Predicting Wireless Channel Utilization at the PHY

Jeffrey Mehlman, Stanford Networked Systems Group, Aaron Adcock, Stanford E.E. Department

***Abstract*—The ISM band is an extremely over utilized communications medium. Wireless LAN technology, Bluetooth, and landline wireless devices all make use of the ISM band. The support vector machine is proposed as a tool for predicting channel availability and thus using the band more efficiently. Data processing, feature extraction, and results are presented. Suggestions for future work in this area are also presented.**

***Index Terms*—Wireless communications, Machine learning, Support vector machine**

I. Introduction

TODAY, wireless communications are an integral part of our daily lives. Depending upon the form and type of communication (e.g. cellular, wireless internet, etc.), wireless protocols use different bands of the radio spectrum. Unlike cellular applications, wireless internet protocols (WiFi 802.11) use the unlicensed Industrial, Scientific, and Medical (ISM) bands to communicate.

Because it is unlicensed and thus free for use (subject to some basic rules), the ISM wireless spectrum in the 2.4-2.5GHz band is an extremely over-utilized communications resource. Wireless LAN technology, landline wireless telephones, microwave ovens, Bluetooth peripherals, and other devices generate both communications traffic and interference in this band. We contend that there are repetitive patterns over short timescales in these bands which can be used to predict the channel state in the near-term future.

As examples of patternistic channel users, we highlight the function of microwave ovens (as dumb interferers) and the 802.11 Media Access Control (MAC) Protocol (as an intelligent user). Microwave ovens generate RF energy in the 2.4-2.5GHz band during portions of the 60Hz AC power waveform. Aside from the initial event of the microwave turning on and the final event of the microwave turning off, the oven RF-energy generation follows a periodic 60Hz activity. On the other hand, the 802.11 MAC protocol provides a simplistic probabilility mechanism for user channel access where a user tries to access the channel in a randomly chosen 10-20μs time slot [2]. If a user attempts to access the channel and it is busy, it "backs off" and waits another random period of time before trying again. In a busy channel, if the user fails in consecutive attempts, it will wait longer and longer periods of time to access the channel. The number of users and their traffic demands have serious impacts on the likelihood of the channel being utilized at any given time. Unfortunately, this probability model is complex due to channel variation, random traffic patterns and the number of users, and the specific backoff algorithms used. An analytic approach to finding this time-varying probability distribution is difficult at best.

Machine learning algorithms which provide reliable prediction of future channel availability would have implications for a number of ubiquitous technologies as well as next-generation products like cognitive radio. For current protocols like 802.11 WiFi, a channel prediction mechanism could allow users to better select time slots for transmission at lower risk of interference, bringing more energy efficient communication and better overall use of the channel. In the future, new technologies will make use of already licensed bands by only using them when it does not interfere with the primary, licensed user. This is generally referred to as cognitive radio [1]. For future cognitive radio protocols, predicting channel freedom to ensure that they do not interfere with the primary user is of the utmost importance [3]. These secondary users of licensed channels, as well as users of the ISM bands, would benefit greatly from a robust channel utilization prediction model. It would allow them to make better use of the spectrum, picking the right time in which to attempt transmission. Unfortunately, since these protocols can not communicate directly with the other users in the band, we must find alternative, passive methods with which to predict channel usage. In this project, we attempt to create a prediction model using only observed signal measurements at the physical layer (PHY).

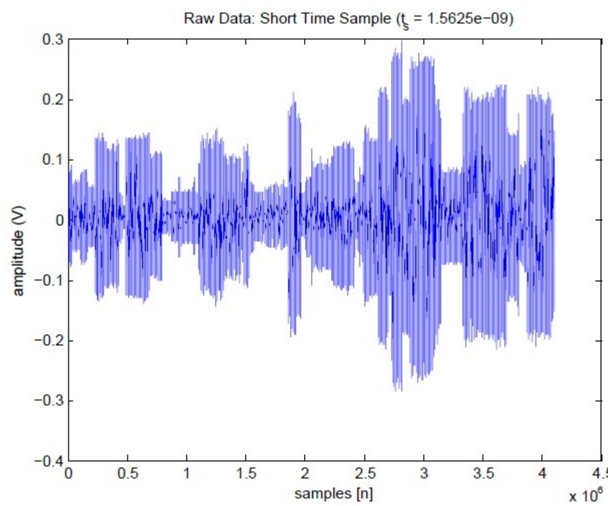

Fig. 1. ISM Band Time Domain Sample (6ms)

Using machine learning techniques, this project aims to utilize signal measurements at the physical layer of wireless ISM band to predict the state of the channel in the near future. We have found no prior research using this approach at the physical layer. In Section II, we review our methodology for this experiment, including data collection, preprocessing, and feature selection. Additionally, we review our supervised machine learning algorithm, making use of the support vector machine (SVM). In Section III, we train and test our algorithm trying a variety of different parameters on portions of a 600ms data sample and provide results. In Section IV we review our results from this experiment, as well as some interesting findings. Throughout the project a number of issues with using a SVM for this application and the size of the collected data were encountered. Some of these issues are highlighted in the paper. Though the methods of this paper require significant computational performance and thus are not feasible for implementation in a wireless communication device, they do yield promising directions for future work.

II. Methodology

In this section, we review the methods used in this project. Beginning with live channel measurement, we highlight methods and considerations for our data needs. Next, we review data preprocessing and feature selection for our learning model. Lastly, we highlight our chosen machine learning algorithm, the support vector machine (SVM), as well as some parametric decisions for this algorithm.

A. Data Acquisition

In order to test our hypothesis, we require a large set of live channel measurements over the entire 100MHz ISM band. Instead of attempting to build large sets of convincing simulation data of wireless traffic, we decided it would be better to obtain live channel measurements. The Stanford Networked Systems Group (SNSG) has access to a RUSK Channel Sounder, the appropriate piece of equipment for collecting this type of data. We collected a 320MHz-wide set of baseband data, centered at 2.45GHz. We took two data samples from Bytes Cafe. The first contained approximately 6ms of raw signals and the second an additional 600ms sample. The 6ms sample is shown in Figure 1 above and was used for initial algorithm testing. The larger data sample was 400MB in size, while the smaller sample was only 4MB. The number of samples is deceptive as the number of time samples is not as important as the number of packets observed. Because of this, and computational considerations, the data was decimated to obtain a more reasonable data set. Because each WiFi packet can be hundreds of microseconds long, the 6ms sample does not include enough packets for anything other than cursory testing. Using scripts provided by RUSK, the channel sounder data files can then be converted into MATLAB .mat files containing the raw data and measurement parameters. With the raw data from the sample sets in hand, we can begin extracting features for our prediction algorithm.

B. Data Preprocessing

The chosen feature set often determines whether or not a learning algorithm is effective. In this case, the raw data contains a huge amount of uninteresting information and noise from the channel. To counteract this, features need to be extracted that are useful for predicting future availability of channels. To select the features, we again note that the most ubiqituous user of the ISM band is 802.11 WiFi, which subdivides the ISM band into smaller frequency channels. In fact, every ISM-band protocol known to the authors uses only some smaller subchannel of the ISM band. Thus, we make the obvious choice to move our data into the frequency domain using a Fast Fourier Transform (FFT), and then subdivide the FFT into smaller frequency subchannels. At this point, we also remove the 220MHz of additional spectrum collected with the RUSK channel sounder, and keep only the 100MHz band (2.4-2.5GHz) of interest. Channel access, in the frequency or time domain, is determined by a power threshold for that band.

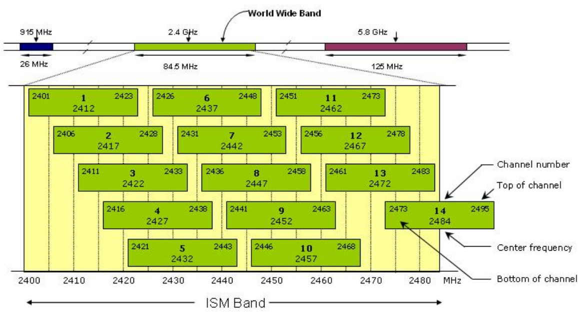

Fig. 2. ISM Channels (http://www.moonblinkwifi.com/2point4freq.cfm)

As shown in Figure 2, The 802.11 family of protocols splits the ISM band into 11 overlapping 20MHz channels, with center frequencies spaced 5MHz apart. Each WiFi Access Point (AP) operates on one of these bands. In a large campus, each AP is instructed to use orthogonal bands, thus channels 1, 6, and 11 are the bands most often utilized. Since channel availability is essentially a function of the average power in a channel, features representing the power in the each channel of interest are the most obvious to extract. To calculate the FFT, more than one time domain data point is required. We slide a time-window of length $T = 2^n$ over our set of raw time measurements, and calculate an FFT for each window. The sliding window is stepped by some fraction of T leaving us with a spectrogram (a set of frequency measurements over time) of the original channel measurements.

Since the length of an FFT is the same as the length of the time window used, the number of FFT samples is much larger than the number of WiFi ISM channels. To further reduce our feature space, we average the FFT samples into 20 bins of 5Mhz over the ISM band. Next, we found that these channel power measurements contained a significant amount of noise from random wireless channel fading and background interference. This made it difficult to set any clear decision boundary for whether or not a packet was in progress. As such, we needed a filtering algorithm which would preserve sharp packet transitions while reducing signal variation due to high frequency noise.

For each frequency channel-time series, we utilized total variation regularization (TVR) de-noising), which is specifically designed to remove noise while preserving large sharp transitions. To denoise a time-series $x \in \mathcal{R}^n$ with TVR, we solve a modified TVR optimization problem

$$\min_{\hat{x}} || \log(x) - \hat{x}||^2 + \lambda \sum_{i=1}^{n-1} |\hat{x}_{i+1} - \hat{x}_i| \tag{1}$$

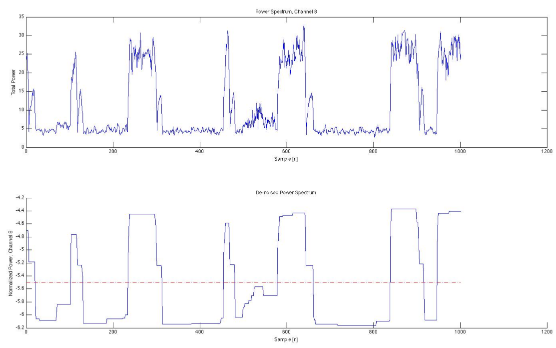

Fig. 3. Noisy and De-noised power-time series (ISM 5Mhz channel = 8)

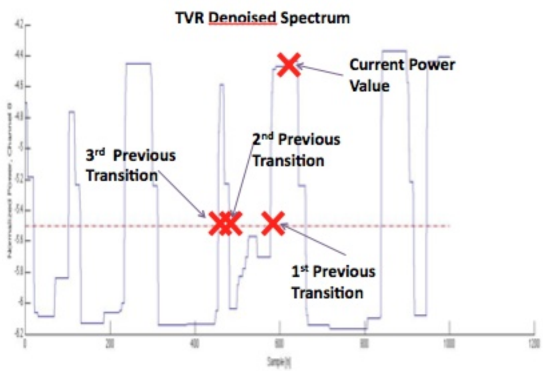

Fig. 4. Transition Feature Vector Extraction

where $\hat{x}$ is the de-noised signal estimate and the parameter λ controls the relative penalty for signal variation. Since interference and changes in signal to noise ratio (SNR) are typically only interesting over orders of magnitude, using the log power on each channel seems a natural choice.

An example of the resulting denoised power-time series for an ISM 5MHz (frequency bin number 8) is shown in Figure 3. In the upper plot, the average power over time for a 5MHz channel is shown. In the lower plot, TVR is applied to this same time series to remove noise while preserving packet transitions. The dotted red line shows our cutoff for deciding whether or not a packet transmission is in progress, which gives the label for our supervised learning algorithm. This is a much cleaner decision boundary in the lower plot, and gives more consistent positive/negative labels for the data set.

C. Feature Extraction

Within each channel, we contend that channel usage will exhibit patterns over time, as a function of the number of users, their traffic demands, the state of the channel interference, etc. This hypothesis regarding patternistic use is at the core of this research effort. We must extract features which clearly represent these patterns, and allow us to accurately predict future patterns.

1) Raw Feature Vectors: Our first attempt at building a feature vector for a single channel involved a simple approach: looking at a set of previous power levels for that channel. Thus, each (label, feature vector) pair for a data point at time t would consist of

- A label indicating whether or not a packet transmission occurred at time step $t+\tau,\ \tau >= 1$, in the future. Note: a time step is approximately 16μs.
- A feature vector consisting of n+1 previous power level samples (from t-n to t).

Unfortunately, the packet patterns we are interested in occur over relatively long time scales (tens of milliseconds), while individual packets occur over much shorter ones (hundreds of microseconds). In order to capture individual packets as well as the large scale patterns, the feature vectors would have to be very long. Additionally, these feature vectors would include significant amounts of useless information. We refer to this approach as the 'raw' feature vector.

2) Transition Feature Vectors: To reduce the dimensionality of the feature vectors and to focus on packet patterns, we developed a new approach. In this case, the (label,feature vector) pair for a time step t consisted of

- A label indicating whether or not a packet transmission occurs at some time step $t+\tau,\ \tau >= 1$, in the future
- One feature with the current power level at time t
- Additional features containing time elapsed since the last N_{PT} "packet transitions"

Packet transitions are defined as the times where a packet starts or ends. The number of previous transitions we consider, N_{PT}, effectively determines how far back we look in the packet history for the current sample, while the current power level determines whether or not a given feature vector f_i occurs during a packet or not. Extraction of a 'packet with 3-transitions' feature vector is illustrated in Figure 4.

3) Prediction Distance τ: The parameter τ is an important design consideration. Ideally, we would like to be able to predict channel utilization at relatively distant times in the future. With this capability, a user could make improved decisions about the likelihood of succesful channel access. It stands to reason that the performance of our prediction algorithm should also become worse as we attempt to predict further into the future.

4) Miscellaneous: During the course of the project, several other methods for feature extraction were tried. For many applications, if the channel is open, it can be used immediately (this would NOT be true in the case of cognitive radio). As such, one of the variants on the methods proposed above involved restricting the data to just the times when the channel was being used. In this case, the attempt was to predict when a transition would occur. Initial attempts at implementing this were slightly more complicated and did not give improved performance over the method outlined above. This case is mentioned as it emphasizes that the transitions are more interesting than the time spent within a packet or in channel dead-time.

D. Application of Machine Learning

The problem that needs to be solved on each channel is a binary classification problem with input data in $\Re^{n+1}$ for the raw feature vectors, and data in $\Re^{N_{PT}+1}$, where $N_{PT}+1$ is

FV Length (# of Transitions)	FV Selection	Predict Steps Ahead	SVM 'C' Value	Channel (train-test)	Training Error	Test Error
10	Contiguous, First 70%	1	10000	(8-8)	0	0.3324
10	Contiguous, First 70%	10	10000	(8-8)	0.1656	0.5821
40	Contiguous, First 70%	10	10000	(8-8)	0.1022	0.46
10	Random, 40%	1	10000	(8-8)	0.0020	0.0061
2	Random, 40%	10	10000	(8-8)	0.2101	0.2339
10	Random, 40%	10	1000	(8-8)	0.2218	0.2583
10	Random, 40%	10	10000	(8-8)	0.2203	0.2373
10	Random, 40%	100	10000	(8-8)	0.2186	0.2625
10	Random, 40%	10	10000	(8-2)	0.1624	0.4408
20	Random, 40%	10	10000	(8-8)	0.2370	0.2328
100	Random, 40%	10	10000	(8-8)	0.2323	0.2745
1000	Random, 70%	10	10000	(8-8)	0.2198	0.2909

TABLE I
SVM CLASSIFICATION RESULTS

the number of time-since-transition features we look at, plus the current power level. Since the patterns we are attempting to classify are quite complicated, and there are no prior results directly studying these patterns, we attempted to classify the data using a common and flexible approach: the SVM. We began with a linear kernel, but expected that the linear kernel would not be optimal. As such, we also tested the Gaussian kernel and the polynomial kernel, since they were easy to implement and are widely used. The Gaussian kernel generally had the best performance, and was used for the tests discussed in the results section.

We form the SVM dual optimization problem with considerations for using different kernels, where

$$\begin{aligned} \max_{\alpha} \ \mathbf{1}^T\alpha - \frac{1}{2}\alpha^T \mathrm{diag}(y) K \mathrm{diag}(y)\alpha \\ \text{s.t } \alpha_i \leq C,\ \alpha_i \geq 0 \\ \alpha^T y = 0 \end{aligned}$$

generates the optimal decision boundary.

We note at this point that the C parameter, which effectively determines how much we penalize incorrect labelings, is chosen to be very large for this application ($C >$1000). This is due to the fact that there tend to be large dead-times in some of the channel measurements, where no packets are transmitted, and small C values led to a skew towards 'no packet' labelings (see figure 5, top plot). If the number of 0-label samples outweighs the number of 1-label samples by a large factor, small C values leads to the algorithm treating a significant portion of 1-labels as outliers and misclassifying them.

After preparing transition feature vectors, we consider two train/test classification cases:

1) Select a contiguous subset of feature vectors from the beginning of a time series, and then test on the latter half of that series. We refer to this as **contiguous training selection**.
2) Select a random subset of feature vectors from the entire series, and then test on the remaining portion of the series. We refer to this as **random training selection**

In the first case, the goal was to show a causal classification ability, i.e. the SVM could be trained earlier in time, and then the results used later in time with good results. We expected the traffic patterns to remain the same over two time periods in close proximity. Due to findings discussed in Section III, the second case with random subset of samples was introduced. This random subset better represents the patterns found over the entire time series because it comes from the entire time series. If the second type of train/test subset selection works better, it would indicate that, given a set of training samples which represent all patterns found over the entire time series, the SVM is capable of predicting channel utilization in the future. A set such as this would need to be constructed from samples coming from a large variety of wireless traffic patterns and situations.

III. RESULTS

A. Raw Feature Vectors

We first consider classification of the raw feature vectors using an SVM. This optimization yields a trivial result, where if τ, the future-prediction distance, is relatively small many of the feature vectors contain power measurements that fall within the same packet as the label. In this case, the SVM essentially guesses that "if the input is high, it will continue to remain high," and vice versa. If an input feature vector represents a transition from high power to low power (or vice versa), the SVM takes the former half of the transition - high power - and guesses that the output will continue to remain high. If τ is very large and the algorithm tries to look more distantly into the future, the SVM is less successful with training and entirely unable to classify a test set (greater than 50% error). This is because the raw features do not contain enough of the packet history, and the information they do contain is obscured by unimportant features, to make anything other than very near term predictions.

If the feature vectors were significantly longer, it is possible that they would catch multiple packet transitions and then begin to classify based upon long-term patterns, but this would require a very large number of features and thus an extremely large training set. Additionally, a great majority of these 'features' would not have any direct relationship with future prediction of packets, and thus would be innefficient for training the classification algorithm. It is clear that this

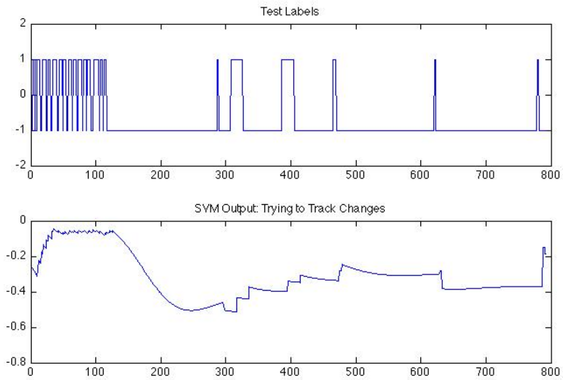

Fig. 5. SVM Output: Tracking but not Classifying

approach is not tenable for packet traffic pattern matching. We do not consider this type of feature vector any further.

B. Transition Feature Vectors

In Table 1, we summarize results for a number of runs with transition feature fectures, including SVM training across a number of parameters, with both contiguous and random subset training selection. The transition feature model clearly performs better for predicting future channel usage than the raw feature model. This model begins to take into account the length of time that the channel has been in the current state and the length of previous packets/dead time. This results in an SVM output that could be used as a probability (by fitting a sigmoid to the output of the training set) of the channel being available in τ time steps.

1) Contiguous Training Selection vs Random Training Selection: As shown in the table, the contiguous training set selection does not perform well for predicting future channels use beyond the next time sample, even when the test set and training set come from the same channel data. This means that simply setting up a program that re-optimizes the SVM using recent channel data will not achieve the performance desired. There is however, some hope here as well. In figure 5, we see that the SVM does in fact track packet transitions. During a long dead period, the output of the SVM begins to rise towards the classification boundary but does not reach it. If we make use of a probabilistic output extension to the SVM and then augment it with Markov forward recursions, it is possible that our results would improve. This is an avenue for future work.

Additionally, random training sample selection from the entire data set gives much better performance. This is because the training set now contains samples very similar to the samples in the test set. This method cannot be used in practice as a wireless device needs to be able to operate causally. This result does suggest that a training set, with enough samples and features, would have enough predictive power for to be useful in many applications. Unfortunately, the size of training set required was computationally beyond our ability to use and, as can be seen from Table 1, one channel of data is not enough data to make accurate predictions on a different channel.

In more closely examining the SVM results with random training, we find that only 10% of the training set (which is in turn only 40% of our sample set for a given channel) are support vectors. From the KKT conditions, we know that alpha values between the bounds (0 and C) are associated with support vectors. This means that only 4% of the total feature vector set is needed to provide 80% test accuracy. In the future, we plan to investigate more deeply if there is special meaning or importance to that 4% of vectors, or if they are just randomly chosen vectors that happen to satisfy the optimization objective in each run of the SVM. If there are common characteristics of support vectors, we may be able to train an SVM algorithm (or some other channel utilization prediction algorithm, for that matter) on only these types of vectors. This also may lead to a more robust solution across many different channels, and the ability to identify proper training data subsets for all channel patterns.

IV. CONCLUSIONS

The application of a SVM to predict future channel use from physical layer measurements was studied. The time domain data had to be taken into the frequency domain using sliding window and a FFT. This allowed individual channels of interest to be used as training and test data. Total variation regulation smoothing was used to remove noise while preserving the sharp packet transitions present in the data. Even after splitting the data into channels and denoising, additional feature extraction was needed to obtain manageable feature vectors. The performance of the algorithm was not as good as desired. Though the algorithm worked for predicting channel use on the same channel as the training set, it failed to achieve reasonable performance on dissimilar channels. To get good results, the training set had to have samples with very similar features as those in the test set, as seen in the superior performance of the randomly selected training sets as compared to the time-contiguous training sets. While this conclusion seems obvious, it does suggest that a large training set with a variety of traffic pattern sample vectors might achieve reasonable performance. Also, the behavior of the SVM output suggests that a probabilistic model, in conjuction with such a training set, would be able to give improved predictions of the length of the current packet or dead time. Future work with an SVM approach would involve finding new features and trying additional kernels. Also, building a training set of appropriate size and content would determine the effectiveness of the SVM for this task. Other future work would include finding appropriate statistical models (beyond using just the SVM output values) to assign probabilities to different channels or time slots that represent the potential availability of that channel/time for use by the device.

REFERENCES

[1] Haykin, S.,"Cognitive radio: brain-empowered wireless communications," *Selected Areas in Communications, IEEE Journal on* , vol.23, no.2, pp. 201- 220, Feb. 2005

[2] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11

[3] Kung, Ling-Hung, "Channel Selection for Cognitive Radio Terminals", CS229 Project Report, 2008