint in Equation (3–2). In this way, the complete expression for the occurrence
The probability of all the sequences with the same $n_0$ and $n_{01}$ is obtained by

$$
F(n_0, n_{01}) = \begin{cases} 
\sum_{(s_1, s_N) \in \{00, 01, 10, 11\}} T(s_1, s_N, n_0, n_{01}), & 1 \leq n_{01} \leq N - n_0 - 2 \\
\sum_{(s_1, s_N) \in \{00, 01, 11\}} T(s_1, s_N, n_0, n_{01}), & n_{01} = N - n_0 - 1 \\
T(0, 1, n_0, n_{01}), & n_{01} = N - n_0 
\end{cases}
$$

(3-11)

The following is another way of deriving the expression of $F(n_0, n_{01})$, which is more efficient. In the calculation of $N_{\text{total}}$, the first and the last state are both considered with the parameter set $\chi$. However, it can be seen from the above description that only $s_N$ affects the value of $N_{\text{total}}$. If we assume a $0 \rightarrow 0$ transition when $s_N = 0$ and $1 \rightarrow 1$ transition when $s_N = 1$, then $n_0$ (the number of all transitions from state 0) is the number of states 0 in the sequence and $n_1$ is the number of states 1. In this way, the effect of $s_N$ can be ignored and $N_{\text{total}}$ is determined only by $n_0$ and $n_{01}$. Finding $N_{\text{total}}(n_0, n_{01})$, which is the approximate number of sequences with the given $n_0$ and $n_{01}$ is also a combinatorics problem.

First, with $n_0$ states 0, we select $n_{01}$ of them to form the transition from state 0 to state 1. Then, $N - n_0 - n_{01}$ states 1 are left to insert into the sequence. Since there are already $n_{01}$ transitions from state 0 to state 1, the remaining states 1 can only be inserted after any of the $n_{01}$ states 1 or at the beginning of the sequence, which are $n_{01} + 1$ possible positions in total. This is essentially the stars and bars problem of combinatorial mathematics [25] with $N - n_0 - n_{01}$ stars and $n_{01} + 1$ bars. So the total number is:
\[ N_{\text{total}}(n_0, n_{01}) = \begin{pmatrix} n_0 \\ n_{01} \end{pmatrix} \begin{pmatrix} N - n_0 \\ n_{01} \end{pmatrix}. \quad (3-12) \]

\[ Q(\chi) \text{ is derived in the same way as Equation (3–5). Therefore, } F(n_0, n_{01}) \text{ is expressed by:} \]

\[
F(n_0, n_{01}) = \begin{cases} 
N_{\text{total}}(n_0, n_{01}) \sum_{(s_1, s_n) \in \{00, 01, 10, 11\}} Q(s_1, s_n, n_0, n_{01}), & 0 \leq n_{01} \leq n - n_0 - 2 \\
N_{\text{total}}(n_0, n_{01}) \sum_{(s_1, s_n) \in \{00, 01, 11\}} Q(s_1, s_n, n_0, n_{01}), & n_{01} = n - n_0 - 1 \\
N_{\text{total}}(n_0, n_{01}) Q(0, 1, n_0, n_{01}). & n_{01} = n - n_0. 
\end{cases} \quad (3-13) 
\]

Define \( x = \frac{n_{01}}{n_0} \), the PMF of \( \hat{p} \) is expressed as follows.

\[
\text{Prob}(\hat{p} = x) = \sum_{(n_0, n_{01})} F(n_0, n_{01}) \quad (3-14) 
\]

A special case is \( n_0 = 0 \). In this case, the state sequence consists of \( N - 1 \) number of consecutive 1 with the last state unknown and its probability is \( P_{\text{busy}}(1 - q)^{N-2} \). The estimate of the transition probability \( \hat{p} \) is defined 1.

The PMF of \( \hat{p} \) when \( p = 0.5, q = 0.5 \) is shown in Figure 3-1.

### 3.3.2 Using Normal Distribution to Approximate the Real Estimation Distribution

Since the time spent on exact distribution evaluation is polynomially contingent on the length of the sample sequence, it becomes computationally cumbersome to resolve the real estimation distribution when the require sample sequence gets large. Instead, according to our observation on the similarity of the real estimation distribution to the
normal distribution, we choose to take an approximation approach to simplify such distribution evaluation. In what follows, only the estimation of $\hat{p}$ is discussed and the distribution of $\hat{q}$ can be derived similarly.

According to Equation (3–1), $\hat{p} = \frac{n_{01}}{n_0}$, where both $n_{01}$ and $n_0$ are random variables and undetermined. However, when $N$ is large enough, the probability that a sample is in state 0 is stationary and it is calculated by Equation (2–1). Therefore, given $p$ and $q$, the number of states 0 in the sample sequence is determined by $n_0 = NP_{idle}$. The problem of deriving the distribution of $\hat{p}$ is thus converted to deriving the distribution of $n_{01}$, which is discussed as follows.

The set of states 0 can be considered as $n_0$ independent Bernoulli trials. For each state 0, it generates a transition to state 1 with the probability $p$. The probability of getting exactly $k$ transitions from state 0 to state 1 in these $n_0$ trials is obtained by the
probability mass function of a binomial distribution.

$$\text{Prob}(n_0 = k) = \binom{n_0}{k} p^k (1 - p)^{n_0 - k}$$

(3–15)

Since $n_0$ is a binomially distributed random variable, the expected value is $n_0 p$ and the variance is $n_0 p (1 - p)$. It is denoted by $n_0 \sim B(n_0 p, n_0 p (1 - p))$.

If $n_0$ is large enough, a close approximation to the binomial distribution of $n_0$ is given by the normal distribution [26]:

$$n_0 \sim \text{Norm}(n_0 p, n_0 p (1 - p))$$

(3–16)

Figure 3-2. The comparison of binomial and norm distribution

Figure 3-2 shows a comparison of the binomial distribution and normal distribution when $n_0 = 100$ and $p = 0.5$.

A suitable continuity correction has to be applied to the above approximation [27]. For example, if $X$ has a binomial distribution and $Y$ has a normal distribution, and both of them have the same expected value and variance, then

$$P(X \leq x) = P(X < x + 1) \approx P(Y \leq x + 0.5)$$

(3–17)
The addition of 0.5 is the continuity correction; the uncorrected normal approximation gives considerably less accurate results. This equation will be used later for the analysis of the estimation accuracy.

According to the ML estimator in Equation (3-1), \( \hat{p} = \frac{n_{01}}{n_0} \). Since \( n_0 \) is determined, \( \hat{p} \) is also approximately normally distributed with the mean \( p \) and the variance \( \frac{p(1-p)}{n_0} \).

\[ \hat{p} \sim \text{Norm}(p, \frac{p(1-p)}{n_0}) \]  

(3–18)

The approximate normal distribution is compared with the exact distribution in Figure 3-3. The curve of exact distribution reflects the summation of probability mass function within each 0.025 subspace in the interval of [0, 1]. For normal distribution, we plot the probability on the mid-point value of each subspace for demonstration. It is noted that as the length of the sample sequence \( N \) increases, the two distributions become more close to each other. This means when \( N \) is very large, the normal distribution provides a good approximation of the exact distribution.

### 3.4 The Analysis of the Estimation Accuracy

#### 3.4.1 Confidence Interval

Our goal is to achieve the estimation accuracy of the ML estimator with the minimum number of samples. In this chapter, the estimation accuracy is evaluated in terms of the confidence interval. A confidence interval is specified with two parameters: a confidence level \( 1 - \alpha \) and an error bound \( \beta \). The accuracy requirement of the estimator is defined as the probability that the true value of the transition probability \( p \) is in the interval \( [\hat{p} - \beta \hat{p}, \hat{p} + \beta \hat{p}] \) is at least \( 1 - \alpha \):

\[ \text{Prob}(\hat{p} - \beta \hat{p} \leq p \leq \hat{p} + \beta \hat{p}) \geq 1 - \alpha, \ 0 < \alpha, \beta < 1. \]  

(3–19)

We introduce how to use the continuity correction to calculate the probability of the above definition as follows. Define two random variables \( X \) and \( Y, X \sim B(\mu, \sigma) \),

34
Figure 3-3. The comparison of exact and norm distribution ($p = 0.5, q = 0.5$)

$Y \sim \text{Norm} (\mu, \sigma)$. The confidence interval of $\mu$ in the binomial distribution is

$$[X - \sigma X_\alpha, X + \sigma X_\alpha]. \quad (3–20)$$

Suppose we have

$$\text{Prob}(-X_\alpha \leq \frac{X - \mu}{\sigma} \leq X_\alpha) = 1 - \alpha. \quad (3–21)$$

According to the continuity correction in Equation (3–17), the relationship between the cumulative distribution of $X$ and $Y$ is

$$1 - \frac{\alpha}{2} = \text{Prob}(\frac{X - \mu}{\sigma} \leq X_\alpha)$$

$$= \text{Prob}(\frac{Y - \mu}{\sigma} \leq X_\alpha + 0.5)$$

$$= \Phi(X_\alpha + 0.5), \quad (3–22)$$
where $\Phi(\cdot)$ stands for the standard normal cumulative distribution function. Therefore, $X_\alpha + 0.5$ is the $1 - \alpha$ percentile for the standard normal distribution. For example, when $1 - \alpha = 95\%$, $X_\alpha + 0.5 = 1.96$, $X_\alpha = 1.46$.

Therefore, the confidence interval of $\hat{\rho}$ is

$$[\hat{\rho} - \sqrt{\frac{p(1-p)}{n_0}} X_\alpha, \hat{\rho} + \sqrt{\frac{p(1-p)}{n_0}} X_\alpha].$$ (3–23)

### 3.4.2 The Required Length of the Sample Sequence

In this section, the relationship among the length of the sample sequence, the PU behavior specified by the transition probabilities and the corresponding estimation accuracy is studied using the analysis of the confidence interval.

The definition of the confidence interval in Equation (3–19) can be rewritten as

$$
\text{Prob}(p - \frac{\beta}{1+\beta} \rho \leq \hat{\rho} \leq p - \frac{\beta}{1-\beta} \rho) \geq 1 - \alpha.
$$ (3–24)

Since $\frac{\beta}{1-\beta} p \leq \frac{\beta}{1+\beta} p$, the above confidence level can be guaranteed if the following inequality holds:

$$
\text{Prob}(p - \frac{\beta}{1+\beta} \rho \leq \hat{\rho} \leq p - \frac{\beta}{1-\beta} \rho) \geq 1 - \alpha.
$$ (3–25)

Similarly, the probability in Equation (3–21) can be rewritten as

$$
\text{Prob}(p - \sqrt{\frac{p(1-p)}{n_0}} X_\alpha \leq \hat{\rho} \leq p + \sqrt{\frac{p(1-p)}{n_0}} X_\alpha).
$$ (3–26)

Therefore, the following condition has to be satisfied for the confidence level of $1 - \alpha$:

$$
\sqrt{\frac{p(1-p)}{n_0}} X_\alpha \leq \frac{\beta p}{1+\beta}.
$$ (3–27)

It can be rewritten as:

$$
n_0 \geq X_\alpha^2 (\frac{1}{\beta} + 1)^2 (\frac{1}{\rho} - 1).
$$ (3–28)
If the length of the sample sequence $N$ is large enough, $n_0 = NP_{idle} = \frac{q}{p + q}$. Therefore, the length of the sample sequence with a confidence level of $p$, which is denoted by $N_p$, is calculated by

$$N_p = n_0 \left(\frac{p}{q} + 1\right) \geq X^2_\alpha \left(\frac{1}{\beta} + 1\right)^2 \left(\frac{1}{p} - 1\right) \left(\frac{p}{q} + 1\right).$$ (3–29)

Similarly, for the estimation of $q$, if $\alpha$ and $\beta$ are given, the required number of states 1 in the sample sequence is derived as follows.

$$n_1 \geq X^2_\alpha \left(\frac{1}{\beta} + 1\right)^2 \left(\frac{1}{q} - 1\right).$$ (3–30)

Since $n_1 = NP_{busy} = \frac{p}{p + q}$, the length of the sample sequence with a confidence level of $q$, which is denoted by $N_q$, is calculated by

$$N_q = n_1 \left(\frac{q}{p} + 1\right) \geq X^2_\alpha \left(\frac{1}{\beta} + 1\right)^2 \left(\frac{1}{q} - 1\right) \left(\frac{q}{p} + 1\right).$$ (3–31)

Given $1 - \alpha$ and $\beta$, in order to guarantee the number of states 0 ($n_0$) and the number of states 1 ($n_1$) satisfy Equation (3–28) and Equation (3–30), respectively, the minimum required length of the sample sequence $N$ is

$$N_{min} = \max(N_p, N_q).$$ (3–32)

Hence we theoretically compute the minimum length of sample sequence required for the given transition probabilities and a certain confidence level specified by $1 - \alpha$ and $\beta$. Figure 3-4 is a 3D graph of the relationship between $N$ and $p, q$ when $1 - \alpha = 95\%$ and $\beta = 0.1$. It shows that the required length of sample sequence increases as the transition probabilities decreases.

3.5 The Adaptation of the Sample Sequence Length

The required length of the sample sequence differs greatly according to varied transition probabilities as in Figure 3-4. Since the PU behavior specified by the transition probabilities of the Markov model varies over time, the length of the sample sequence
Figure 3-4. The impact of transition probabilities on the required sample sequence length

should be selected dynamically over each frame. According to Figure 3-4, when the transition probabilities decreases, the length of the sample sequence should be increased for a better estimation accuracy. On the other hand, the length of the sequence can be reduced with the increased transition probabilities. To this end, an algorithm which estimates the transition probabilities of the sample sequence with the required length should be conducted at the beginning of each frame to adapt to the changing PU behavior.

Since the transition probabilities of PU behavior are not known by the SU in advance, Equation (3–32) cannot be exploited directly to derive the required number of states in the sequence. Therefore, we develop a learning algorithm which initializes the process with a coarse estimate of the transition probabilities for the Markov model and then learns the model iteratively until a given accuracy requirement is satisfied.
Suppose $\hat{p}_0$ and $\hat{q}_0$ as the estimate of $p$ and $q$ from the previous frame. The estimation accuracy requirement is specified by the confidence level $1 - \alpha$ and the error bound $\beta$. In the first step, the required length of the sample sequence has to be determined. $\hat{p}_0$ and $\hat{q}_0$ are used as the initialized estimates and they substitute for $p$ and $q$ in Equation (3–29) and Equation (3–31). Therefore, the required length of the sample sequence for an accurate estimation based on $\hat{p}_0$ and $\hat{q}_0$ is calculated by Equation (3–29) and Equation (3–31), respectively. The required number of samples for the first iteration $N_1$ is derived by Equation (3–32). In the second step, the channel occupancy state is monitored for $N_1$ consecutive slots to form a sample sequence. Equation (3–1) is used to generate the estimate $\hat{p}_1$ and $\hat{q}_1$. The process stops if the termination condition is met. Otherwise, it returns to the first step with $\hat{p}_1$ and $\hat{q}_1$ substituting for $p$ and $q$. In this way, the learning algorithm updates the estimation result after each iteration.

![Figure 3-5. The flow chart of the proposed learning algorithm](image)

An important functionality of the iterative learning algorithm is to determine whether the termination condition is satisfied. It is discussed as follows. Let $\hat{p}_{i-1}$ and $\hat{q}_{i-1}$ be
the substitute for \( p \) and \( q \) at the \( i \)th iteration, respectively. The required length of the sequence is denoted as \( N_i \). The estimation results are \( \hat{p}_i \) and \( \hat{q}_i \). The confidence interval of \( \hat{p}_i \) is

\[
[\hat{p}_i - \sqrt{\frac{\hat{p}_{i-1}(1 - \hat{p}_{i-1})}{n_0}} X_\alpha, \hat{p}_i + \sqrt{\frac{\hat{p}_{i-1}(1 - \hat{p}_{i-1})}{n_0}} X_\alpha].
\] (3–33)

To meet the requirement of the confidence level, the number of states 0 in the sample sequence \( n_0 \) should satisfy:

\[
\sqrt{\frac{\hat{p}_{i-1}(1 - \hat{p}_{i-1})}{n_0}} X_\alpha \leq \hat{p}_i \beta.
\] (3–34)

Similarly, the number of states 1 \( n_1 \) should satisfy:

\[
\sqrt{\frac{\hat{q}_{i-1}(1 - \hat{q}_{i-1})}{n_1}} X_\alpha \leq \hat{q}_i \beta.
\] (3–35)

The iterative process terminates when both conditions are satisfied. The overall operation of the proposed learning algorithm is summarized in Figure 3-5. In this way, the length of the sample sequence adapts to the changing PU behavior so that the estimation accuracy can be achieved while the unnecessary sensing slots are reduced.

### 3.6 Simulation Results

#### 3.6.1 Estimation Accuracy of Exact PMF

The estimation accuracy is evaluated by the standard deviation, which is determined by both real transition probabilities and the number of samples. The impact of the transition probabilities on the standard deviation is depicted in Figure 5-4. This is a theoretical result on the relationship among the standard deviation of \( \hat{p} \) and real values of \( p, q \), when \( N = 50 \) and 100, respectively. It is noted that \( p \) and \( q \) have different impacts on the standard deviation of \( \hat{p} \). For a fixed \( q \), the standard deviation of \( \hat{p} \) takes on small values when \( p \) stays close to its two extreme points (0 and 1). This is because when \( p \) moves towards 0.5, the PMF of \( \hat{p} \) has its left-side lobe and right-side lobe more evenly and freely distributed towards its low end (0) and high end (1), as shown in
Figure 3-6. Impact of transition probabilities on estimation accuracy of $\hat{p}$

Figure 3-1. This renders a high standard deviation compared with $p$ moving towards either low end or high end of $[0, 1]$, due to the spatial restriction placed on the latter PMF. Therefore the standard deviation has its maximum value at $p = 0.5$ if $q$ is fixed. For a fixed $p$, the standard deviation decreases monotonously as $q$ increases. Since $P_{idle} = \frac{q}{p+q} = 1 - \frac{p}{p+q}$, bigger $q$ renders higher probability for the Markov model to stay in the idle state. As a result, a bigger $n_0$ is expected and so is a more accurate $\hat{p}$.

In Figure 3-7, the curve with the circle illustrates the impact of the number of samples on the estimation accuracy when $p = 0.1$, $q = 0.1$. It is shown that as the number of samples increases, the standard deviation decreases monotonously. Therefore the required number of samples for an accurate estimation could be obtained if the transition probabilities and the standard deviation are given. Even though the real transition probabilities are not known in advance, the estimated transition probabilities from the previous time frame can be used to substitute the real values since PU
behavior is very likely to be similar over consecutive frames for many applications [22].

Long et al. [28] also analyze the relationship between the estimation accuracy and the number of samples. An approximate normal distribution is derived for the ML estimator. Figure 3-7 compares the standard deviation of the exact distribution and the approximate normal distribution. It can be seen that there is a visible difference between the two distributions when the number of samples is less than 100. Since the normal distribution has smaller standard deviation, it results in derivation of less number of samples than what is theoretically required for a precise estimation. Therefore, the approximate distribution is not accurate for the estimation with a small number of samples, and it is less applicable to the SU with low capability.

The accuracy of the estimated transition probabilities in this letter is investigated under the assumption that all the PU states are accurately detected. Since the PU detection may be affected due to interference/noise, the estimation accuracy needs
to be examined under the error detection. The false alarm probability is introduced to specify how the interference/noise affects the PU detection. Figure 3-8 shows the impact of the false alarm probability on the standard deviation of the estimation. It is shown that the estimation accuracy decreases as the false alarm probability increases. However, there is only a slight difference between each curve, which means the proposed approach for analyzing the estimation accuracy is still applicable when the false alarm probability stays below $10^{-2}$.

### 3.6.2 Estimation Accuracy of Adaptation Algorithm

The performance of proposed algorithm is evaluated through numerical results. Some parameters used in the simulation are as follows: The length of a frame is 200 ms and each slot lasts 20 $\mu$s, therefore there are 10000 slots in a frame [22]; The PU behavior according to the transition probabilities changes every 200 frames; The accuracy requirement in terms of the confidence level is given as $1 - \alpha = 0.95$ and $\beta = 0.1$. 

![Impact of false alarm probability on estimation accuracy](image)
Figure 3-9. Estimates of PU behavior over frames

Figure 3-10. The comparison of theoretical and estimated results
Figure 3-9 shows how well the proposed algorithm estimates the PU behavior. The initial value of the estimated transition probabilities is given as $\hat{p}_0 = \hat{q}_0 = 0.5$. This figure shows that the estimated transition probabilities fluctuate around the real values. It demonstrates that the proposed algorithm well tracks the variation of the PU behavior.

The minimum required length of the sample sequence is obtained by Equation (3–32). In the proposed learning algorithm, the estimated values of $p$ and $q$ from the previous frame substitutes for the real transition probabilities in Equation (3–29) and Equation (3–31). Figure 3-10 shows that the required length of the sample sequence using the estimated transition probabilities is very close to the minimum length from theoretical analysis using real values.

Figure 3-11 shows the confidence level of the proposed learning algorithm when the accuracy requirement is $1 - \alpha = 0.95$ and $\beta = 0.1$. The confidence level is calculated as the ratio of the number of frames in which the confidence interval includes the true
Figure 3-12. The comparison of the confidence level between fixed and proposed algorithm value to the total number of frames. It shows that the estimation results of the proposed algorithm are very close to the expected value.

The proposed learning algorithm which uses adaptive length of the sample sequence is compared with the method using the fixed sequence length over frames. We only show the results for the estimation of $\rho$ and the results for $q$ have the similar trend. Figure 3-12 shows the comparison in terms of the confidence level. The confidence level is calculated as the ratio of the number of frames in which the confidence interval includes the true value to the total number of frames. According to this figure, if $1 - \alpha = 0.95$, the proposed adaptive algorithm on average requires 570 samples for sensing per frame and achieves the same confidence level as the fixed method with 771 samples. If the fixed method uses 570 samples in each frame, it only reaches the confidence level of 0.919 while the adaptive algorithm achieves 0.948. When $1 - \alpha = 0.9$, the proposed algorithm requires 366 samples per frame and achieves the same confidence level as the fixed method with 429 samples. If the fixed method
Figure 3-13. The comparison of the relative error between fixed and proposed algorithm uses 366 samples in each frame, it only reaches the confidence level of 0.866 while the proposed algorithm achieves 0.886.

Another comparison based on the relative error can be observed in Figure 3-13. The relative error is defined as \( \delta = \frac{1}{R} \sum_{r=1}^{R} \left( p_r - \hat{p}_r \right)^2 \). This figure shows the similar results as in Figure 3-12. Therefore, the proposed adaptive algorithm needs less samples per each frame to reach the same estimation accuracy in a statistical sense. It is also shown that the proposed adaptive algorithm outperforms the fixed algorithm in terms of the confidence level when the number of samples is the same.

### 3.7 Summary

This chapter investigates the estimation accuracy of the ML estimator for the PU Markov model in cognitive radio networks. An approximate normal distribution of the ML estimator is derived to analyze its confidence level. We show that the required length of the sample sequence differs greatly for the varied PU behavior. A learning algorithm which iteratively learns the Markov model is proposed with adaptive length of the sample sequence.
The proposed learning algorithm which uses adaptive length of the sample sequence is compared with the method using the fixed sequence length over frames. Numerical results show that the proposed adaptive algorithm needs less samples per each frame to reach the same estimation accuracy in a statistical sense. It is also shown that the proposed adaptive algorithm outperforms the fixed algorithm in terms of the confidence level when the number of samples is the same. Therefore, the proposed estimation algorithm achieves the requirement of the estimation accuracy while reducing unnecessary sensing slots.
CHAPTER 4
PRIMARY USER BEHAVIOR ESTIMATION WITH IMPERFECT SENSING

The estimation accuracy of PU behavior depends on the result of spectrum sensing. In the last chapter, sensing result is assumed to be perfect without sensing errors. The problem of imperfect sensing needs to be studied for real applications. A two-state Hidden Markov model (HMM) is utilized to model PU behavior and estimate PU state transition probabilities with sensing error. The transition probabilities are estimated by the Baum-Welch algorithm. The estimation accuracy is evaluated in terms of the confidence level, by utilizing the probability distribution of the estimator. Simulation results show how the estimation accuracy is affected by the length of the sample sequence, the real state transition probabilities, the selection of the initial parameter and the level of sensing error.

This chapter is organized as follows. Section 4.1 introduces an overview of PU behavior estimation with imperfect sensing and proposes HMM as the model for estimation. Section 4.2 introduces the recent studies related to imperfect sensing problems. Section 4.4 introduces the HMM-based PU behavior model. The Baum-Welch algorithm of transition probability estimation is described in Section 4.5. Section 4.6 provides the numerical results for the evaluation of the estimation accuracy and the comparison of perfect and imperfect sensing. Section 4.7 concludes this chapter.

4.1 Overview of Imperfect Sensing

In cognitive radio networks, spectrum sensing is one of the most important functionalities, as mentioned in Chapter 3. The accurate estimation of PU behavior improves the spectrum utilization and prevents the harmful interference of SUs to PUs. SUs keep sensing the channel for consecutive periods and analyze the received signals to decide whether PUs are active or not. However, there may be the undesired interference signal, such as system noise, on the same channel, which greatly affects the decision of SUs. This may result in either false alarm or miss detection of a PU
signal. As a consequence, the spectrum may be under-utilized or there may be collisions of PU and SU transmission.

Energy detection is commonly used for spectrum sensing because of its low complexity [1]. It evaluates the received energy in a given spectrum and compares it with a predefined threshold. The channel is considered "busy" if the energy value is above the given threshold and "idle" if it is below the threshold. The received energy includes the PU signal energy and the noise energy. Although it is simple to implement, it has several weaknesses. For example, the value of the noise energy is critical for determining the sensing threshold, which significantly affects the sensing performance. Inaccurate estimation of possibly changing environment noise may cause sensing errors, which leads to a degradation of estimation accuracy of PU behavior. Moreover, it does not consider the dependency of current state on the previous state.

In short, without knowing the noise level, it is difficult to obtain the correct state of PU behavior. However, there are three facts by exploring the problem of the imperfect sensing.

First, although the true states of Markov process representing PU behavior cannot be directly seen by an SU, the output generated by each state can be observed. The SU measures the received energy over consecutive time periods and stores the output samples to form an observed sequence. Then it conjectures the PU state based on the sequence.

Second, each sample of the observed sequence depends on the true state with certain false alarm and miss detection probabilities, i.e., a certain probability distribution. The observed samples are independent from each other since each observation are only determined by the true state in the same slot.

Third, the false alarm and miss detection probabilities caused by the environment noise may not change frequently, which means they usually stay the same during an observation period.
The Markov process can be illustrated in Figure 4-1, where $S_i$ represents $i$th sample of state sequence, $O_n$ represents the sample of $n$th sample of the observed sequence, $A$ represents state transition probabilities, $B$ represents the observation probability distribution.

Figure 4-1. PU behavior model with imperfect sensing

The Markov process is determined by the current state and the state transition probabilities. Only $O_n$ can be observed and it is related to the states of the Markov process by the observation probability distribution $B$. Since the observed sequence gives some information about the state sequence by PU behavior, a Hidden Markov model (HMM) could be introduced to solve the problem of imperfect sensing.

In this HMM, both the state space of the hidden variables, formed by PU behavior, and the observed sequence generated by the PU state, are discrete. The HMM has two types of parameters, transition probabilities and observation probabilities. The transition probabilities determine how the hidden state at the current slot is related to the hidden state at the previous slot. In addition, for each of the possible states, the observation probability governs the output variable at a particular time given the state of the hidden variable at that time. The nature of the output variables determines the size of this set.

In this chapter, a two-state HMM is employed to model PU behavior, where the presence and absence of PU signals are represented as busy and idle states, respectively. As in Chapter 3, the transition probabilities of the hidden states are estimated for the SU to predict the future PU behavior and based on the PU behavior
determine whether to transmit packets on the channel. The estimation is based on samples of the observed sequence instead of the on samples of PU state sequence directly. The estimation accuracy has to be enforced to get a proper knowledge of PU behavior. Estimation accuracy is evaluated in terms of the confidence level, by utilizing the probability distribution of the estimator. We evaluate the relationship among the length of the observed sequence, the real state transition probabilities, the false alarm and miss detection probabilities, the selection of initial parameters and the accuracy of the estimated values.

4.2 Related Work

The concept of HMM has been widely used to speech and image recognition [29]. A survey of HMM can be found in [30]. It recently has attracted the attention of research in cognitive radio networks. Real-time measurements were used to validate the Markov existence of PU behavior in [31] and the HMM is used to predict the true channel states. In [32, 33], the blind spectrum sensing exploits the HMM based on classification of multiple interferers. In [34, 35], an HMM-based spectrum sensing method is proposed based on quickest detection. Sequence detection algorithms are proposed for spectrum sensing for HMM-based PU behavior in [36].

The HMM parameters are estimated using the Baum-Welch algorithm [30, 37] for cognitive radio networks in [38–41]. However, the estimation accuracy is not discussed and how to decide the required number of samples is not mentioned. Moreover, the selection of initial parameters is not studied in these papers. To the best of our knowledge, our work is the first work to evaluate the estimation accuracy of Baum-Welch algorithm for HMM estimation in cognitive radio networks.

4.3 The Hidden Markov Model of PU behavior

As described in Chapter 2.1.1, the SU keeps sensing the channel for consecutive time slots in the channel sensing phase. The observed results form a sample sequence of length $N$. The transition probability of PU behavior is estimated based on the
observed sequence. The SU transmits data packets in the channel access phase in light of the estimation results.

4.3.1 Model of Imperfect Sensing

For each sensing slot, the SU has to determine whether the PU is active or not. The PU behavior and system noise are assumed to remain unchanged within the duration of a frame. Let $B$ denote the bandwidth of the frequency channel, $T$ denote the sensing slot duration. The SU takes $BT$ baseband complex signal samples on the channel. If $y(n, m)$ denotes the $m$th signal sample taken at the $n$th slot, $x(n, m)$ denotes PU signals, $w(n, m)$ denotes interference signals such as white noise, $y(n, m)$ can be represented as:

$$y(n, m) = \begin{cases} 
  x(n, m) + w(n, m), & \text{PU present at } n\text{th slot} \\
  w(n, m), & \text{PU absent at } n\text{th slot}. 
\end{cases} \tag{4-1}$$

The test statistic is average received energy of the $BT$ samples at the $n$th sensing slot, which can be calculated as:

$$e_n = \frac{1}{B} \sum_{m=1}^{BT} |y(n, i)|^2. \tag{4-2}$$

The SU distinguishes the two sensing results by comparing $e_n$ with the threshold $\lambda$,

$$O_n = \begin{cases} 
  1, & e_n \geq \lambda \\
  0, & e_n < \lambda. 
\end{cases} \tag{4-3}$$

The false alarm probability $P_f$ is the probability that $O_n = 1$ given that PU signal is absent. On the other hand, the miss detection probability $P_m$ is the probability that $O_n = 0$ given that PU signal is present.
4.3.2 Structure of Hidden Markov Model

The HMM modeling of PU behavior with imperfect sensing results is illustrated in Figure 4-2. The presence and absence of PU signals are represented as busy and idle states, respectively. The HMM model could be viewed as a discrete-time bivariate random process \( \{(S_n, O_n), n = 1, 2, \cdots, N\} \). \( \{S_n\} \) is a binary sequence with 0 and 1 representing idle and busy state, respectively. It is the hidden process which cannot be seen by SUs. It is defined as:

\[
S_n = \begin{cases} 
1, & \text{PU present at } n\text{th slot} \\
0, & \text{PU absent at } n\text{th slot}.
\end{cases}
\]  

The element of state transition probability matrix \( A \) is represented as:

\[
a_{i,j} = \text{Prob}(S_{n+1} = j|S_n = i). \tag{4–5}
\]

Specifically, \( p \) is the probability that the state at the \((n+1)\)th slot is 1 given the state at the \(n\)th slot is 0. \( q \) is the probability that the state at the \((n+1)\)th slot is 0 given the state at the \(n\)th slot is 1. They can be represented as:
Then the state transition probability matrix $A$ is given as:

$$
A = \begin{bmatrix}
1 - p & p \\
q & 1 - q
\end{bmatrix}.
$$  \quad (4-7)

$\{O_n\}$ is also a binary sequence with 0 and 1 representing the decision of PU activity. It is the observed process from which the SUs estimate the transition probabilities of PU behavior. Each $O_n$ is conditionally dependent on the corresponding $S_n$ with probability

$$
b_i(y) = \text{Prob}(O_n = y | S_n = i).
$$  \quad (4-8)

Then the false alarm probability $P_f$ and miss detection probability $P_m$ can be represented as:

$$
\begin{cases}
  P_f = b_0(1) \\
  P_m = b_1(0).
\end{cases}
$$  \quad (4-9)

Moreover, by the assumption made in Section 4.3.1, $P_f$ and $P_m$ are independent of $n$ and remain the same within a frame. The observation matrix $B$ is defined as

$$
B = \begin{bmatrix}
1 - P_f & P_f \\
P_m & 1 - P_m
\end{bmatrix}.
$$  \quad (4-10)

The HMM of imperfect sensing is determined and characterized by the parameter set $\phi = (\pi, A, B)$, where $\pi$ is the initial state probability matrix $\pi = [P_{idle}, P_{busy}]$. The two probabilities represents that the channel is idle and busy at the first sensing slot. The mathematical symbols for this model is summarized in Table 4-1.
Table 4-1. HMM Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>Length of the sample sequence</td>
</tr>
<tr>
<td>$S_n$</td>
<td>State of PU at the $n$th sensing slot</td>
</tr>
<tr>
<td>$O_n$</td>
<td>Observation of the channel at the $n$th sensing slot</td>
</tr>
<tr>
<td>$\pi_i$</td>
<td>Initial probability of state $i$</td>
</tr>
<tr>
<td>$P_{idle}$</td>
<td>Initial probability of state 0</td>
</tr>
<tr>
<td>$P_{busy}$</td>
<td>Initial probability of state 1</td>
</tr>
<tr>
<td>$\pi$</td>
<td>Initial state probability matrix</td>
</tr>
<tr>
<td>$a_{i,j}$</td>
<td>Transition probability of PU from state $i$ to state $j$</td>
</tr>
<tr>
<td>$p$</td>
<td>Transition probability of PU from state 0 to state 1</td>
</tr>
<tr>
<td>$q$</td>
<td>Transition probability of PU from state 0 to state 1</td>
</tr>
<tr>
<td>$A$</td>
<td>State transition probability matrix</td>
</tr>
<tr>
<td>$b_i(y)$</td>
<td>Observation probability from state $i$ to observation $y$</td>
</tr>
<tr>
<td>$P_f$</td>
<td>Observation probability from state 0 to observation 1</td>
</tr>
<tr>
<td>$P_m$</td>
<td>Observation probability from state 1 to observation 0</td>
</tr>
<tr>
<td>$B$</td>
<td>Observation probability matrix</td>
</tr>
<tr>
<td>$\phi$</td>
<td>HMM model parameter set $\phi = (\pi, A, B)$</td>
</tr>
</tbody>
</table>

### 4.4 Estimation of Transition Probabilities Using HMM

The parameters of the HMM are estimated using the observed sequence. Given the observed sequence $O = \{O_1, O_2, \cdots, O_N\}$, the optimal value of the estimation is a parameter set $\hat{\phi}_{opt}$ which maximizes the probability of current observed sequence.

It is denoted as:

$$\hat{\phi}_{opt} = \arg\max_{\phi} \text{Prob}(O|\phi) \quad (4-11)$$

In this work, we will first calculate the probability of the observed sequence given each possible state sequence. Since the calculation is generally infeasible for $2^N$ possible state sequences, the estimation of HMM model parameters is conducted based on a dynamic programming algorithm, which is called Baum-Welch algorithm.

#### 4.4.1 Probability of a given observed sequence

We want to find out the probability of a given observed sequence $O$. Let $S = \{S_1, S_2, \cdots, S_N\}$ be a certain state sequence. The conditional probability of $O$ based on $S$ is calculated as:
\[
\text{Prob}(O|S, \phi) = b_{S_1}(O_1) b_{S_2}(O_2) \cdots b_{S_N}(O_N).
\]  
(4–12)

The occurrence probability of the state sequence is calculated as:

\[
\text{Prob}(S|\phi) = \pi_{S_1} a_{S_1, S_2} a_{S_2, S_3} \cdots a_{S_{N-1}, S_N}.
\]  
(4–13)

The expression for the occurrence probability of the observed sequence \(O\) is obtained by summing up all possible state sequences,

\[
\text{Prob}(O|\phi) = \sum_s \text{Prob}(O, S|\phi) = \sum_s \text{Prob}(O|S, \phi) \text{Prob}(S|\phi) = \sum_s \pi_{S_1} b_{S_1}(O_1) a_{S_1, S_2} b_{S_2}(O_2) \cdots a_{S_{N-1}, S_N} b_{S_N}(O_N).
\]  
(4–14)

In this two-state HMM, the expression could be more specific based on its unique characteristics. Let \(n_{ij}\) \((i, j \in \{0, 1\})\) represents the number of state transitions from state \(i\) to state \(j\). \(n_i\) \((i \in \{0, 1\})\) denotes the number of all transitions from state \(i\).

Recall that in Section 3.3.1, the individual occurrence probability of a state sequence is dependent on a parameter set \(\chi = (S_1, S_N, n_0, n_{01})\). If we assume a \(0 \rightarrow 0\) transition when \(S_N = 0\) and \(1 \rightarrow 1\) transition when \(S_N = 1\), then \(n_0\) (the number of all transitions from state \(0\)) is the number of states \(0\) in the sequence and \(N - n_0\) is the number of states \(1\). In this way, the effect of \(S_N\) can be ignored and this probability of the state sequence is determined only by \(n_0\) and \(n_{01}\). Specifically, there is \(n_{01}\) transitions from state \(0\) to state \(1\), \(n_0 - n_{01}\) transitions from state \(0\) to state \(0\), \(n_{10}\) transitions from state \(1\) to state \(0\) and \(N - n_0 - n_{01}\) transitions from state \(1\) to state \(1\).

Given the real transition probabilities \(p\) and \(q\), \(\text{Prob}(S|\phi)\) is calculated by:

\[
\text{Prob}(S|\phi) = \pi_{S_1} p^{n_{01}} q^{n_{01}} (1-p)^{n_0-n_{01}} (1-q)^{N-n_0-n_{01}}.
\]  
(4–15)
Note that in this case, the relationship of $N, n_0, n_{01}$ must satisfy the following constraints:

$$n_{01} \leq \min(n_0, N - n_0).$$  \hspace{1cm} (4–16)

The proof is similar to the proof of Equation (3–2).

Let $m_i (i \in \{0, 1\})$ denote the number of observation $i$. Suppose out of $m_0$ observations 0, the number of 0s emitted by states 0 in the state sequence is $L$. Then there is $L$ emissions from state 0 to observation 1, $n_0 - L$ emissions from state 0 to observation 0, $m_0 - L$ emissions from state 1 to observation 0 and $N - n_0 - (m_0 - L)$ emissions from state 1 to observation 1. Apparently, $L \leq \min(n_0, m_0)$.

Then the conditional probability of $O$ based on $S$ with a given $n_0$ is calculated by:

$$Prob(O|S, \phi) = P_f^L (1 - P_f)^{n_0 - L} P_m^{m_0 - L} (1 - P_m)^{N - n_0 - (m_0 - L)}.$$  \hspace{1cm} (4–17)

Because $S$ can take one of possible $2^N$ values at each step, it becomes computationally overwhelming to enumerate each state sequence and calculate its corresponding joint probability with the given observed sequence. The Baum-Welch algorithm, which is a special case of a generalized expectation maximization (EM) algorithm, is used to iteratively estimate the parameter set [42].

4.4.2 Estimation of HMM Model Parameters Using Baum-Welch Algorithm

Baum-Welch algorithm calculates maximum likelihood estimates for the model parameters of an HMM, when the training samples are only given by observations. It makes use of both forward and backward algorithm.

**Forward algorithm.**

Define $\alpha_i (n)$ is the probability of obtaining the partial observation sequence, $O_1, O_2, \cdots, O_n$, and $S_n$ is state $i (i = 0, 1)$. It is denoted as:

$$\alpha_i (n) = Prob(O_1, O_2, \cdots, O_n, S_n = i|\phi).$$  \hspace{1cm} (4–18)
It can be calculated recursively as follows.

\[
\alpha_i(1) = \pi b_i(O_1), \quad i = 0, 1
\]

\[
\alpha_i(n) = \sum_{j=0,1} \alpha_j(n-1) a_{j,i} b_i(O_n), \quad i = \{0, 1\}, n \in [2, N]
\]

Therefore the occurrence probability of the observed sequence \(O\) is computed by:

\[
Prob(O|\phi) = \sum_{i=0,1} \alpha_i(n).
\]

(4–20)

**Backward algorithm.**

Define \(\beta_i(n)\) is the probability of obtaining the partial observation sequence, \(O_{n+1}, O_{n+2}, \cdots, O_N\), given that \(S_n\) is state \(i (i = 0, 1)\). It is denoted as:

\[
\beta_i(n) = Prob(O_{n+1}, O_{n+2}, \cdots, O_N, |S_n = i, \phi).
\]

(4–21)

It can be calculated recursively as follows.

\[
\beta_i(N) = 1, \quad i = 0, 1
\]

\[
\beta_i(n) = \sum_{j=0,1} \beta_j(n+1) a_{i,j} b_j(O_{n+1}), \quad i = \{0, 1\}, n \in [1, N-1]
\]

(4–22)

Therefore the occurrence probability of \(O\) is computed by:

\[
Prob(O|\phi) = \sum_{i=0,1} \beta_i(1).
\]

(4–23)

**Baum-Welch algorithm.**
\( \alpha_i(n) \) and \( \beta_i(n) \) are used in Baum-Welch algorithm to calculate the probability that \( S_n \) is state \( i (i = 0, 1) \), given the observed sequence is \( O \). It is obtained by:

\[
\gamma_i(n) = \text{Prob}(S_n = i | O, \phi) = \frac{\alpha_i(n) \beta_i(n)}{\sum_{i=0,1} \alpha_i(n) \beta_i(n)} \tag{4–24}
\]

The probability that \( S_n \) is state \( i (i = 0, 1) \) and \( S_{n+1} \) is state \( j (j = 0, 1) \) given the observed sequence is \( O \) is obtained by:

\[
\gamma_{i,j}(n) = \text{Prob}(S_n = i, S_{n+1} = j | O, \phi) = \frac{\alpha_i(n) a_{i,j} \beta_i(n+1) b_j(O_{n+1})}{\sum_{i=0,1} \sum_{j=0,1} \alpha_i(n) a_{i,j} \beta_i(n+1) b_j(O_{n+1})} \tag{4–25}
\]

Using these two expressions, the expected number of transitions from state \( i \) to state \( j \) can be represented as \( \sum_{n=1}^{N-1} \gamma_{i,j}(n) \). In the same way, the expected number of total transitions from state \( i \) can be represented as \( \sum_{n=1}^{N-1} \gamma_i(n) \). Moreover, the expected number of emissions from state \( i \) to observation \( y \) can be represented as \( \sum_{n \in \{1, 2, \ldots, N-1\}, O_n = y} \gamma_i(n) \).

Then the model parameter set \( \phi = (\pi, A, B) \) is estimated as follows.

1. The initial state probability matrix \( \hat{\pi} \):

\[
\hat{\pi}_0 = \hat{P}_{idle} = \gamma_0(1) \tag{4–26}
\]

\[
\hat{\pi}_1 = \hat{P}_{busy} = \gamma_1(1)
\]

2. The transition probability matrix \( \hat{A} \):
\[ \hat{a}_{0,1} = \hat{p} = \frac{\sum_{n=1}^{N-1} \gamma_{0,1}(n)}{\sum_{n=1}^{N-1} \gamma_0(n)} \]  
\[ \hat{a}_{1,0} = \hat{q} = \frac{\sum_{n=1}^{N-1} \gamma_{1,0}(n)}{\sum_{n=1}^{N-1} \gamma_1(n)} \]  

(4–27)

3. The observation probability matrix \( \hat{B} \):

\[ \hat{b}_0(1) = \hat{P}_f = \frac{\sum_{n \in \{1,2,...,N-1\}, O_n = 1} \gamma_0(n)}{\sum_{n=1}^{N-1} \gamma_0(n)} \]  
\[ \hat{b}_1(0) = \hat{P}_m = \frac{\sum_{n \in \{1,2,...,N-1\}, O_n = 0} \gamma_1(n)}{\sum_{n=1}^{N-1} \gamma_1(n)} \]  

(4–28)

Since the real model parameter set \( \phi \) is not known by the SU in advance, the above equations cannot be used directly to perform the estimation. Therefore, a training algorithm has to be used to update the model parameter. It initializes the process with a coarse estimate of \( \phi \) and then trains the model iteratively until a termination criteria is satisfied.

Given the observed sequence \( O \), the algorithm is conducted as follows.

1. Initialize the HMM model parameter set as \( \hat{\phi}_0 = (\hat{\pi}_0, \hat{A}_0, \hat{B}_0) \). Then calculate the probability of the given observed sequence \( O \) \( Prob(O | \hat{\phi}_0) \) by the forward or backward algorithm.

2. Compute \( \gamma_{i,j}(n) \) and \( \gamma_i(n) \) using \( \hat{\phi} \).
3. Update the model parameter set $\hat{\phi}_1 = (\hat{\pi}_1, \hat{A}_1, \hat{B}_1)$ by the results of Step 2. Then calculate $\text{Prob}(\mathcal{O} | \hat{\phi}_1)$.

4. If $\text{Prob}(\mathcal{O} | \hat{\phi}_1) - \text{Prob}(\mathcal{O} | \hat{\phi}_0) > \delta$, where $\delta$ is a predefined threshold, then $\hat{\phi}_0$ is apparently not the parameter set that maximizes current observation. Then it returns to Step 1 with $\hat{\phi}_1$ substituting for $\hat{\phi}_0$.

In this way, the training algorithm updates the estimation result after each iteration. The process will terminate if $\text{Prob}(\mathcal{O} | \hat{\phi}_1)$ does not increase by $\delta$ or after a pre-set maximum number of iterations. Then it yields the final estimation of the model parameter set $\hat{\phi}_{opt}$.

4.5 The Estimation Accuracy Analysis of HMM

Since a precise model and estimation of PU behavior is very important to improve utilization and avoid harmful interference to PUs, the estimation accuracy has to be enforced for the SU. As described in Chapter 3, the length of the sample sequence plays an important part on the estimation accuracy. It should be long enough to achieve certain precision of the estimation. However, the energy wasted for spectrum sensing and the memory used for storing the samples are expected to be kept as low as possible for the low-capable SUs. Therefore, our goal is selecting the length of sample sequence to achieve the estimation accuracy while reducing unnecessary sensing slot. In this chapter, we are only concerned about the estimation accuracy of transition probability matrix for imperfect sensing. Then comparing it with the estimation accuracy for perfect sensing, we can observe the impact of system noise on the estimation results. In what follows, only the estimation of $\hat{p}$ is discussed and the estimation of $\hat{q}$ can be derived similarly.

4.5.1 Confidence Level of the Estimation

As in Chapter 3, deriving the probability distribution is a straightforward way to evaluate the performance of the HMM estimator. Denote $\text{Prob}(\hat{p} = x)$ as the probability that $p$ takes on value $x$. Given the observed sequence with $N$ observations is $\mathcal{O} =$
\{O_1, O_2, \cdots, O_N\}, two parameters need to be calculated for computing \(\text{Prob}(\hat{\rho} = x)\). One is the probability of current observed sequence \(\text{Prob}(O|\phi)\). This is obtained by Equation (4–20) or Equation (4–23). The other parameter is the estimated value of \(\hat{\rho} = x\) based on the current observed sequence \(O\). However, since it uses Baul-Welch algorithm, which is conducted by iteratively training the model, \(x\) cannot be computed. Therefore, without knowing \(x\), we cannot group all the sequences that lead to the same \(x\) as in Section 3.3.1 for perfect sensing. The exact probability distribution cannot be computed in a polynomial time. The only way to calculate it is to generate the two parameters for all \(2^N\) possible sequences. Because enumerating all possible sequences becomes computationally infeasible when \(N\) gets large, we randomly generate a certain number of observed sequences based on HMM and calculate its corresponding \(\text{Prob}(O|\phi)\) and \(\hat{\rho}\). When the number of sequences is large enough, the results could approximately represent the probability distribution of HMM.

Figure 4-3 shows the approximate probability distribution of HMM for imperfect sensing compared with the probability distribution of ML estimator for perfect sensing. The false alarm probability \(P_f\) and the miss detection probability \(P_m\) are both set to 0.1. The number of observed sequence is 1000. From this figure, we can see that when \(N\) is increased, the PMF concentrates towards the real transition probability value, which indicates a more accurate estimation. We can also see that the probability distribution for perfect sensing is more concentrated than that of HMM for imperfect sensing, which verifies the impact of sensing errors on the estimation accuracy.

A more specific indication of the estimation accuracy is represented by the confidence level. It can be obtained using the probability distribution. The same as in Chapter 3, the accuracy requirement of the estimator is defined as the probability that the true value of the transition probability \(p\) is in the confidence interval \([\hat{\rho} - \beta \hat{\rho}, \hat{\rho} + \beta \hat{\rho}]\) is at least \(1 - \alpha\), where \(1 - \alpha\) is the confidence level and \(\beta\) is the error bound. It is
Perfect sensing
Imperfect sensing

Figure 4-3. The probability distribution of perfect and imperfect sensing represented as:

\[ \Pr(\hat{p} - \beta \leq p \leq \hat{p} + \beta) \geq 1 - \alpha, \ 0 < \alpha, \beta < 1. \]  \tag{4–29} \]

It can be rewritten as:

\[ \Pr\left(p - \frac{\beta}{1 + \beta} \leq \hat{p} \leq p - \frac{\beta}{1 - \beta} \right) \geq 1 - \alpha. \]  \tag{4–30} \]

Given the HMM model parameters, the probability that \( \hat{p} \) falls into the interval \([p - \frac{\beta}{1 + \beta} p, p - \frac{\beta}{1 - \beta} p]\) is determined only by the length of the sample sequence. This can be calculated by summing over all the probabilities within the interval, using the approximate probability distribution. Therefore, the minimum sample sequence length should be selected which satisfies the condition in Equation 4–30. The relationship among the length of the sample sequence, the PU behavior specified by the transition
probabilities and the corresponding estimation accuracy will be evaluated using simulation in Section 4.6.

4.5.2 Selection of Initial Parameters

Recall that in Baum-Welch algorithm, the first step is to initialize the model parameters. Since the Baum-Welch algorithm is a locally iterative method [43], the final estimated values of HMM and the number of iterations depend significantly on the initial model. The Baum-Welch algorithm is guaranteed to converge to one $\hat{\phi}$ of HMM that locally maximizes the probability of the current observations. It is well known that if the initial parameters are not chosen properly, it may converge to a local optimum and lead to inaccurate estimates, which may be significantly worse than the global optimum [44, 45].

Although HMMs have been employed in many applications, a frequent problem is selecting or estimating initial values for the parameters of the model [29, 46]. The initialization of the main parameters of HMMs is studied for many applications, such as the application of clustering algorithms [44, 47–52], Gaussian Mixture Models [53–57] and the use of random values [29, 44, 45, 58, 59]. However, there is not a common consensus concerning the use of any criterion to select the technique to be used. The characteristics of the scenarios have to be utilized for the initialization.

For the spectrum sensing, there are two features which can be utilized for the selection of initial transition probabilities.

1. There are only two states for PU behavior. If we assume a perfect sensing, based on the state sequence, the transition probabilities can be estimated by the ML estimator using Equation 3–1.

2. For an imperfect sensing, the false alarm and miss detection probabilities are usually kept as low as possible through a number of technique [60–65] because of the requirement of the cognitive radio networks. It means the observed sequence can
still be considered as an approximation of the hidden state sequence for a coarse evaluation.

Base on these two features, the initial guess of transition probabilities can be estimated by assuming the observed sequence \( O = \{ O_1, O_2, \cdots, O_N \} \) represents the state sequence \( S = \{ S_1, S_2, \cdots, S_N \} \).

Let \( m_{xy} (x, y \in \{0, 1\}) \) represents the number of transitions from observation \( x \) to observation \( y \). \( m_x (x \in \{0, 1\}) \) denotes the number of all transitions from observation \( x \), the initial guess of the state transition probabilities are given by:

\[
\begin{align*}
\hat{p} &= \frac{m_{01}}{m_0} \\
\hat{q} &= \frac{m_{10}}{m_1}.
\end{align*}
\]

(4–31)

The above is a coarse estimation of the state transition probabilities, which can be used as an initial guess of the Baum-Welch algorithm. The complexity of the estimator is low, which is applicable for low-capable SUs. Then the Baum-Welch algorithm will converge to local optimum around this value, which is usually a good estimation of the parameters.

The effects of an improper initialization on the estimation accuracy will be illustrated through simulations, which can validate the significance of the initial parameter selection. We will also show the estimation accuracy using our approach to select the initial guess of state transition probabilities.

4.6 Numerical Results

The performance of the Baum-Welch algorithm for HMM is evaluated through numerical results. In the simulation, the accuracy requirement in terms of the confidence level is given as \( 1 - \alpha = 0.9 \) and \( \beta = 0.1 \). 1000 observed sequences are generated using HMM as the sample set. Each sequence generates an estimate of the transition probability. The metric representing the estimation accuracy is the confidence level,
Figure 4-4. The comparison of confidence level between perfect and imperfect sensing which is calculated as the ratio of the number of sequences where the condition in Equation (4–30) is satisfied to the total number of sequences. In the following figures, only the estimation results of $p$ are evaluated.

Figure 4-4 shows the confidence level of $\hat{p}$ for both perfect sensing and imperfect sensing using HMM for estimation with regard to difference length of sequences, when $p = q = 0.5$. For imperfect sensing, sensing error is specified by $P_r = P_m = 0.1$. The sample sequence length $N$ changes from 300 to 1000. This figure shows that the estimation accuracy in terms of confidence level increases when the length of the sample sequence increases. The estimation accuracy of imperfect sensing is lower than that of perfect sensing because of the sensing errors.

The sample sequence length has to be carefully selected to achieve the estimation accuracy requirement while reducing unnecessary sensing slots. Figure 4-5 is the 3D graph showing the minimum length of sample sequence required to achieve the requirement of estimation accuracy with regard to different $p$ and $q$ combinations. The
The required length of sample sequence
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000

Figure 4-5. The impact of transition probabilities on the required sequence length for imperfect sensing.

accuracy requirement is specified by $1 - \alpha = 0.9$ and $\beta = 0.1$. We can see that the required length of sample sequence increases as the transition probabilities decreases. It also shows that the required length is symmetric for $p$ and $q$ because we need to satisfy the confidence level for both of them. When $p$ is larger than $q$, the bottle neck goes to the confidence level of $q$. Symmetrically, while the bottle neck is the confidence level of $p$ if $q$ is larger than $p$. Compared with the required sample sequence length for perfect sensing in Figure 4-6, the impact of imperfect sensing can be seen by the increased length of the sample sequence required for the same $p$ and $q$ combination.

In order to understand the impact of imperfect sensing in a more straightforward way, Figure 4-7 shows the required sample sequence length with $p = q$ for perfect sensing and imperfect sensing. In this simulation, $p$ is set to the same value as $q$ which changes from 0.1 to 0.9. From this figure, we can see that the imperfect sensing requires longer sequence to achieve the same confidence level as that of perfect sensing. But they show similar trend that when the real transition probability increases, the required length decreases.
Figure 4-6. The impact of transition probabilities on the required sequence length for perfect sensing

Figure 4-7. The impact of transition probabilities on the required sequence length (p=q)
The impact of an improper initialization on the estimation accuracy is illustrated in Figure 4-8, which shows the significance of the initial parameter selection. The simulation is conducted when $p = q = 0.5$. The curve represents the confidence level of the Baum-Welch algorithm using a fixed value from 0.1 to 0.9 as the initial parameter for all 1000 sequences. The value of the horizontal line with squares represents the confidence level of the estimation using a different random value in $(0, 1)$ for each observed sequence. The value of the horizontal line with stars is the confidence level of the estimation with $\hat{p}_0 = \frac{m_{01}}{m_0}$ for each sequence. It is shown that the estimation accuracy using the value which is very close to the real $p$ as the initial value is better than using any other initial values. However, it is difficult to guess the real $p$ at the first step. The simulation shows that the estimation using our approach to select the initial state transition probabilities performs better than using most of the other fixed values. Our approach is also much better than that uses random selected values for each sequence. The simulation results highlight the significance of properly selecting the initial model parameters prior to the Baum-Welch algorithm.

Figure 4-9 presents the impact of the false alarm probability on the confidence level of $\hat{p}$ for different length of sample sequence. It is shown that the confidence level decreases as the false alarm probability increases, which validates the impact of level of sensing errors on the estimation accuracy. It also provides another insight of the impact of sample sequence length on the estimation accuracy. The confidence level increases as the sample sequence length increases.

4.7 Summary

In this chapter, a two-state HMM is employed to model PU behavior with imperfect sensing. The estimation of transition probabilities is based on samples of the observed sequence instead of the on samples of PU state sequence directly. Estimation accuracy is evaluated in terms of the confidence level, by utilizing the probability distribution of the estimator. Simulation results validate the relationship among the length of the observed
Figure 4-8. The impact of initialization on the estimation accuracy

Figure 4-9. Impact of false alarm probability on the estimation accuracy
sequence, the real state transition probabilities, the false alarm and miss detection probabilities, the selection of initial parameters and the accuracy of the estimated values.
CHAPTER 5
RESIDUAL ENERGY AWARE CHANNEL ASSIGNMENT SCHEMES

In this chapter, we investigate the channel assignment problem in a cluster-based multi-channel CRSN, as described in Chapter 2, Section 2.1.2. Due to the inherent power and resource constraints of sensor networks, energy efficiency is the primary concern for network design. An $R$-coefficient is developed to estimate the predicted residual energy using sensor information (current residual energy and expected energy consumption) and channel conditions (primary user behavior). We examine three channel assignment approaches: Random pairing, Greedy channel search and Optimization-based channel assignment. The last two exploit $R$-coefficient to obtain a residual energy aware channel assignment solution. Simulation results show that $R$-coefficient-based approaches lead to better performance in terms of energy consumption and residual energy balance. Optimization-based channel assignment outperforms the other two approaches with respect to network lifetime.

This chapter is organized as follows. Section 5.1 introduces the challenges and our motivation. We review the related work and compare it with our study in Section 5.2. In Section 5.3, we propose an $R$-coefficient as the metric for channel assignment. Section 5.4 describes three different channel assignment approaches. Simulation results are provided in 5.5 and summaries are made in Section 5.6.

5.1 Spectrum Sharing in Cognitive Radio Sensor Networks (CRSNs)

In CRSNs, a sensor selects the most appropriate channel once an available band is identified and vacates the band when a PU's transmission is detected. Since multiple sensors may try to access the same spectrum, a spectrum sharing mechanism is needed to coordinate multiple simultaneous transmissions, which includes both the management of coexistence with PUs and resource allocation among sensors.

There is a large amount of work in the literature on dynamic spectrum sharing and channel assignment problem [1, 66], such as the multi-carrier modulation technology
and the use of common control channel for spectrum related information exchange. Most of the studies concentrate on sensing channel availability to improve spectrum utilization, modeling PU activity to avoid collision or analyzing QoS performance such as delay and throughput. However, only a few of the current studies for channel assignment in cognitive radio networks consider energy consumption problem, which is the critical concern for energy-constrained WSNs. Reducing energy consumption helps non-rechargeable sensors to operate for a longer time. When a certain number of sensors die, the network will be considered nonfunctional. The time duration from sensor deployment to the instant of network nonfunction is called network lifetime [67]. Sensors should consume the energy at the same pace in order to maximize network lifetime. Therefore, energy consumption and residual energy balance are both critical in WSN design. There are some related studies about energy efficiency in ordinary WSNs. In [15], a realistic power consumption model for WSN devices is proposed to derive the conditions for minimum power consumption in data transmissions. In [67], residual energy information and channel state information are considered for lifetime-maximizing MAC protocols. For CRSNs, the primary concern is how to reduce energy consumption and prolong the network lifetime with an appropriate dynamic spectrum allocation scheme.

In this chapter, we consider a cluster-based multi-channel CRSN. The channel assignment problem is investigated from the aspect of energy consumption and network lifetime. An \textit{R-coefficient} determined by sensor energy information and PU behavior is proposed to represent the predicted residual energy. We try to balance residual energy for each sensor with channel assignment based on \textit{R-coefficient}. Three channel assignment approaches are provided: Random pairing, Greedy channel search and Optimization-based channel assignment.
5.2 Related Work

There are many existing works studying channel assignment problems in cognitive radio networks. In [68], OSA-MAC protocol based on IEEE 802.11 DCF model is proposed for opportunistic spectrum access. It provides both uniformly random channel selection and spectrum opportunity-based channel selection. However, it does not consider the state change of PU behavior, which is studied in our work. In [20], the authors evaluate the data link layer QoS performance of cognitive users, such as average throughput and packet loss rate. It models PU behavior as a two-state Markov Chain. However, it assumes that if the channel is not used by PUs at the beginning of a time frame, it remains unoccupied during the transmission of cognitive users. In our study, a PU may take over the channel at any time even before a cognitive transmission finishes. In this case, the cognitive user will stop the transmission immediately when a PU transmission is detected. Our goal is to assign the cognitive user to the channel with the least probability of interruption.

As far as we have seen, there are few papers focusing on the energy consumption and network lifetime for channel assignment in cognitive radio networks. In [69], an optimization model for energy-efficient spectrum access is formulated to minimize the energy per bit for each single user. However, this network model considers and ignores PU behavior. Also, the channel selection decision is made individually without considering collisions to other cognitive users and the energy consumption based on the whole network. In [8], the channel selection problem based on PU behavior and energy consumption is formulated as a multi-armed bandit problem. PU behavior is also modeled as a two-state Markov Chain. But still, the cognitive user makes decisions only based on its own observation and it does not consider other cognitive users. In our work, we consider the energy consumption and residual energy balance with respect to the network performance and our final goal is to prolong the network lifetime. To the best
of our knowledge, we are the first to consider network-wide residual energy balance for channel assignment problems in cognitive radio networks.

5.3 R-Coefficient

In this chapter, we use the network model and time frame structure described in Chapter 2, Section 2.1.2. As mentioned previously, when a sensor has data to send, it wakes up and transmits an assign request message. When CH receives the assign request messages from all the active sensors, CH begins to process the channel assignment based on the sensor information and PU's state on each available channel. Our primary concern in this process is energy consumption in data transmission. Based on our energy consumption model, we know that transmission energy for the sensor is dependent on the length of transmitted data, which in our design is the number of the transmission slots.

For data packet transmission, we refer to the frame structure in Figure 2-3, where a complete data transmission takes up \( L \) slots. Sensors access idle channels according to the channel assignment decision from the CH. If a channel is assigned to a CM that needs to transmit, the CM will start transmission in the coming data slot. Suppose that CM \( i \) attempts to transmit a packet with the length of \( L \) slots on channel \( j \). If CM \( i \) can successfully transmit the whole packet, the energy consumption is \( E_{tr}^j(L) \), as in Equation (2–2). According to the Markov model, the probability that a sensor can successfully transmit a packet on channel \( j \) is equal to the probability that the channel remains in the idle state for \( L \) consecutive time slots as follows:

\[
P_j^{success} = P(\text{idle for next } L \text{ slots} | \text{idle in the initial slot})
= (1 - p_j)^L.
\]

(5–1)

If the CM only transmits for \( l < L \) slots, i.e., the CM detects PU transmission, then the energy consumption is \( E_{tr}^i(l) \). \( P_j^l \), which represents the probability that a CM only transmits for \( l \) slots on channel \( j \) due to the collision with PU is
\[ P_j^l = (1 - p_j)^{l-1}p_j \quad 1 \leq l \leq L. \quad (5-2) \]

Based on the above equations, the expected energy consumption for CM \( i \) transmitting on channel \( j \) is derived as follows:

\[
\bar{E}_{ij}^{data} = \sum_{l=1}^{L} E_{tr}^i(l) P_j^l + E_{tr}^i(L) P_j^{success}. \quad (5-3)
\]

Our goal is to reduce the total energy consumption while at the same time balance residual energy of each sensor in order to maximize the network lifetime. According to this objective, we propose an \( R \)-coefficient to represent the predicted Residual energy. If sensor \( i \) transmits on channel \( j \) with current residual energy \( R_i^c \), the predicted residual energy is denoted as

\[
R_{ij} = R_i^c - \bar{E}_{ij}^{data} \quad (5-4)
\]

The value of \( R_{ij} \) describes the statistically expected residual energy of sensor \( i \) to transmit on channel \( j \). It is an estimated value for prediction of the possible residual energy and depends on the current residual energy of sensor \( i \), the distance from sensor \( i \) to CH, and PU statistics on channel \( j \). We try to find \((i, j)\) pair with \( R_{ij} \) as large as possible when carrying out channel assignment.

From the definition of \( R_{ij} \), we can see that potentially, sensor \( i \) will select the channel with minimum expected energy consumption. Therefore the actual energy consumption will be reduced with a greater probability and the residual energy after transmission for each active sensor will be raised. Also, channel \( j \) will be assigned to the sensor with larger residual energy. Or equivalently, a sensor with larger residual energy is more likely to undertake the task of transmission. In this way, the residual energy for each active sensor can be kept about the same level. In other words, the residual energy will be balanced across all sensors in the cluster using this \( R \)-coefficient.
Fairness among sensors is not guaranteed in this work. However, it can be improved with simple modifications. For example, in the case that a sensor has a packet to transmit but could not get through for a certain number of time frames, it will inform CH and be assigned with higher priority. This problem will be studied more in our future work.

5.4 Channel Assignment

Next we will describe three channel assignment approaches. We briefly introduce random pairing, which is a simple and commonly used method. Then based on the R-coefficient, we propose Greedy channel search and optimization-based channel assignment.

5.4.1 Random Pairing

Assume that there are $N$ active sensors and $M$ available channels in the current frame. The CH picks up sensor $i$ randomly with probability $\frac{1}{N}$, selects channel $j$ with probability $\frac{1}{M}$ and then makes $i$ and $j$ as a pair. In this way, channel $j$ is assigned to sensor $i$. Sensor $i$ is marked assigned sensor and channel $j$ is marked assigned channel. Then $N$ is updated as the number of remaining active sensors and $M$ is updated as the number of remaining available channels. The same procedure continues until either all the active sensors get assigned or there are no more available channels. This random pairing is simple to implement. However, it does not consider any information either from the sensor energy or from the PU behavior.

5.4.2 Greedy Channel Search

When the energy consumption and the residual energy are considered for CRSN, it would be more efficient to exploit our R-coefficient by allowing sensor $i$ to access channel $j$ with the largest $R$ value. In this approach, we come up with two-step Greedy allocation. In the first step, for each active sensor, we find the largest R-coefficient value over all the available channels. In the second step, we search over these largest values, the maximum R-coefficient is obtained. Then the corresponding channel $j$ will
be assigned to the corresponding sensor \( i \). Sensor \( i \) and channel \( j \) are marked assigned respectively. It continues for the remaining sensors and channels until every active sensor is assigned or every available channel is used. Using this two-step Greedy allocation during each iteration for unassigned sensors and unassigned channels, we can always find an \( (i, j) \) pair with the maximum value of remaining \( R\)-coefficient.

The pseudo code of this approach is illustrated in Figure 5-1.

```plaintext
Set \([R]_{N\times M}=R\) coefficient matrix;
for node \( i=1 \) to \( N \) do
  if \( i \) is active and not assigned then
    for channel \( j=1 \) to \( M \) do
      if \( j \) is available and not assigned then
        if \( r_{ij}=\text{Max of } [R] \) then
          Assign \( j \) to \( i \);
          Mark \( i \) and \( j \) as assigned;
          update \([R]\) with \( r_{ij}=0\);
        end if
      end if
    end for
  end if
end for
```

Figure 5-1. Pseudo code of Greedy algorithm

### 5.4.3 Optimization-based Channel Assignment

With respect to the network performance for the cluster, the network-wide residual energy is critical to prolong the network lifetime. Greedy channel search always searches for the maximum value of \( R\)-coefficient in each single iteration. It cannot guarantee that the total residual energy of all sensors in the network is maximized after this frame. If we maximize the sum of \( R\)-coefficient for each \( (i, j) \) pair when carrying out channel assignment, the total actual residual energy of the whole cluster after data transmission will be kept as much as possible.
Suppose that there are \( N \) active sensors and \( M \) available channels. The decision variable \( x_{ij} \) represents the assignment of sensor \( i \) to channel \( j \), taking value 1 if assigned and 0 otherwise. \( \sum_{i=1}^{N} \sum_{j=1}^{M} R_{ij} x_{ij} \) is our objective function.

Maximize \( \sum_{i=1}^{N} \sum_{j=1}^{M} R_{ij} x_{ij} \)

subject to

Case 1: \( N > M \)
\[ \sum_{j=1}^{M} x_{ij} \leq 1, \quad \sum_{i=1}^{N} x_{ij} = 1 \]

Case 2: \( N < M \)
\[ \sum_{j=1}^{M} x_{ij} = 1, \quad \sum_{i=1}^{N} x_{ij} \leq 1 \]

Case 3: \( N = M \)
\[ \sum_{j=1}^{M} x_{ij} = 1, \quad \sum_{i=1}^{N} x_{ij} = 1 \] (5–5)

The three constraints are related to three different cases, respectively. If \( N > M \), those \( M \) available channels are assigned to \( M \) selected sensors with a one-to-one matching. \( N - M \) remaining sensors will be not be able to transmit in this frame. If \( N < M \), the \( N \) active sensors get assigned to \( N \) selected sensors. \( M - N \) remaining channels will not be used in this frame. If \( N = M \), the number of active sensors is equal to the number of available channels. This is an ideal case with the best utilization, in which we can assign exactly one channel to one sensor.

Now we will discuss the case when \( N = M \). The assignment problem is actually a one-to-one matching problem in a weighted bipartite graph. We have a complete bipartite graph \( G = (S, T; E) \) with \( N \) sensor vertices \((S)\) and \( N \) channel vertices \((T)\). And each edge has a nonnegative weight \( R_{ij} \). We want to find a perfect matching with maximum weight. The Hungarian algorithm [70] is one of the famous algorithms that solve this assignment problem in polynomial time.
This optimization model can be made rather more flexible to different cases. If \( N > M \), then \( N - M \) dummy channels are invented, perhaps called "sitting still doing nothing", with \( R\)-coefficient of 0 for the sensors assigned to them. In this way, the constraints in Case 1 become two equations. Then the assignment problem can be solved in the same way as in Case 3 and still give the best solution to the problem. Similar tricks can be played when \( N < M \) with \( M - N \) dummy sensors.

The pseudo code of this Optimization-based channel assignment is shown in Figure 5-2.

```plaintext
n=number of active sensors;
m=number of available channels;
f=objective function;
if n > m then
    Add n-m dummy channels;
    Constraints=Constraints of Case 1;
else
    if n < m then
        Add m-n dummy channels;
        Constraints=Constraints of Case 2;
    end if
else
    Constraints=Constraints of Case 3;
end if
Call linear programming function;
```

Figure 5-2. Pseudo code of Optimization-based channel assignment

5.5 Simulation Results

In this section, we apply the three channel assignment approaches (Random, Greedy and Optimization) in the simulation and compare the numerical performance with respect to energy consumption and residual energy balance. The result for the number of alive sensors over frames, which represents network lifetime, is also provided.

The simulation is conducted within of a cluster of CRSN. PU behavior on each channel is represented as a two-state Markov model (idle/busy) parameterized with transition probabilities as described in Chapter 2, Section 2.2. Sensor activity
is governed by an independent probability $P_{\text{Active}}$. All three channel assignment approaches are applied in the simulation on the same set of sensor activity and PU behavior. TABLE 5-1 shows all the parameters used in the simulation.

Table 5-1. Simulation parameters of residual energy aware channel assignment

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of sensors ($N$)</td>
<td>25/30/35/40/45/50/55</td>
</tr>
<tr>
<td>Number of channels ($M$)</td>
<td>10</td>
</tr>
<tr>
<td>Sensor active probability ($P_{\text{active}}$)</td>
<td>0.5</td>
</tr>
<tr>
<td>Channel transition probability ($P_{\text{idle} \rightarrow \text{busy}}$)</td>
<td>0.15-0.35</td>
</tr>
<tr>
<td>Channel transition probability ($P_{\text{busy} \rightarrow \text{idle}}$)</td>
<td>0.65-0.85</td>
</tr>
<tr>
<td>Number of slots in each frame ($K + 1$)</td>
<td>10</td>
</tr>
<tr>
<td>Number of frames ($F$)</td>
<td>50/250</td>
</tr>
<tr>
<td>Slot period ($t_s$)</td>
<td>0.001 second</td>
</tr>
<tr>
<td>Sensor data transmission rate ($R$)</td>
<td>1Mbits/second</td>
</tr>
<tr>
<td>Sensor packet size ($L$)</td>
<td>6 slots</td>
</tr>
<tr>
<td>Sensor initial energy ($E_{\text{in}}$)</td>
<td>1000 mJ</td>
</tr>
<tr>
<td>Energy consumption for channel assignment ($E_c$)</td>
<td>1 mJ</td>
</tr>
<tr>
<td>RF circuit energy consumption ($E_{\text{cir}}$)</td>
<td>5 mJ/slot</td>
</tr>
<tr>
<td>Amplifier energy required at CH ($\varepsilon$)</td>
<td>0.001 mJ/slot/m$^2$</td>
</tr>
<tr>
<td>Distance between sensor and CH ($d$)</td>
<td>Uniformly distributed between 50m-100m</td>
</tr>
</tbody>
</table>

Figure 5-3 shows average network energy consumption over frames for three approaches in networks of different numbers of sensors. It is observed that there is a big gap between Random pairing and other two approaches, which indicates that a great amount of energy can be saved through Greedy channel search and Optimization-based channel assignment. Recall that when R-coefficient is considered, a sensor is more likely to access the channel with low energy consumption demand. Therefore, total energy consumption of the network is much lower for the two R-coefficient-based channel assignment approaches. Besides, Optimization-based channel assignment performs slightly better than Greedy channel search, which verifies the meaning of the optimal solution to the optimization model.

Figure 5-4 shows the comparison of three approaches based on average standard deviation of sensor residual energy over frames related to the number of sensors, which
indicates residual energy balance of the sensors. The difference among curves is similar to Figure 5-3. The objective of using $R$-coefficient is to suppress low-energy sensors but encourage high-energy sensors to access the channels, which results in a balanced residual energy distribution. This leads to the big gap between Random pairing and other two $R$-coefficient-based approaches.

Figure 5-5 shows the number of alive sensors after each frame, which represents network lifetime. Using the $R$-coefficient as the metric in channel assignment helps balance the residual energies of the sensors, which leads to a better chance to extend network lifetime. The Optimization-based channel assignment performs better than Greedy channel search with regard to network lifetime. This is because Greedy approach tries to maximize the expected residual energy for a single sensor-channel pair while Optimization method reaches the optimal solution of maximizing the total predicted residual energy for the entire network.
Figure 5-4. Average standard deviation of sensor residual energy over frames (number of frames = 50)

Figure 5-5. Number of remaining alive sensors after each frame (number of sensors = 30)
Figure 5-6 shows the comparison between OSA-MAC [68] and our optimization method in terms of effective energy consumptions defined as the total energy consumption of the network divided by the total transmission slots. For OSA-MAC, we simulate their spectrum opportunity-based channel selection, in which the selection approach takes the probabilities of spectrum availability according to PU behavior. PU behavior is assumed not to change during the packet transmission of cognitive users and the collision among cognitive users is reduced by backoff time. In our work, the Markov model of PU behavior makes it more realistic and the one-to-one matching between sensors and channels addresses the collision problem. Hence, our approach can achieve better energy efficiency.

![Graph showing comparison between Optimization and OSA-MAC](image)

**Figure 5-6.** Average effective energy consumption over frames (number of frames = 50)

### 5.6 Summary

In this chapter, we study the channel assignment problem in a cluster-based multi-channel CRSN with consideration of energy consumption, residual energy balancing and network lifetime. An *R-coefficient* is proposed according to the predicted
residual energy. Based on the *R-coefficient*, *Greedy channel search* is proposed to consecutively maximize the expected residual energy for each single sensor-channel pair, and an optimization model is proposed to maximize the total expected residual energy in order to improve the network-wide performance. The simulation results show evident improvement coming from the *R-coefficient* based channel assignment on both energy consumption and residual energy balance. The *Optimization-based channel assignment* outperforms *Greedy channel search* in network lifetime. Besides, the comparison with OSA-MAC shows that the optimization assignment has better performance in terms of effective energy consumption.
CHAPTER 6
DYNAMIC SPECTRUM ACCESS WITH PACKET SIZE ADAPTATION AND RESIDUAL ENERGY BALANCING

In this chapter, the effective energy is considered, which is the energy consumption for the successfully transmitted data. We demonstrate an improvement in the energy efficiency and the network lifetime by packet size adaptation and residual energy balancing. This is achieved by investigating the dynamic spectrum access issues with two methods. The first employs a packet size adaptation: a variable size to adapt the time-varying channel states, to improve the energy efficiency in the network. The second method exploits an awareness of the residual energy in the channel assignment. This helps to balance the residual energy to prolong the network lifetime, compared to the random pairing approach.

This chapter is organized as follows. Section 6.1 provides the introduction of the problem. The related work is reviewed in Section 6.2. Section 6.3 introduces the packet size adaptation scheme, which improves energy utilization. Section 6.4 and the residual energy balancing channel assignment. Section 6.5 provides simulation results and analysis. Section 6.6 summarizes this chapter.

6.1 Dynamic Spectrum Access and Energy Consumption

In the last chapter, we focus on reducing energy consumption and prolonging network lifetime. This is achieved by maximizing total residual energy of the whole network. Since the ongoing transmission of a sensor stops when a PU signal is detected, the packet is dropped and the energy is wasted. Even if the total residual energy is maximized, the amount of packets which can be successfully transmitted may be very small. Moreover, transmitting the assign_request message and receiving the assign_request message also consumes energy. In this chapter, we consider energy efficiency, another important concern for proposing an appropriate opportunistic spectrum access scheme. We minimize the total energy consumption needed to successfully transmit a certain amount of information bits. It helps sensors to transmit
as much information as possible during their lifetime. We are also concerned about the network lifetime, which is introduced in Chapter 5. The average wasted energy, which is defined as the total unused energy in the network when it dies, should be reduced to prolong the network lifetime [67]. Thus, the residual energy of all the sensors in the network need to be kept around the same level with a balancing scheme so that the wasted energy is minimized when the first sensor dies. In this way, the network lifetime is maximized.

In this chapter, we investigate the intra-cluster communication of a cluster-based CRSN. With respect to the time-varying channel conditions depending on the PU behavior, the packet size is adjusted to improve energy efficiency of the network. After the packet size is determined, the channel assignment of CRSN is investigated with the objective of prolonging the network lifetime. Residual energy is balanced for each sensor with channel assignment based on sensor energy information and PU behavior.

6.2 Related Work

There were many existing works studying dynamic spectrum access problems in cognitive radio networks [1, 16–18, 21, 28, 34, 66, 71–73]. However, all these papers focused on QoS performance, such as throughput and packet loss rate, without the consideration of the energy efficiency and network lifetime for spectrum access in cognitive radio networks. In [71], effective capacity of cognitive radio channels was studied under QoS constraints and channel uncertainty, in which the transmitter is unaware of the channel fading coefficients. In [21], a frame-based opportunistic spectrum scheduling scheme was implemented to maximize the aggregate throughput of all secondary users.

There were also many studies addressing the energy constraints in WSNs. In [74], a link adaptation mechanism with an adaptive frame size was proposed at the MAC layer to improve energy efficiency. In [75], an optimized-MAC protocol with high energy efficiency was proposed by adjusting the sensor duty cycle based on the network traffic.
However, these studies were based on ordinary WSNs and they did not need to address the dynamic spectrum access in a cognitive radio network. In [7], an optimization model for energy-efficient spectrum access in CRSNs was formulated to minimize the energy per bit for each single user. Each SU made the decision of channel individually and a power control mechanism is used to reduce the interference among SUs. However, it did not need to consider the impacts of PU behavior.

6.3 Packet Size Adaptation

As shown in the last chapter, the probability of successfully transmitting a packet can be calculated by:

\[
P_{\text{success}}^j = P(\text{idle for next L slots | idle in the initial slot}) = (1 - p_j)^L. \tag{6–1}
\]

We observe that as the packet size increases, the probability that the packet can successfully get transmitted is decreased. The energy is wasted if the packet collides with PU packets. On the other hand, if the packet size reduces, the ratio of energy consumption in the access control slot of Figure 2-3 to the energy consumption in the data transmission slots increases, which also reduces energy efficiency. Taking into account this tradeoff, there may exist an optimal packet size leading to the best energy efficiency. In this work, the metric energy-per-bit (EPB) is employed to denote the ratio between the total energy consumption and the amount of data successfully transmitted. As introduced in Chapter 1, protocol design for WSNs is effective on a network basis rather than on an individual basis, so the objective is to minimize the EPB for the whole network.

Since both the PU behavior and sensor activity are time-varying in nature, the adaptation of the packet size is considered for every frame. As mentioned in Chapter 2, Section 2.1.2, when a CM has data to send, it wakes up and transmits an access_request message. When the CH receives the access_request messages from
all the active CMs, it begins to determine the packet size for the data transmission of the current frame based on the PU behavior and sensor activity.

The total energy consumption in the network is composed of two parts: (1) the energy consumed in the access control slot, which includes transmission energy for access_request packets and receiving energy for the broadcast access_reply packet and (2) the energy consumed in the data transmission slots. In this work, the star topology is applied with an equal distance $d$ between each CM and the CH. Let the size of access_request packet and access_reply packet be $K_1$ and $K_2$ bits, respectively. The energy consumed in the access control slot is calculated by:

$$E_{\text{control}} = (E_{\text{cir}} + \varepsilon d^2) \times K_1 + E_{\text{cir}} \times K_2.$$  

(6–2)

For data transmission, the expected energy consumption for CM $i$ transmitting on channel $j$ is calculated by Equation (6–3),

$$E_{\text{data}}^i = \sum_{l=1}^{L} E_{i,l}^{tr}(l) \times P_l^j + E_{i,L}^{tr}(L) \times P_{success}^j.$$  

(6–3)

Since the distance between each CM and the CH is the same, the value of $E_{\text{data}}^i$ is not related to the index $i$ and it can be written as $E_{\text{data}}^j$.

According to Equation (6–1), the probability of a successful transmission is only dependent on the PU behavior of each channel. Therefore, the expected amount of the successfully transmitted data bits on channel $j$ is:

$$K_{data}^j = P_{success}^j \times B \times T \times L.$$  

(6–4)

If the number of active CMs is more than the number of available channels, all the available channels could be used for data transmission. In this case, we write the predicted network EPB as the following equation:
\[
EPB = \frac{E_{control} \times N_{active} + \sum_{j=1}^{M} E_{data}^j}{\sum_{j=1}^{M} R_{data}^j},
\] (6–5)

where \(N_{active}\) is the number of active CMs.

If the number of active CMs is less than the number of available channels, the channels with less \(p_j\) are selected in order to reduce the probability of collisions with PUs during the data transmission. This is because with less \(p_j\), these channels are more likely to stay in state idle if they are sensed idle in the initial slot. The number of the selected available channels is \(N_{active}\).

If the minimum packet size is \(L_{\text{min}}\) and the maximum packet size is \(L_{\text{max}}\), which depends on specific network parameter constraints on packet size, the optimal packet size in terms of number of slots is obtained by

\[
L_{\text{opt}} = \arg\min_{L \in [L_{\text{min}}, L_{\text{max}}]} EPB
\] (6–6)

Since active CMs and available channels are time-varying, the packet size should be adaptively changed to minimize the network EPB for the current frame. The CH keeps tracking the changes of channel conditions by estimating state transition probabilities and use this information together with the access request messages from CMs to make decisions on the packet size at the beginning of each frame.

### 6.4 Residual Energy Balancing Channel Assignment

After the CH decides the optimal packet size, it begins to conduct the channel assignment based on the information about both the channel conditions and the active CMs’ residual energy. The primary goal in this process is to balance the residual energy among all the CMs in the network in order to maximize the network lifetime. We also use the \(R\)-coefficient introduced in Chapter 5, Section 5.3 to represent the predicted Residual energy. In this chapter, since the energy consumed for control information is
considered, the expression of the $R$-coefficient is different. If an active CM is assigned to channel $j$, the total expected energy consumption in the current frame is:

$$
\bar{E}_j = E_{\text{control}} + \bar{E}_{j}^{\text{data}}, \quad (6-7)
$$

Where $E_{\text{control}}$ and $\bar{E}_{j}^{\text{data}}$ is calculated by Equation (6–2) and Equation (6–3), respectively. With current residual energy $R_i^c$, the predicted residual energy is obtained by:

$$
R_{ij} = R_i^c - \bar{E}_j. \quad (6–8)
$$

It is an estimated value for prediction of the possible residual energy and depends on both the current residual energy of CM $i$ and PU statistics on channel $j$. In this chapter, the basic idea for assigning channel $j$ to CM $i$ is to find $(i, j)$ pairs with the values of $R_{ij}$ close to each other. In this way, the residual energy of each active CM could be maintained at the same level after the data transmission in the current frame. Therefore, the objective is to find the minimum of the variance for the $R$-coefficient of each CM.

Suppose that there are several active CMs and available channels. According to Section 6.3, if the number of active CMs is less than the number of available channels, the channels with less $p_j$ are selected for the assignment. These channels become candidate channels. Also, if the number of active CMs is greater than the number of available channels, the sensors with larger residual energy are selected as candidate CMs. Therefore, the number of candidate channels is equal to the number of candidate CMs. This number is denoted as $N$. The decision variable $x_{ij}$ represents the assignment of CM $i$ to channel $j$, taking value 1 if assigned and 0 otherwise.

The average value of the $R$-coefficients for a given channel assignment is defined as
\[
\bar{R} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} R_{ij} x_{ij}}{N}.
\]

(6–9)

The variance minimization model is developed as follows:

\[
\text{Minimize } \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} (R_{ij} - \bar{R})^2 x_{ij}
\]

subject to \[
\sum_{j=1}^{N} x_{ij} = 1, \quad \sum_{i=1}^{N} x_{ij} = 1.
\]

(6–10)

Note that all the candidate CMs and candidate channels will be assigned. The total current residual energy and the total expected energy consumption remain the same for any assignment, which makes the total expected residual energy (\(R\)-coefficients) a constant. Therefore, \(\bar{R}\) is also a constant. This means that the optimization model is an integer linear program (ILP). The constraints indicate that each candidate CM can only be assigned to one channel and each candidate channel can only be used by one CM. The constraint matrix \(A\) consists of \(n^2\) columns and \(2n\) rows.

\[
A = \begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1 \\
1 & 1 & \cdots & 1
\end{bmatrix}
\]

(6–11)

\(1\) is a \(n\)-row vector of all ones and \(I\) is an \(n \times n\) identity matrix. The determinant of every square submatrix formed from \(A\) has value -1, 0, or +1. It means that \(A\) is totally unimodular \([?]\). Applying this property to the optimization model, an optimal basic
feasible solution with $x_{ij} = 0$ or 1 replaced by $x_{ij} \geq 0$ will be all integers. Therefore, the solution to the linear program (LP) is actually the solution to the ILP.

The simplex algorithm is a popular algorithm for LP. However, it is not very efficient and sometimes may have exponential time complexity. Based on the characteristics of this problem, an efficient algorithm can be used to minimize the variance. The basic idea is to make the CMs with larger residual energy select the channels with larger expected energy consumption. The current residual energy of all candidate CMs is denoted by $R_i = (R_1, R_2, \ldots, R_N)$. The expected energy consumption for a CM transmitting on channel $j$ is $E_j = (E_1, E_2, \ldots, E_N)$. Both $R_i$ and $E_j$ are sorted in a descending order. Then CM $i$ with the largest $R_i$ is assigned to channel $j$ with the largest $E_j$, and mark CM $i$ and channel $j$ as assigned. The same procedure continues for the remaining CMs and channels until every candidate CM is assigned and every candidate channel is used. The residual energy of CMs after data transmission is expected to have the minimum variance.

The following is to prove that the proposed algorithm gives the optimal solution.

The pair $(R_i, E_j)$ denotes that CM $i$ is assigned to channel $j$. For a given assignment, if there exist two pairs: $(R_1, E_2)$ and $(R_2, E_1)$ with $R_1 > R_2$ and $E_1 > E_2$, the variance is decreased by switching the two pairs to $(R_1, E_1)$ and $(R_2, E_2)$.

**Proof:** Denote the variance for the assignment with $(R_1, E_1)$ $(R_2, E_2)$ and the assignment with $(R_1, E_2)$ $(R_2, E_1)$ as $Var_1$ and $Var_2$, respectively. Since the assignment of other CMs and channels is not changed, the difference of the variance is determined by the assignment of CM 1 and CM 2. Assume $Var_1 > Var_2$, according to Eq.(??), we have

$$
(R_1 - E_1 - \bar{R})^2 + (R_2 - E_2 - \bar{R})^2 > (R_1 - E_2 - \bar{R})^2 + (R_2 - E_1 - \bar{R})^2.
$$

(6–12)

From this equation, we get

$$
(R_1 - R_2)(E_1 - E_2) < 0.
$$

(6–13)
This contradicts with the fact that $R_1 > R_2$ and $E_1 > E_2$. Therefore $\text{Var}_1 > \text{Var}_2$, which indicates the variance is decreased by the switch.

Based on the above analysis, for any given assignment, we can always take a certain number of switches to reach the result of the proposed algorithm. As each switch increases the variance, the proposed algorithm produces the minimum variance. The switching procedure is as follows: Suppose there are $N$ candidate CMs with $R_1 > R_2 > \ldots > R_N$ and $N$ candidate channels with $E_1 > E_2 > \ldots > E_N$. For the proposed algorithm, the channels assigned to $R_1, R_2, \ldots, R_N$ should be $E_1, E_2, \ldots, E_N$, respectively. For any other assignments, if $E_1$ is not assigned to $R_1$, we can first switch two pairs $(R_1, E_j)$ and $(R_i, E_1)$ to $(R_1, E_1)$ and $(R_i, E_j)$. As described above, this switch reduces the variance. After $E_1$ is assigned to $R_1$, another switch is taken so that $E_2$ can be assigned to $R_2$, which also reduces the variance. The same process is done for the remaining CMs and channels, after which the assignment is the same as the result of the proposed algorithm and has the minimal variance. Therefore, the proposed algorithm provides the optimal solution to the optimization model.

In this way, the residual energy of CMs after data transmission is expected to have the minimum variance. The total unused energy when the first CM dies is reduced and hence the network lifetime is prolonged, as described in Chapter 5, Section 5.1.

### 6.5 Simulation Results

We conduct the simulation within a cluster of CRSN. PU behavior on each channel is represented as (idle/busy) state, which is described in Chapter 2, Section 2.2. The probability that a CM has a packet to transmit is $P_{\text{Active}}$. TABLE 6-1 shows all the parameters used in the simulation.

#### 6.5.1 Performance of Packet Size Adaptation

The proposed packet size adaptation scheme, as described in Section 6.3, is examined by comparing with the fixed packet size. Figure 6-1 plots the accumulative network energy efficiency across the whole simulation duration for the packet size
Table 6-1. Simulation parameters of dynamic spectrum access with packet size adaption

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of sensors ($N$)</td>
<td>30</td>
</tr>
<tr>
<td>Number of channels ($M$)</td>
<td>20</td>
</tr>
<tr>
<td>Sensor active probability ($P_{active}$)</td>
<td>0.3</td>
</tr>
<tr>
<td>Channel transition probability ($P_{idle\rightarrow busy}$)</td>
<td>(0,1)</td>
</tr>
<tr>
<td>Channel transition probability ($P_{busy\rightarrow idle}$)</td>
<td>(0,1)</td>
</tr>
<tr>
<td>Slot duration ($t_s$)</td>
<td>0.002 second</td>
</tr>
<tr>
<td>Data transmission rate ($B$)</td>
<td>40 kbps</td>
</tr>
<tr>
<td>CM control packet size ($L$)</td>
<td>4 bytes</td>
</tr>
<tr>
<td>CH broadcast packet size ($L$)</td>
<td>25 bytes</td>
</tr>
<tr>
<td>CM data packet size ($L$)</td>
<td>[25,128] bytes</td>
</tr>
<tr>
<td>Sensor initial energy ($E_{in}$)</td>
<td>1 J</td>
</tr>
<tr>
<td>RF circuit energy consumption ($E_{cir}$)</td>
<td>50 nJ/bit</td>
</tr>
<tr>
<td>Amplifier energy required at CH ($\epsilon$)</td>
<td>100 PJ/bit/m$^2$</td>
</tr>
<tr>
<td>Distance between sensor and CH ($d$)</td>
<td>25 m on average</td>
</tr>
</tbody>
</table>

![Figure 6-1. The comparison of accumulative network EPB among different packet-sizing schemes](image-url)
Figure 6-2. The comparison of overall network EPB among different packet-sizing schemes

Figure 6-3. The comparison of the volume of successfully delivered information among different packet-sizing schemes
adaptation scheme and the fixed packet-sizing schemes. The network energy efficiency is characterized in terms of EPB. We observe that the proposed packet size adaptation scheme achieves the best energy efficiency in successful information delivery by keeping its EPB at the lowest level among all packet-sizing schemes. More direct comparison among the packet-sizing schemes can be observed in Figure 6-2 and Figure 6-3, which provide more insights on the impacts of packet sizes on the overall network EPB and the overall volume of successfully delivered information bits. Using adaptive packet size, the EPB is 0.431 mJ/bit and the number of transmitted information is 66.95 Mb. For the fixed packet sizing scheme, the EPB value first decreases as the packet size increases, and then it keeps increasing. The reverse trend exits in terms of the transmitted data. Note that the optimal packet size of the fixed scheme is 50 bytes, for which the EPB is 0.437 mJ/bit and the number of transmitted information is 66.87 Mb. It is still worse than the performance of adaptive packet size. Since the packet size adaptation scheme dynamically tracks the channel behavior and calculates the new packet size accordingly, it is able to achieve a better energy efficiency than any fixed frame-sizing schemes.

6.5.2 Residual Energy Balancing Channel Assignment

The other result of interest is the network residual energy balance. We compare the balance-aware channel assignment discussed in Section 6.4 with the random pairing scheme in which the match between the sensors and the channels is in a purely random fashion. We also compare the proposed algorithm with the Greedy algorithm which always searches for the pair with the maximum $R_{ij}$. The results shown in Figure 6-4 verifies the design goal of residual energy balancing channel assignment in that the death of sensors is distributed in a narrow time period compared with the other channel assignment schemes. This is desirable because the total unused energy is reduced when the network dies.
Figure 6-4. The comparison of network lifetime among different channel assignment schemes

Figure 6-5. The impact of estimation accuracy on accumulative network throughput
Table 6-2. The impact of estimation accuracy on other metrics

<table>
<thead>
<tr>
<th>Number of samples</th>
<th>EPB (mJ/bit)</th>
<th>Transmitted data (Mb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>0.433</td>
<td>12.25</td>
</tr>
<tr>
<td>20</td>
<td>0.469</td>
<td>9.90</td>
</tr>
<tr>
<td>40</td>
<td>0.448</td>
<td>11.25</td>
</tr>
<tr>
<td>60</td>
<td>0.441</td>
<td>11.71</td>
</tr>
<tr>
<td>80</td>
<td>0.439</td>
<td>11.89</td>
</tr>
<tr>
<td>100</td>
<td>0.436</td>
<td>11.95</td>
</tr>
</tbody>
</table>

6.5.3 Impacts of estimation accuracy

Following the discussion of transition probability estimation in the two-state Markov model in Chapter 3, we are interested in evaluating the impact of estimation accuracy on the network energy efficiency. Fig. 6-5 plots the accumulative network throughput across the whole simulation duration via applying different numbers of samples \((n)\). The accumulative network throughput is defined by the total number of successfully delivered bits divided by the elapsed simulation time. The accumulative throughput curve led by applying the real transition probability serves as an upper bound for the purpose of comparison. As we can observe, the network throughput increases as the number of sample increases, which shows the relationship between estimation accuracy of the PU behavior and the network energy performance. This relation is illustrated in more details in Table 6-2 where the overall network EPB and successfully delivered information bits are shown.

6.6 Summary

We have demonstrated the use of the packet size adaptation scheme and the residual energy balancing channel assignment approach in a cluster-based multi-channel CRSN with the concern of energy constraints. The packet size adaptation is employed to exploit the current information from both PU behavior and sensor activity to minimize energy-per-bit, thereby improving the energy efficiency during the lifetime of sensors. The residual energy balancing channel assignment relies on the R-coefficient to balance the residual energy among the networks, which helps to prolong the network
lifetime. Additionally, we have evaluated the impact of PU behavior estimation on the network energy efficiency. It is demonstrated that the performance with the estimated values gets closer to the performance with the prior known transition probabilities as the number of samples increases.
CHAPTER 7
CONCLUSIONS

Cognitive radio technology improves spectrum utilization by allowing secondary users (SUs) to access the licensed spectrum bands in an opportunistic manner as long as it does not interfere with the activity of the primary users (PUs). CRSN has been proposed as a promising application in recent studies where a cognitive radio is installed in each sensor and it can be tuned to any available channel. In this dissertation, the spectrum sensing and spectrum access problems are investigated for the CRSN. Due to the inherent power and resource constraints of sensor networks, energy efficiency is the primary concern for the network design.

For spectrum sensing, ML estimation is used for the Markov model of PU behavior. The PMF of the ML estimator is derived to evaluate the accuracy of the estimated transition probabilities. Its approximate normal distribution is also derived to analyze its confidence level. We show that the required length of the sample sequence differs greatly for the varied PU behavior. A learning algorithm which iteratively learns the Markov model is proposed with adaptive length of the sample sequence. Numerical results show that the proposed estimation algorithm achieves the requirement of the estimation accuracy while reducing unnecessary sensing slots. For the imperfect sensing with sensing errors, a two-state HMM is employed to model PU behavior with imperfect sensing. Baum-Welch algorithm is used to estimate the transition probabilities. The estimation accuracy is compared with that of perfect sensing.

We investigate the dynamic spectrum access problem in a cluster-based multi-channel CRSN. The total residual energy based on the $R$-coefficient is maximized to reduce energy consumption and prolong the network lifetime.

Another important concern is the total energy consumption needed to successfully transmit a certain amount of information bits. We employ the packet size adaptation to
exploit the current information from both PU behavior and sensor activity to minimize EPB, thereby improving the energy efficiency during the lifetime of sensors.

In order to further extend this work, the following problems can be studied:

1. In the primary user behavior estimation work, more termination conditions and initial probability estimation methods can be explored to further improve its accuracy. Another interesting direction can be the enhancement of imperfection sensing: since the Baum-Welch algorithm is locally optimized, the initial value is very important, so more studies can be done to improve the accuracy of its estimation. It will also be great if we can find a globally optimized algorithm to avoid this problem.

2. We estimate the transition probabilities of PU behavior without knowing the false alarm and miss detection probabilities. If the false alarm and miss detection probabilities are known in prior, a different model may be needed for the estimation. The estimation accuracy can be analyzed to compare with the current HMM model.

2. In the dynamic spectrum access work, we only discuss intra-cluster channel assignment, so this can be extended to include inter cluster performance. Also, the throughput impact can be studied using more sophisticated mathematics models.
REFERENCES


BIOGRAPHICAL SKETCH

Xiaoyuan Li received her bachelor’s degree in Communications Engineering from the School of Communication and Information Engineering, in University of Electronic Science and Technology of China, Chengdu, Sichuan, China, in 2007. She received her Master’s degree and PhD degree in Department of Electrical and Computer Engineering, in University of Florida, Gainesville, Florida, USA, in 2009 and 2013, respectively.