CLASSIFICATION STYLE REGRESSION FOR SPECTRAL OPENING PMF ESTIMATION

Garrett Fosdick, Michael Marefat, Tamal Bose
University of Arizona, Electrical and Computer Engineering Dept.
Tucson, AZ, 85721
[gfosdick,marefat,tbose]@email.arizona.edu

ABSTRACT

Dynamic spectrum allocation (DSA) permits unlicensed users to access spectrum owned by a licensed user given they do so without interference to the primary user. To avoid interference with other users, the unlicensed user needs to be aware of channel availability. Spectrum sensing allows a radio to find spectrum holes, but costs energy and time. Predictive methods can be used to decrease the amount of spectrum sensing needed to find an available channel. We designed a novel neural network architecture for spectrum hole prediction. This neural network is capable of creating probability mass functions (PMF) estimates of the length of channel openings with no assumptions of the initial probability distribution or prior knowledge about the traffic. This architecture is shown to work through a mathematical proof, and its performance is measured through simulation.

INTRODUCTION

As the number of wireless devices increases, unlicensed bands have become crowded while licensed bands frequently are underutilized[1][2]. The FCC measured spectrum usage in 2002 in Atlanta, Chicago, New Orleans, San Diego, and Washington DC suburbs and found that many licensed bands were not in use or only in use part of the time. This is because allocation of licensed spectrum is often based on peak usage and not typical usage. By allocating spectrum based on the peak usage, bands with high variability can be left mostly empty the majority of the time[3]. By making use of the spectral holes in these licensed channels, the overcrowding on unlicensed bands can be alleviated. When unlicensed users make use of spectral holes it is refereed to as opportunistic spectrum access.

Opportunistic spectrum access requires the secondary user to scan the spectrum for spectral holes to exploit. Spectrum sensing can find these spectral holes, but requires time and energy to perform. Minimizing the amount of spectrum sensing required is ideal. Predicting likely frequency locations for spectral holes can considerably reduce the amount of spectrum sensing and channel hopping required. This has lead to the creation of cognitive engines (CE) with the goal of forecasting likely frequencies for channel openings. Forecasting spectral openings is not a trivial task. However, the probability of a spectral opening is effected by frequency, time, and space.

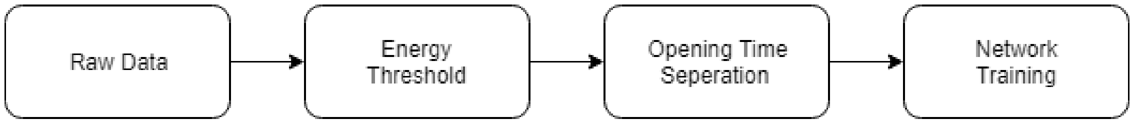

Figure 1: Example of data flow using real world data.

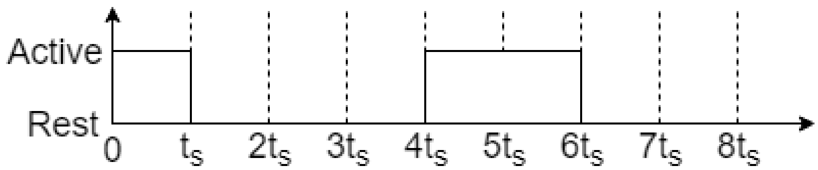

Figure 2: Example of the output of energy thresholding.

The contribution of our work is the creation of a neural network that estimates the PMF of channel opening length as well as the duty cycle of the channel. This allows for confidence based planning around the length of time a secondary user will have access to transmit. Our method is an improvement on previous methods that mainly only try to predict the duty cycle of a channel.[4] The remainder of this paper is organized as follows. The first section will discuss the design and data flow of our system. The second section will discuss the proof showing that this system will approach the actual PMF to minimize loss. The third section will discuss the setup and results of simulation before ending with the conclusions of our paper.

APPROACH

A. Date Processing

Figure 1 shows the flow of the data using data recorded by a receiver. It is assumed the data is recorded through frequency sweeping and that the sweeping time remains consistent. This provides a value t_s which is the maximum time resolution available for the recorded data. The maximum resolution of our recorded data will effect several design steps later.

The first step of pre-processing the recorded data is to identify when the channel is active. There are several methods to identify if a channel is active, but the most popular method is energy thresholding due to the ease of implementation and lack of required prior knowledge. This method works by picking an energy threshold based on the noise floor. If the energy is above the threshold the channel is considered active, but if it is below the threshold the channel is considered at rest. In most papers the threshold is set to about 6 dB above the noise floor[5][6]. It is important to note this method does not work to detect all signal types. Some signals with low energy require waveform based sensing for accurate classification.

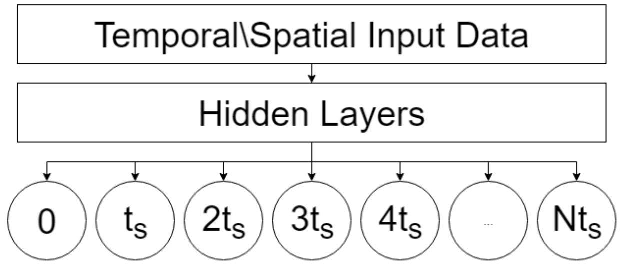

Figure 3: The basic structure of a PMF estimating neural network.

Next, the data is segmented into training data. A training sample is generated for each time step. It records the length of the opening as well as input data such as time and spatial location. For example the first six opening times for Figure 2 would be [0, $3t_s$, $2t_s$, t_s, 0, 0]. Once this data is processed it is used with back propagation to train the neural network.

B. Network Design

As will be shown in the next section through our proof, if the output of a neural network is an interval range of numbers, it can be trained to accurately estimate the PMF of an temporal/spatial input. The training process is treated like a classification problem in that each output interval is treated as a possible class.

The number of neurons in the output layer must be finite in length so an upper bound (b_u) and lower bound (b_l) must be selected. The lower bound is easy to select for our application since opening length can not be negative b_l = 0. The upper bound requires some data analysis to select. The upper bound should be selected so there is a near zero probability that an opening length will be longer than the upper bound. An idea of what upper bound will work can be found by finding the max opening length in the training data. Some channels can lay dormant long enough that this method will require an excessive output layer size. In this scenario, the upper bounds should be selected instead by estimating the max time that will be needed for planning transmissions. In this method the final node should include all values above itself.

The loss of our network is calculated using mean squared error which is shown in Equation 1. The variable y represents the set of estimated PMF values and l represents a set of values for the one hot encoded label. This loss function will be used in the next section proving the functionality of the neural network.

$$L = \sum_{n=1}^{N} (y_n - l_n)^2 \tag{1}$$

PROOF: MINIMUM LOSS AT PMF

Like all neural network, our network will be attempting to minimize its average loss. A neural network will almost never reach the minimum for loss, but it will attempt to approach it. Our proof will show that the optimal minimum our neural network is approaching is the actual PMF.

We will start this proof by establishing the significance of several of our variables. First, Y_s is the set of actual PMF values so $Y_s = \{Y_1, Y_2, ..., Y_N\}$ while y_s is the set of predicted PMF values so $y_s = \{y_1, y_2, ..., y_N\}$. Since Y_s and y_s are probabilities we can also establish the following constraint.

$$1 = \sum_{y \in y_s} y \tag{2}$$

We can also establish the average loss using Equation 1. This average is found by accounting for the likelihood of all loss possibility. Since the labels are one hot encoded there are N possible labels where N is the number of output nodes.

$$L_{avg} = \sum_{n=1}^{N} Y_n((1 - y_n)^2 + \sum_{y \neq y_n} y^2) \tag{3}$$

We are trying to find the minimum average loss for our network. We can solve for this minimum using a constrained optimization problem.

$$\begin{bmatrix} \frac{\partial L_{avg}}{\partial y_1} \\ \frac{\partial L_{avg}}{\partial y_2} \\ \dots \\ \frac{\partial L_{avg}}{\partial y_N} \end{bmatrix} = \lambda \begin{bmatrix} \frac{\partial \sum_{y \in y_s} y}{\partial y_1} \\ \frac{\partial \sum_{y \in y_s} y}{\partial y_2} \\ \dots \\ \frac{\partial \sum_{y \in y_s} y}{\partial y_N} \end{bmatrix} \tag{4}$$

Evaluating both sides of Equation 4 we get Equation 5.

$$\begin{bmatrix} \sum_{n=2}^{N} Y(2 * y_1 + 2\sum_{y \in (y_s \neq y_n)} y) \\ \sum_{n=1, n\neq 2}^{N} Y(2 * y_2 + 2\sum_{y \in (y_s \neq y_n)} y) \\ \dots \\ \sum_{n=1, n\neq N}^{N} Y(2 * y_3 + 2\sum_{y \in (y_s \neq y_n)} y) \end{bmatrix} = \lambda \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} \tag{5}$$

Equation 5 can now be solved for lambda to get Equation 6.

$$\lambda = 2 - 2\sum_{n=1}^{N} -Y_n y_n \tag{6}$$

We now plug Lambda back into equation 5 getting.

$$\begin{bmatrix} \sum_{n=2}^{N} Y(2 * y_1 + 2\sum_{y \in (y_s \neq y_n)} y) \\ \sum_{n=1, n\neq 2}^{N} Y(2 * y_2 + 2\sum_{y \in (y_s \neq y_n)} y) \\ \dots \\ \sum_{n=1, n\neq N}^{N} Y(2 * y_3 + 2\sum_{y \in (y_s \neq y_n)} y) \end{bmatrix} = \begin{bmatrix} 2 - 2\sum_{n=1}^{N} -Y_n y_n \\ 2 - 2\sum_{n=1}^{N} -Y_n y_n \\ \dots \\ 2 - 2\sum_{n=1}^{N} -Y_n y_n \end{bmatrix} \tag{7}$$

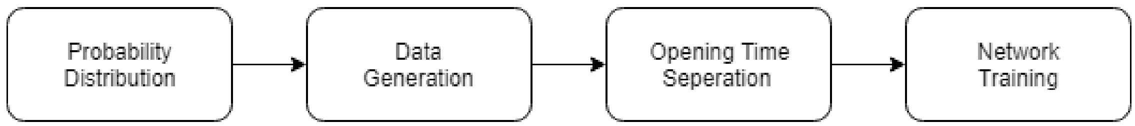

Figure 4: Example of data flow using generated data.

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \dots \\ Y_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix} \tag{8}$$

Solving for Equation 8 we get Equation 9. Equation 9 is the final equation we needed for our proof. It shows that if the average loss is optimally minimized that the predicted PMF will be equal to the actual PMF.

SIMULATION SETUP AND RESULTS

Evaluation of the PMF estimations will be done through total Error and average total Error. Total error is a calculation of the sum of errors of the estimated PMF compared to the actual PMF. This is displayed in Equation 9. The mean total error is the average total error over a 24 hour period.

$$E_{Total} = \sum_{n=1}^{N} |Y_n - y_n| \tag{9}$$

C. Generated Data Design

As seen in Figure 4, data generation replaces raw data gathering and energy thresholding. There are three steps used to generate our data. The first step is to create an artificial PMF that changes with time. The second step is to generate several thresholds using the aritifical PMF. The thresholds are placed so that a uniform random variable is as likely to end up between two thresholds as the corresponding PMF value is likely. Finally a uniform random variable is generated for each opening that needs to be generated and is then converted to the corresponding opening length using the thresholds from step 2. This generated data model is what is used for our simulations. While it has the disadvantage of being too clean lacking changes that could be expected from the real world, it has the advantage of having a known PMF to compare our results to.

The artificial PMF is constructed from two components. The first component is modeled as a discrete Rayleigh distribution. This component accounts for the likelihood of opening lengths and is represented by $f_R(sigma)$ where $sigma$ is the standard deviation. The second component is a PMF for a constantly active channel. These two components are combined to create the opening generator PMF. The probability of rest is determined by $alpha$ based on equation 10. $alpha$ can be any value between 0 and 1.

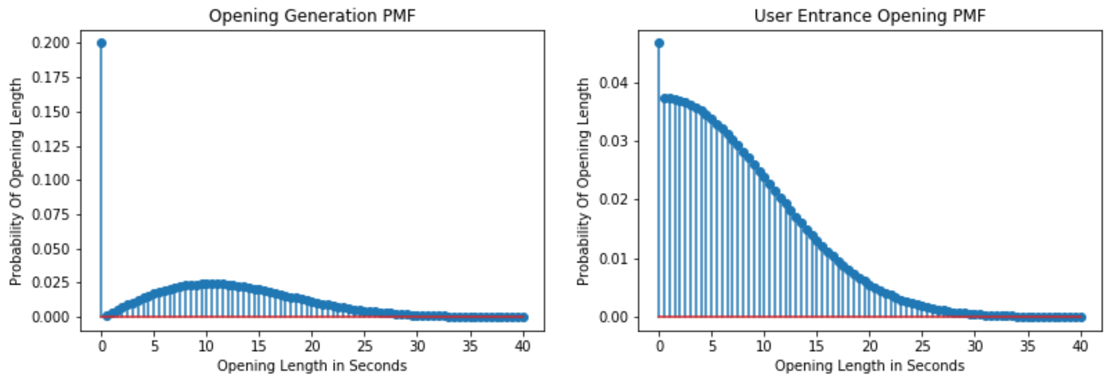

Figure 5: Left is a Generation PMF with Right being its corresponding Entrance PMF.

$$G(\sigma, \alpha, n) = (1 - \alpha) f_R(\sigma) + \alpha * \delta(n) \tag{10}$$

The data generation PMF is not the same as the PMF our system will predict. We want to predict the opening after entering the channel. It is just as likely to enter at the beginning of a spectral opening as it is to enter at any other point in the opening. This implies every generation event implies itself and all shorter opening events with equal likelihood for entrance length. This PMF is called the entrance PMF. An example of this conversion can be seen in figure 5.

D. Network Architecture

The neural network used has two different versions. The versions are mostly identical. The only difference is the input layer. Network 2 is used for the ”Varying Day” simulation and Network 1 is used for all other simulations. This is because Network 2 also needs to account for the day of the week. Network 1 only needs to account for the time of day. Development and training of the network was done using Keras. Batch size for training was 32.

Layer	Network 1	Network 2
Input	1-Dense	8-Dense
Hidden Layer 1	100-Dense ReLU	100-Dense ReLU
Hidden Layer 2	200-Dense ReLU	200-Dense ReLU
Hidden Layer 3	200-Dense ReLU	200-Dense ReLU
Output	81-Dense Softmax	81-Dense Softmax

Table 1: Network architecture for the two neural networks trained. They are mostly identical aside from the input layer.

E. Varying Data PMF

This data generation method changes the PMF as a function of time. The data generation environment has its peak channel availability at midnight and its minimum channel availability at noon.

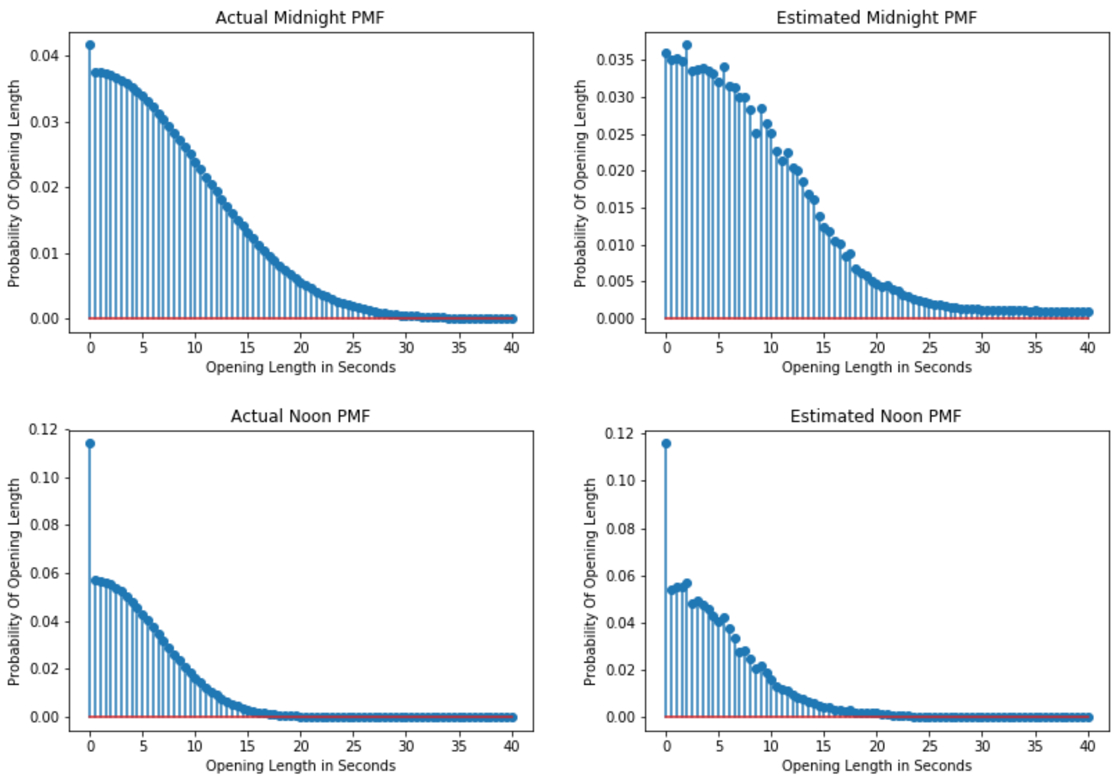

Figure 6: PMF estimation for a varying PMF with one days worth of data for training. The network was trained for 5 epochs with 81 intervals. The final total error for midnight was 0.078 and final loss of 0.0001. The final total error for noon was 0.076 and final loss of 0.0001.

The attempt of this simulation is to simulate the effects of the sleep/wake cycle of human users. The variable p_d represents the percentage of the day that has passed and starts at zero and ends at one. In this simulation $\alpha = 0.3 - 0.2cos(p_d \times 2\pi)$ and $\sigma = 8 + 2cos(p_d \times 2\pi)$. A days worth of data was generated. The results, as to be expected from a more complex model, were slightly worse, but with only an increase of 3% to the total error. The total average error was 8%. The midnight and noon PMF's can be seen in figure 6.

F. Missing Data PMF

This data set is a modification of the Varying PMF data set. The hour from 6:00AM till 7:00AM were removed from the Varying PMF data set before training. This is to evaluate how the system handles a situation where it can't directly memorize the data. In this simulation $\alpha = 0.3 - 0.2cos(p_d \times 2\pi)$ and $\sigma = 8 + 2cos(p_d \times 2\pi)$. 23 hours worth of data was generated. The final results had a total error of 11%. This rise in error may be the result of some memorization but is still low enough to be usable. The actual and predicted PMF can be viewed in figure 7.

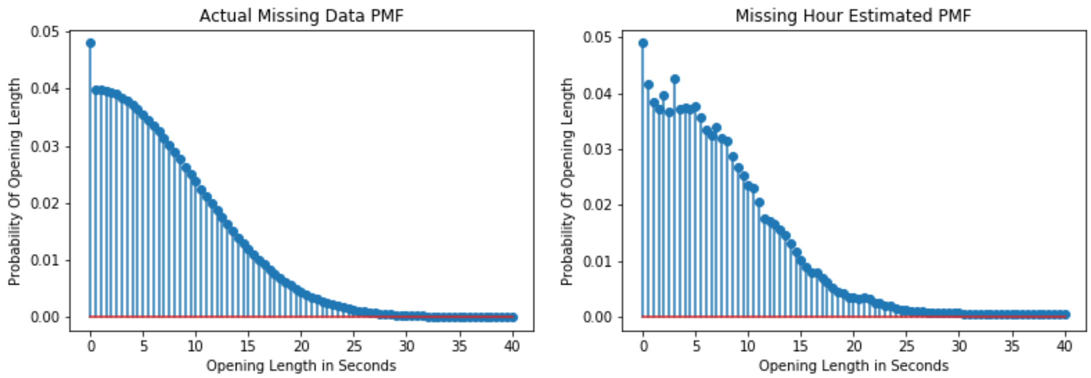

Figure 7: PMF estimation for a varying PMF with one days worth of data for training. The time from 6:00AM till 7:00AM is excluded. The network was trained for 5 epochs with 81 intervals. The final total error for 6:30AM was 0.116 and final loss of 0.0003.

G. Varying Day Data PMF

This data generation method is an expansion of the Varying PMF data generation method. The α and σ values are different on every day. The input vector for the neural network is also altered for this method to include one hot encoding for the current day. This is to evaluate the system's ability to handle more complex inputs. The α and σ values are listed in table 1. 7 days worth of data was generated.

Day	α	σ	Average Total Error
1	0.3-0.08cos($p_d \times 2\pi$)	8+sin($p_d \times 2\pi$)	0.05
2	0.3-0.15cos($p_d \times 2\pi$)	8+2sin($p_d \times 2\pi$)	0.06
3	0.3-0.2cos($p_d \times 2\pi$)	8+3sin($p_d \times 2\pi$)	0.08
4	0.3-0.3cos($p_d \times 2\pi$)	8+4sin($p_d \times 2\pi$)	0.13
5	0.3-0.15cos($p_d \times 2\pi$)	8+2sin($p_d \times 2\pi$)	0.06
6	0.3-0.04cos($p_d \times 2\pi$)	8+0.5sin($p_d \times 2\pi$)	0.05
7	0.3-0.04cos($p_d \times 2\pi$)	8+0.5sin($p_d \times 2\pi$)	0.04

Table 2: Values for each day for the Varying Day PMF.

Day 4 has the highest average total error at 13%. This is likely because Day 4 diverts the most from the rest of the data set. Day 6 and 7 both end up with the lowest average total error, most likely due to being identical days. The average total error for each day is listed in table 1.

Conclusion and Further Work

In this paper, we show and have presented an approach to accurately estimate the probability of spectral opening lengths. We were able to evaluate the performance of our neural network to achieve this task in several different environments. The predictions were accurate with variations to time and missing data.

Day	Average Total Error
1	0.05
2	0.06
3	0.08
4	0.13
5	0.06
6	0.05
7	0.04

Table 3: The average total error for each day in the varying daily scenario.

This approach has two major areas of future work. The first is testing the system against real world data. The data for this paper was simulated data which made the underlying PMF accessible, but also made the data easier to predict. Testing the system against real world data would give a more realistic evaluation of its performance. The second area of work is that there are properties of the uderlying PMF that may allow for creating more accurate PMF estimates. For example, each interval of the actual PMF value is decreasing, but the estimated PMF does not have this restriction as of yet.

REFERENCES

[1] K. Patil, K. E. Skouby, and R. Prasad, "Spectrum measurement and analysis of tv band support of cognitive radio operation in india," *Wireless VITAE 2013*, October 2013.

[2] S. Pagadarai and A. M. Wyglinski, "A quantitative assessment of wireless spectrum measurements for dynamic spectrum access," *2009 4th International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, August 2016.

[3] S. P. T. Force, "Spectrum policy task force," November 2002.

[4] K. E. Baddour, A. Ghasemi, and H. Rutagemwa, "Spectrum occupancy prediction for land mobile radio bands using a recommender system," *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, August 2019.

[5] R. I. C. Chiang, G. B. Rowe, and K. W. Sowerby, "A quantitative analysis of spectral occupancy measurements for cognitive radio," *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, May 2007.

[6] B. E. Khamlichi, C. Abdelaali, L. Ahmed, and J. E. Abbadi, "A quantitative investigation of spectrum utilization in uhf and vhf bands in morocco: The road to cognitive radio networks," *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, October 2016.