

The effect of the spectrum opportunities diversity on opportunistic access

Hamed Ahmadi*, Irene Macaluso*, Luiz A. DaSilva*[†]

*CTVR Telecommunications Research Center, Trinity College Dublin, Ireland

[†]Department of Electrical and Computer Engineering, Virginia Tech, U.S.A.

Email: ahmadih,macalusi,dasilval@tcd.ie

Abstract—To improve their ability to find spectrum opportunities, intelligent secondary radios (SR) can learn from their past observations and predict possible spectrum opportunities. However, because of the diverse behavior of primary users (PU) in different spectrum bands, spectrum holes exhibit diverse characteristics, which in turn affect the performance of a learning algorithm. This paper studies the effect of the PU's activity on channel predictability. In particular, we introduce a Markov process-based learning algorithm, and we investigate the dependency of its spectrum decisions on the duty cycle (DC) and on the complexity of each channel activity, for both synthetic and real data. Our findings show that the probability of finding a free channel among a group of considered channels strongly depends on the DC and the complexity of the channel activity. Moreover, it is possible to reduce the number of observed channels without compromising the probability of finding a free channel, by only considering the more informative channels.

I. INTRODUCTION

Opportunistic spectrum access may rely on a combination of geolocation databases and spectrum sensing to detect spectrum holes. In their search for spectrum holes, secondary radios (SRs) can learn from their past observations on each channel to predict the next channel state using intelligent learning methods. However, because of the diverse behavior of primary users (PUs) in different spectrum bands, spectrum holes exhibit different characteristics, which in turn affect the performance of a learning algorithm [1], [2].

Different learning algorithms have been proposed in the literature for radios to build a channel occupancy model. In these works, the channel is either occupied by the PUs (busy) or it is unoccupied (free). In [3], Clancy claims that a hidden Markov model (HMM) can be a suitable method to model the channel occupancy as a function of time. In [4], the authors model each channel as a Poisson distribution, and use an HMM to predict the availability of a channel. The HMM is trained with the Baum-Welsh algorithm (BWA) [5], predicting the presence of PUs to avoid transmission collisions. An SR will occupy an idle channel until a PU becomes active in that channel, then it will switch to another predicted idle channel. Simulation results show that the probability of collision is reduced compared to a random selection of channels to be sensed by the SR. An artificial neural network is proposed in [6] to predict the channel state for the next time slot. In [7], the authors evaluate the performance of their proposed neural network when the statistics of the channel are changing. We

propose a modified, less complex HMM in [8]. We compare the prediction accuracy of our proposed method with that of conventional HMM and illustrate that it achieves the same prediction accuracy with much less computational complexity.

The aforementioned works test their proposed algorithms on data which are generated based on some assumptions on the probability distribution of the PU activity (synthetic data), and none of them applies their algorithm on data that are collected from actual sensing (real data).

There are some recent works which use real data in their studies. In [9], authors conduct spectrum measurements in Guangzhou city, and then approximate the prediction error with the beta distribution. Kone et al. propose frequency bundling in [10], where secondary devices build reliable channels by combining multiple unreliable frequencies into virtual frequency bundles. Their experiments on real data show that bundling random channels together can provide sustained periods of reliable transmission. In [1], [2], we present a reinforcement learning-based method for spectrum opportunity prediction. The learning algorithm is applied to real data and the results are analyzed with respect to the duty cycle (DC) of the channels and to the complexity of the PU's activity.

In this work, we analyze the probability of finding a free channel among a group of observed channels using the Markov process-based learning algorithm we proposed in [8]. We test the algorithm on both synthetic and real data. In particular, we analyze the prediction accuracy of the learning method with respect to the characteristics of the spectrum occupancy sequences, as we first proposed in [1]. Moreover, we investigate the effect that the number of observed channels has on the performance of the Markov process-based learning algorithm. Our findings show that, if an SR is able to select the channels with the best characteristics, reducing the number of observed channels has little effect on the performance of the learning algorithm.

II. SYSTEM MODEL AND LEARNING ALGORITHM

In this study, the spectrum under consideration is divided into K channels and each PU, when it transmits, will occupy at least one of the channels. The presence of a PU on a channel is represented with a "1" and the absence of PUs with a "0". We assume that channel sensings are performed periodically and the channel sensing is ideal, i.e., the effects of noise,

missed detection and false alarm errors are negligible. Each SR uses a Markov process-based learning algorithm to learn the occupancy model of each channel and predict the availability of the channels for the next time slot.

Our Markov process-based learning algorithm has N states and M possible observations, where \mathcal{S} and \mathcal{O} represent the sets of possible states and observations, respectively. We denote sequences of states by \mathbf{x} , and we use \mathbf{y} to indicate the sequence of observations. Each element of \mathbf{x} , denoted by $x(t) \in \mathcal{S}, \forall t$, is the state at time t ; each element of \mathbf{y} , denoted by $y(t) \in \mathcal{O}, \forall t$, is the observation at time t .

The transition probabilities between the states are stored in an $N \times N$ matrix (\mathbf{A}). The distribution of the observation outcomes at each state is described by the respective column vector of the $N \times M$ emission matrix \mathbf{B} . We represent this Markov process by $\lambda = \{\mathbf{A}, \mathbf{B}, \pi\}$, where π is the initial state distribution.

We have previously compared the Markov-based learning algorithm with conventional HMM in [8]. In that work, we showed that the Markov-based learning algorithm and HMM have similar performance in predicting the spectrum opportunities. However, the computational complexity of our Markov-based learning algorithm is much less than HMM. Thus, in this work we only apply our learning algorithm.

A. Learning algorithm

We train the learning algorithm off-line over a training sequence, which means that we find λ for the training sequence. In this Markov process-based learning model, we do not specify the number of states in advance. In other words, the number of states (i.e. N) in our model grows dynamically as learning proceeds. In this system we have two possible observations ($M = 2$). We observe a zero when we sense a free channel, and we observe a one by sensing a busy channel.

The transitions between states depend on the length of the string of consecutive zeros or ones observed. This means that in our system each state represents a number of observed consecutive zeros or ones. Positive states represent the number of observed consecutive ones, and non-positive (negative and zero) states represent the number of observed consecutive zeros. At the beginning, the system can either be in state zero or state one. Being in a positive state, the system moves to the next higher state whenever a busy channel (one) is observed, and it goes back to state zero whenever a free channel (zero) is observed. On the other hand, when the system is in a non-positive state, it moves to the next lower (negative) state whenever a zero is observed and it moves to state one whenever a one is observed. The system will expand itself on the fly by adding new states as needed. Suppose that the maximum number of negative and positive states in the Markov chain, after the training, are q and p , respectively. Then, the set of states will be $\mathcal{S} = \{s_{-q}, \dots, s_0, \dots, s_p\}$ which has the cardinality of $N = q + p + 1$. Fig. 1 shows the Markov chain and its state transitions.

During the training phase, we create the Markov chain using the training data set and based on the aforementioned

rules. Then, it is possible to count the number of times each particular transition or output observation is applied in a set of training data. As proven in [5], counting functions for the output observations provide maximum likelihood estimates for the desired model parameters. Therefore, elements of transition and emission matrices will be computed by:

$$a_{i,j} = \frac{f_{i,j}(\mathbf{x})}{\sum_{n=-q}^p f_{i,n}(\mathbf{x})} \quad \forall i, j \in \{-q, \dots, p\}, \quad (1)$$

$$b_{i,j} = \frac{g_{i,j}(\mathbf{x}, \mathbf{y})}{\sum_{m=1}^M g_{i,m}(\mathbf{x}, \mathbf{y})} \quad \forall i \in \{-q, \dots, p\}, j \in \{1, \dots, M\}, \quad (2)$$

where $a_{i,j}$ is the transition probability from s_i to s_j , and $b_{i,j}$ is the probability of o_j at s_i . The counting functions $f_{i,j}$ and $g_{i,j}$ simply count the number of transitions from state s_i to state s_j and the number of observations o_j at state s_i , respectively.

B. Prediction

During the training process, the transition and emission probabilities over the observations can be easily calculated by (1) and (2). As there are only two possible observations, the Markov chain is designed in a way that only two states are reachable from the current state. Moreover, the probability of transition from the current state to either of the two states is equal to the probability of observing either a zero or a one. Thus, having the emission matrix (\mathbf{B}) means having the probability of the next state as well as the distribution of observations. This property makes the states sequence trackable. Therefore, at each time slot we know the probability of each observation. As a result, we can predict the observation by:

$$\check{y}(t) = \begin{cases} 0 & p(x(t), 0|\lambda) \geq p(x(t), 1|\lambda) \\ 1 & \text{otherwise} \end{cases}, \quad (3)$$

where $\check{y}(t)$ indicates the predicted observation for time t .

In case of an inaccurate prediction, the system will notice the prediction error after observing $y(t)$. Since $x(t + 1)$ only depends on the observation outcome rather than on the predicted result, the system will move to the correct state and errors will not propagate. Retraining the system is only needed when the statistics of the behavior of PUs on the channel are changing. To account for this, the system can be retrained after a certain number of time slots or whenever the prediction accuracy drops below a certain threshold.

III. IMPACT OF LZ AND DC ON THE PREDICTION ACCURACY

In this section we analyze the performance of the proposed Markov process-based learning algorithm with respect to the characteristics of the spectrum occupancy sequences. As we discussed in [1], a spectrum occupancy sequence can be characterized in terms of the observed DC and the complexity of the PU activity. In [1] and [2], we showed that the traffic load, of which the DC is an estimation, is not sufficient to characterize the performance of learning algorithms applied to a dynamic channel selection problem. In particular, the amount of structure in channel availability, i.e. the complexity

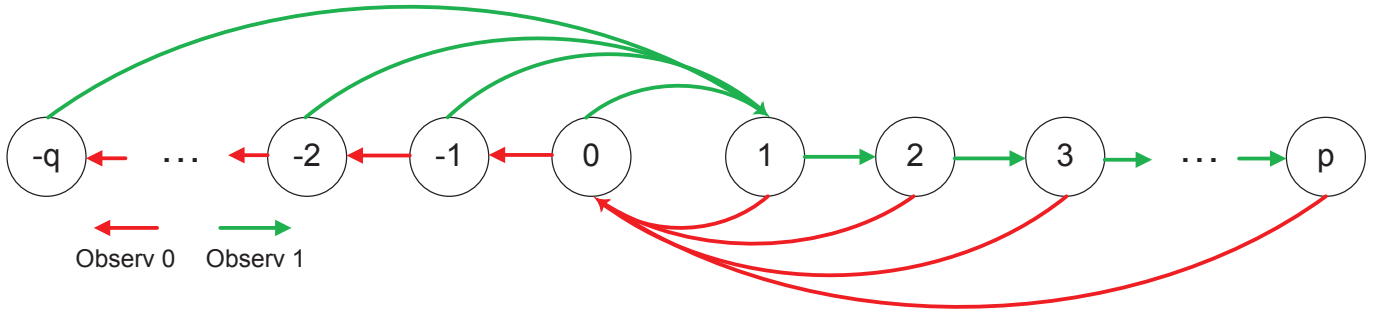


Fig. 1. The Markov chain state transitions.

of the PU activity, cannot be overlooked when analyzing the prediction accuracy of learning-enhanced dynamic spectrum access techniques. In [1] and [2], we quantified the amount of structure in a spectrum occupancy sequence using the Lempel-Ziv complexity [11], which measures the rate of production of new patterns in a binary sequence. The same measure of complexity is used in this work.

To analyze the performance of the Markov process-based learning algorithm, we modeled the K channels as independent random variables. Each channel is the realization of a 2-state first order Markov chain (MC). Therefore we generated MCs with different values of stationary distribution $\delta = [\delta_0, \delta_1]$, i.e. DC, and LZ complexity. For an ergodic source the Lempel-Ziv complexity equals the entropy rate of the source [12], which for a Markov chain X is given by:

$$h(X) = - \sum_{ij} \delta_i p_{ij} \log p_{ij}. \quad (4)$$

We considered 5 possible δ_0 values in the range $0.5, \dots, 0.9$. For each of these values, we considered 5 transition probability matrices, each corresponding to a different value of entropy rate, thus obtaining 25 different transition probability matrices. Finally the Markov process-based learning algorithm has been applied to all the possible combinations of $K = 3$ channels over the 25 possible transition matrices.

Fig. 2 shows the prediction accuracy obtained by the Markov process-based learning algorithm as a function of the average entropy rate and the probability p_f of at least one free channel existing, calculated as:

$$p_f = 1 - \prod_{i=1}^N \delta_{1,i}, \quad (5)$$

where $\delta_{1,i}$ is the stationary distribution of the i^{th} channel, i.e. the DC of the i^{th} channel.

Each point in the figure corresponds to one scenario, i.e. a combination of 3 channels. For each instance, we run 10^2 independent simulations. For each simulation, first the occupancy model of each channel is computed using the Markov process-based learning algorithm described in the previous section over a training sequence of 10^3 time steps, then the resulting model is evaluated over a sequence of $2 * 10^4$ time steps. For each simulation the prediction accuracy is computed according to

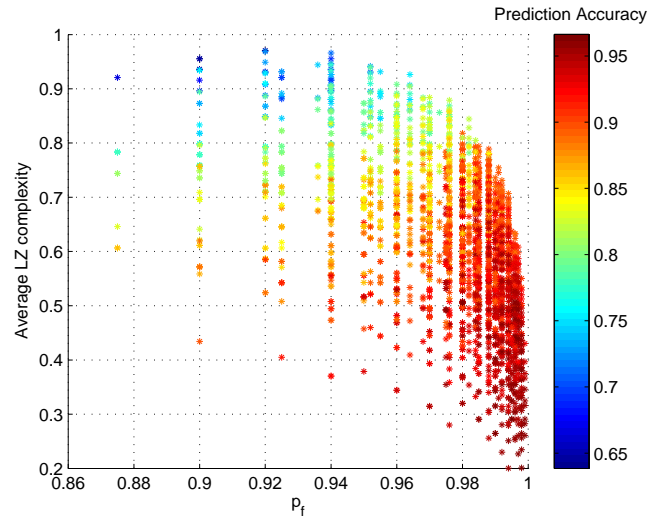


Fig. 2. Prediction accuracy of the Markov process-based learning algorithm as a function of the average LZ complexity and the probability of at least one free channel existing. Each point represents a particular instance of the Markov process-based learning algorithm applied to $K = 3$ channels. The total number of possible combinations which we analyzed is $\binom{25}{3} = 2300$.

the number of times that a free channel was selected or all the channels were correctly predicted as being busy over the length of the trial.

Each point in Fig. 2 represents the average of the results obtained for each of the 10^2 experiments. As expected, the prediction accuracy increases with p_f . However, it can be observed that the performance of the Markov process-based learning algorithm is also strongly dependent on the complexity of the channel activity. For a certain value of p_f , the variation of the prediction accuracy is up to 30%. These results are in accordance with our findings in [1] as to the importance of considering how various degrees of complexity influence the agent's ability to learn and to exploit the spectrum holes.

In order to provide a better understanding of the Markov process-based learning performance, Fig. 3 shows the probability of success (P_{suc}), i.e. the probability of selecting a free channel, corresponding to three configurations with respect to the number of channels. To facilitate the comparison, the three configurations refer to the same stationary distribution

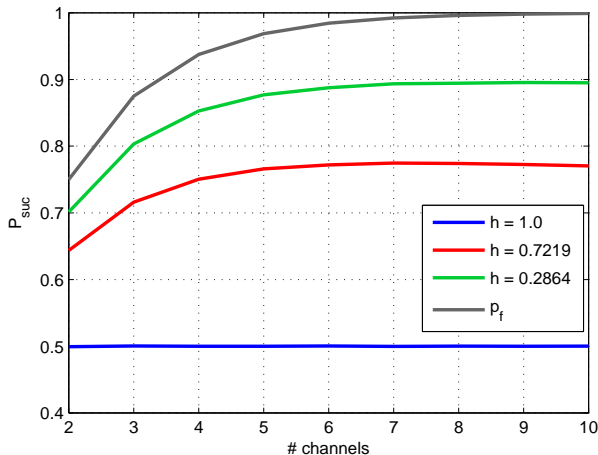


Fig. 3. Probability of success of the Markov process-based learning algorithm as a function of the number of channels. Each curve refers to combinations of different numbers of channels characterized by the same DC and different values of entropy rate h . The grey curve on the top shows the probability of at least one free channel existing, as a function of the number of channels.

$\delta = [0.5, 0.5]$. In other words, for a given number of channels, the three configurations refer to the same p_f , e.g. for $K = 3$ the corresponding p_f is $1 - 0.5^3 = 0.875$. Furthermore, each of the channels in a configuration is modeled with the same Markov transition matrix.

In the first case (blue line in figure), the Markov transition matrix is $\begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$ and the corresponding entropy rate is 1.0. In the second case (red line in figure), the Markov transition matrix is $\begin{pmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{pmatrix}$ and the corresponding entropy rate is 0.7219. In the third case (green line in figure), the Markov transition matrix is $\begin{pmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{pmatrix}$ and the corresponding entropy rate is 0.2864. For each scenario, we run 10^2 independent simulations. Analogously to the results in Fig. 2, for each simulation, we used a training sequence of 10^3 time steps and an evaluation sequence of $2 * 10^4$ time steps.

Despite referring to the same p_f for each number of channels, the Markov process-based learning performance is quite different in the three cases. It should be noted that in the third scenario, which corresponds to a highly predictable channel usage, the probability of success of the Markov process-based learning algorithm is close to p_f (grey line in figure). Clearly the probability of success can never exactly match p_f due to the stochastic nature of the channel usage. However, the more predictable, i.e. the less complex, the channel activity is, the higher the probability of success is, independently on the number of channels.

IV. REDUCING THE NUMBER OF OBSERVED CHANNELS

The results in Fig. 3 show that, as expected, the probability of success of the Markov process-based learning algorithm generally increases with the number of channels. It is interesting to observe that the increment of the probability of success P_{suc} decreases when the number of channels increases, in accordance with our findings in [2]. In other

words, the performance improvement becomes negligible after a certain point. Also, the increment of P_{suc} obtained by increasing the number of observed channels decreases when the complexity of the channel activity increases. Indeed, in the case of highly unpredictable channels (blue line in Fig. 3), increasing the number of observed channels does not result in an improvement of the P_{suc} .

In light of the above results, it is important to investigate not only how many channels an SR should observe, but also how those channels should be selected in order to achieve a high P_{suc} whilst keeping the number of observed channels to a minimum. To this end, we analyze the effect of reducing the number of observed channels on P_{suc} , while also considering the impact of selecting channels characterized by different values of DC and LZ complexity. In Table I, we consider 4 channels, where 2 channels are characterized by DC = 0.6, one with medium complexity and one with low complexity; the other 2 channels correspond to DC = 0.5, one with high complexity and one with medium complexity. The P_{suc} of the Markov process-based learning algorithm is 0.74, when all 4 channels are used. Table I shows the P_{suc} when we exclude one the channels, and we use the remaining 3 channels. The results show that removing the channel with low complexity and higher DC yields the worst performance. In other words, excluding the channel with a higher DC, i.e. selecting a combination characterized by a higher p_f , does not always correspond to a higher P_{suc} . The performance of the Markov process-based learning algorithm when using 3 channels (and excluding the channel with the higher DC and medium complexity) is very close to the probability of success corresponding to the 4 channels. Moreover, we should also note that p_f for the case of using 3 channels is lower than the time that we use 4 channels.

TABLE I
THE EFFECT OF REDUCING THE NUMBER OF CHANNELS UNDER CONSIDERATION FROM 4 CHANNELS TO 3 ON P_{suc} AND p_f . THE RESULTS ON EACH COLUMN PRESENT P_{suc} AND p_f WHEN THE CORRESPONDING CHANNEL IS NOT USED.

	DC = 0.6		DC = 0.5	
	Medium LZ (0.81)	Low LZ (0.29)	High LZ (1.0)	Medium LZ (0.81)
P_{suc}	0.724	0.65	0.7	0.69
p_f	0.85	0.85	0.82	0.82

In Fig. 4, we expand our experiment and consider 6 channels, where 3 channels are characterized by DC = 0.6 (low, medium and high complexity) and the other 3 channels correspond to DC = 0.5 (low, medium and high complexity). The star in the figure corresponds to the P_{suc} of the Markov process-based learning algorithm when all 6 channels are used. The circles refer to P_{suc} when one of the channels is removed and the remaining 5 channels are used. The squares correspond to the P_{suc} when 2 channels are excluded and the remaining 4 are used. It should be noted that if we exclude either one of the two channels characterized by high complexity (independently on the DC), the resulting P_{suc} is quite close

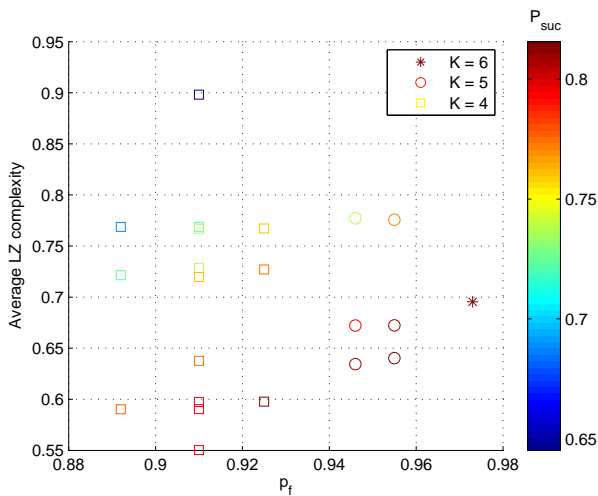


Fig. 4. The effect of reducing under consideration channels from 6 channels to 5 and 4 on P_{suc} . The star, circles and squares represent the P_{suc} using 6, 5 and 4 channels, respectively.

to the performance corresponding to 6 channels. In fact, the difference in performance is less than 0.01. Although further reducing the number of used channels overall yields a lower P_{suc} , excluding the 2 channels corresponding to DC=0.6 with high and medium complexity, or the 2 channels with high complexity, does not affect too much the performance of the Markov process-based learning algorithm. The difference in performance in these cases is less than 0.02.

As shown, the P_{suc} strongly depends on the DC of the channels and the complexity of the sequence. Therefore, to be able to remove the less informative channels it is important to know the LZ complexity in advance. The LZ complexity converges to the entropy rate of a sequence if we compute it over infinite samples. We conducted 10^3 independent simulations and computed the LZ complexity values of binary sequences generated according to four different channel transition matrices. The LZ complexity of each sequence is computed using the algorithm described in [13]. Table II presents the mean of computed LZ complexities. Since all the variances are less than $3 \cdot 10^{-3}$, they are not presented in the table. As expected, the table shows that the LZ complexity estimation improves as the number of given samples increases. In fact, the difference between the complexity estimation with 1000 samples and the entropy rate computed according to (4) is just 0.06.

V. IMPACT ON REAL SPECTRUM DATA

In this section, we provide our analysis and simulation results over the Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen University data set [14]. In this data set the power spectral density (PSD) is recorded across several bands. In this work, we use the data collected over the 2.4-GHz ISM and GSM 1800 bands. Moreover, we consider all the channels with $DC \in [0.3, 0.8]$.

In our simulations, we have applied the Markov process-based learning algorithm to all combinations of $K = 4$

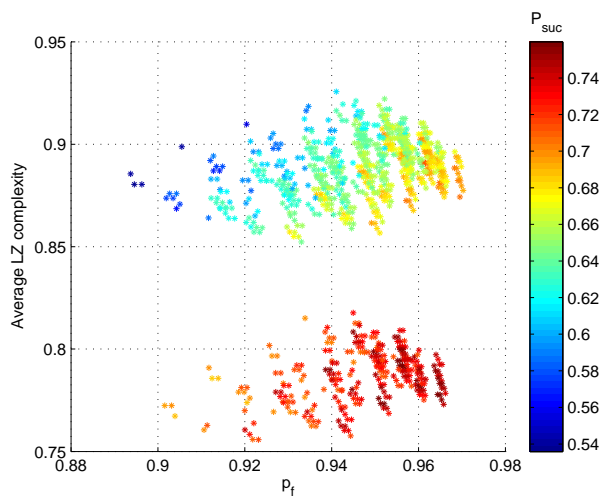


Fig. 5. Probability of success (P_{suc}) of the Markov process-based learning algorithm as a function of the average LZ complexity and the probability of at least one free channel existing. Each point represents a particular instance of the Markov process-based learning algorithm applied to $K = 4$ channels of 2.4-GHz ISM band.

channels over the set of considered channels, and we have computed the probability of finding a free channel (P_{suc}). The data set is recorded over 12 hours and it consists of 25000 samples. We train the Markov process-based algorithm over the first 1000 data samples and test it over the remaining 24000 samples. At each time slot the algorithm either selects the channel with the highest predicted probability of being free, or predicts that all the channels are busy.

Fig. 5 and Fig. 6 present the P_{suc} as a function of LZ complexity and p_f for the 2.4-GHz ISM and GSM 1800 bands, respectively. Similar to synthetic data, in both Fig. 5 and Fig. 6 we observe a higher P_{suc} when the LZ complexity and p_f are respectively lower and higher. In the ISM band the p_f is higher than for GSM 1800 and also the channel activity is less complex (on average). Accordingly, the Markov process-based learning algorithm is more successful in finding a free channel among its four considered channels over the ISM band.

As mentioned, each point in both Fig. 5 and Fig. 6 represents the P_{suc} when 4 channels are considered. Here, we have computed the P_{suc} for all $\binom{4}{3}$ combinations in order to investigate the effect of excluding a channel. For the ISM band data, the difference of the best 3 channels P_{suc} and the original P_{suc} is lower than 10^{-2} for more than 55% of the points, and for only about 7.8% of the points it is higher than $5 \cdot 10^{-2}$. The results of selecting 3 channels among the available 4 are even better for the case of GSM data. For GSM data, the difference of the best 3 channels P_{suc} and the original P_{suc} is lower than 10^{-2} for more than 66% of the points, and only 2.7% of the points have a P_{suc} difference higher than $5 \cdot 10^{-2}$. These results show that, similar to synthetic data, removing the less informative channels will not significantly reduce the probability of finding a free channel in real data.

TABLE II
ESTIMATED LZ COMPLEXITY OVER DIFFERENT SEQUENCE LENGTHS.

Channel Transition Matrix	Number of Samples							Entropy Rate (h)
	1000	1500	2000	2500	3000	3500	4000	
$\begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$	1.0628	1.0535	1.0472	1.0437	1.0409	1.0389	1.0375	1.00
$\begin{pmatrix} 0.65 & 0.35 \\ 0.35 & 0.65 \end{pmatrix}$	0.9938	0.9831	0.9794	0.9748	0.9722	0.9708	0.9687	0.9341
$\begin{pmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{pmatrix}$	0.8626	0.8542	0.8502	0.8483	0.8447	0.8427	0.8417	0.8113
$\begin{pmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{pmatrix}$	0.6485	0.6419	0.6376	0.6352	0.6328	0.6303	0.6303	0.6098

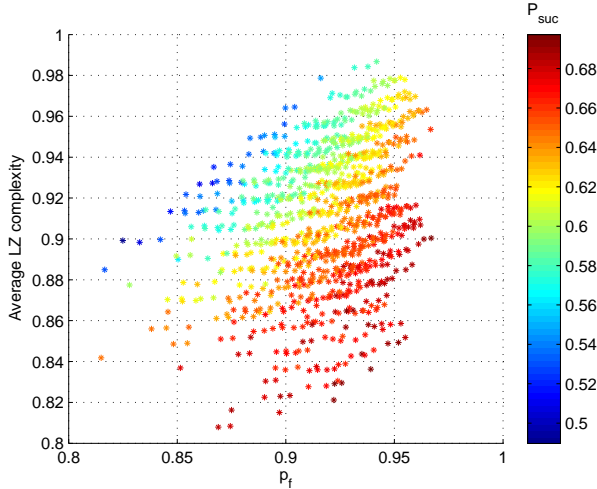


Fig. 6. Probability of success (P_{suc}) of the Markov process-based learning algorithm as a function of the average LZ complexity and the probability of at least one free channel existing. Each point represents a particular instance of the Markov process-based learning algorithm applied to $K = 4$ channels of GSM 1800.

VI. CONCLUSIONS AND FUTURE WORK

In this study we presented a Markov process-based learning algorithm to predict the channel occupancy in an opportunistic channel access scenario. We also investigated the dependency of the prediction accuracy on the DC of a channel and on the complexity of the channel occupancy model. Our findings show that the predictability of a channel strongly depends on its DC and LZ complexity. The results also showed that not considering the channels with high LZ complexity and/or DC does not strongly affect the probability of finding a free channel, and therefore, by removing them from the consideration list we can save time and energy.

Moreover, the LZ complexity can be approximated with a limited number of samples. Therefore, we are investigating the possibility of using the estimated LZ and DC in order to allow an SR to focus its resources on a limited subset of channels without compromising the performance of the learning algorithm.

ACKNOWLEDGMENT

This material is based upon works supported by the Science Foundation Ireland under Grants No. 10/CE/I1853 and 10/IN.1/I3007.

We gratefully acknowledge the use of wireless data from Spectrum Data Archive of the Institute of Networked Systems at RWTH Aachen University.

REFERENCES

- [1] I. Macaluso, T. Forde, L. DaSilva, and L. Doyle, "Recognition and Informed Exploitation of Grey Spectrum Opportunities," *IEEE Vehicular Tech. Mag.*, vol. 7, no. 2, pp. 85–90, 2012.
- [2] I. Macaluso, D. Finn, B. Ozgul, and L. DaSilva, "Complexity of Spectrum Activity and Benefits of Reinforcement Learning for Dynamic Channel Selection," *IEEE J. Sel. Areas Commun. - Cognitive Radio Series*, 2013.
- [3] T. Clancy and B. Walker, "Predictive dynamic spectrum access," in *SDR forum tech. conf.*, 2006.
- [4] I. Akbar and W. Tranter, "Dynamic spectrum allocation in cognitive radio using Hidden Markov Models: Poisson distributed case," in *IEEE SoutheastCon.'07*, 2007.
- [5] J. Candy, *Bayesian signal processing: classical, modern, and particle filtering methods*. Wiley-Interscience, 2009.
- [6] V. Tumuluru, P. Wang, and D. Niyato, "A neural network based spectrum prediction scheme for cognitive radio," in *IEEE ICC'10*, 2010.
- [7] —, "Channel status prediction for cognitive radio networks," *Wireless Communications and Mobile Computing*, 2010.
- [8] H. Ahmadi, Y. H. Chew, P. K. Tang, and Y. A. Nijssure, "Predictive opportunistic spectrum access using learning based hidden markov models," in *IEEE PIMRC'11*, 2011.
- [9] C. Song and Q. Zhang, "Intelligent dynamic spectrum access assisted by channel usage prediction," in *IEEE INFOCOM'10*, 2010.
- [10] V. Kone, L. Yang, X. Yang, B. Zhao, and H. Zheng, "On the feasibility of effective opportunistic spectrum access," in *Proceedings of the 10th annual conference on Internet measurement*. ACM, 2010, pp. 151–164.
- [11] A. Lempel and J. Ziv, "On the complexity of finite sequences," *IEEE Trans. on Information Theory*, vol. 22, no. 1, pp. 75–81, 1976.
- [12] J. Ziv, "Coding theorems for individual sequences," *IEEE Trans. Inform. Theory*, vol. 24, no. 4, pp. 405–412, 1978.
- [13] F. Kaspar and H. Schuster, "Easily calculable measure for the complexity of spatiotemporal patterns," *Physical Review A*, vol. 36, no. 2, pp. 842–848, 1987.
- [14] M. Wellens and P. Mähönen, "Lessons learned from an extensive spectrum occupancy measurement campaign and a stochastic duty cycle model," *Mobile Netw. App.*, vol. 15, no. 3, pp. 461–474, 2010.