Self-Organizing Maps for Improving the Channel Estimation and Predictive Modelling Phase of Cognitive Radio Systems

Aimilia Bantouna, Kostas Tsagkaris, and Panagiotis Demestichas

University of Piraeus, 80 Karaoli Dimitriou str., 18534, Piraeus, Greece

Abstract. Rapid evolution of wireless communications, especially in terms of managing and allocating the scarce, radio spectrum in the highly varying and disparate modern environments asks for a technology for its intelligent handling. Cognitive radio systems (CRSs) have been proposed as one. A typical CRS implements a so called "cognition cycle", during which it senses its environment, evaluates a set of candidate radio configurations to operate with and finally decides and adjusts its operating parameters expecting to move the radio toward an optimized operational state. As the process is often proved to be rather arduous and time consuming, learning mechanisms that are capable of exploiting measurements sensed from the environment, gathered experience and stored knowledge can be judged as rather beneficial in terms of speeding it up. Framed within this statement, this paper introduces and evaluates a mechanism which is based on a well-known unsupervised learning technique, called Self-Organizing maps (SOM), and is used for assisting a CRS to predict the bit-rate that can be obtained, when it senses specific input data from its environment, such as Received Signal Strength Identification (RSSI), number of input/output packets etc. Results show that the proposed method is successful up to a percent of 75.4%.

Keywords: Cognitive Radio Systems (CRS), Cognition Cycle, Learning, Self-Organizing Maps (SOMs).

1 Introduction

Each wireless communication needs its different piece of a specific limited natural source, the electromagnetic radio spectrum, whose current static assignment often leads to its underutilization. Accordingly, the deployment of a technology which will have the ability to exploit the underutilized frequency bands is needed.

Cognitive radio systems have been proposed as a promising technology for this cause [1] [2] due to their ability to adjust their function according to the external, environmental stimuli, the demands of the users/applications and their past experience. Based on this ability, future cognitive systems will be able to change their parameters (carrier frequency, radio access technology, transmit power, modulation type etc), observe the results and decide which is the best combination of those parameters in order to get into a better operational state. So, in terms of flexible spectrum

K. Diamantaras, W. Duch, L.S. Iliadis (Eds.): ICANN 2010, Part II, LNCS 6353, pp. 382 – 391, 2010.

management concept, use of cognitive systems will allow the use of a spectrum band in different radio access technologies (RATs) [3], [4].

A typical cognitive operation consists of three cooperative phases (**Fig. 2**) [2] [3]. During, the first phase, known as radio scene analysis, the system takes measurements from the environment (e.g. conditions related to interference) and explores of different configurations. In the 2nd phase, channel estimation and predictive modeling, the output of the 1st phase is used for discovering the capabilities of each candidate configuration, wherein past experience of the system may also be used. Finally, in the last phase, known as "configuration selection", the system adjusts its operation parameters according to its selected best configuration. In particular during the 2nd phase of the cognition cycle, numerous candidate configurations for the CRS need to be evaluated. This is proved to be a very arduous and time-consuming task, which can be relaxed by using learning mechanisms.

Supervised learning through neural networks-based schemes have been used recently in [4] [5]. Bayesian networks have been also used in [3]. Our proposal is an unsupervised neural network technique called Self-Organizing Maps (SOMs). SOM is a technique for representation and classification of multidimensional data into 2D maps. These maps consist of rectangular or hexagonal cells on a regular grid and according to the technique; each data sample correlates with one cell/neuron of the map in order to be closest to those who are most like it. In this term, the created map represents the similarity of the data and their classification. Due to their ability, SOMs are very popular in data mining problems such as identification of illicit drugs [6], chemical analysis [7], document collections [8], speech recognition [9], identification of a cancer cell gene [10], hematopoietic differentiation [11] and more. In our case we examined the possibility of connecting parameters observed during a configuration, such as noise, received signal strength Indication (RSSI), errors (input and output), packets (received and sent) and Bytes (received and sent) with an anticipated QoS metric namely the bitrate that can be achieved under the configuration in question.

Finally, in order to validate the technique, we have setup and executed a program by using MATLAB SOM toolbox. The developed SOMs are trained with measurements that have taken place in a real working environment within our University premises. The method exhibits a satisfactory capability of predicting the achieved bitrate when facing both known and unknown exemplars (combinations of monitored parameters given a configuration).

The rest of the paper is structured as follows: a review of SOM technique and a short analysis of our proposal are presented in section 2. Section 3 presents the results of our test cases, including a comparison of different versions of our program (section 3.1), the choice of the variables of the input data samples (section 3.2), different cases which include different number of data samples (section 3.3) and different evaluating scenarios with different parameters of SOM technique (section 3.4). Finally, the paper is concluded in section 4.

2 Self-Organizing Maps (SOMs) and Contribution

SOM, introduced by [12], is a type of neural network that belongs to the set of unsupervised learning techniques. An overview of its theoretical foundation may be found in [13]. It is a technique for representation and classification of multidimensional data into 2D maps. These maps consist of cells, whose shape is rectangular or hexagonal, on a regular grid. According to the technique, each data sample correlates with one cell/neuron of the map, called Best Matching Unit (BMU). The process during which the BMU and a neighborhood around it stretch towards the inserted data sample (**Fig. 1**) is called SOM training and results in an ordered SOM map where similar data samples are close. In this term, the created map represents the similarity of the data and their classification.

Two different training algorithms are used for SOM training: the sequential and the batch training algorithm. In the first algorithm each data sample is inserted in the training process individually, thus affecting its own best matching unit (BMU) and a neighborhood around it. In the 2nd algorithm, all data samples are inserted together in the process and eventually affect their BMUs and neighbors in parallel.

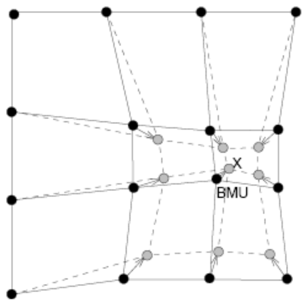

Fig. 1. The inserted data sample x affects its BMU and its neighborhood. The solid and dash-doted lines correspond to the situation before and after the input of the data sample [14]

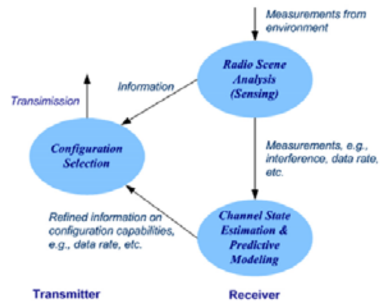

Fig. 2. Simplified representation of cognitive radio cycle [2]

As mentioned in section 1, we will examine the possibility of connecting parameters observed during a configuration, such as noise, received signal strength Indication (RSSI), errors (input and output), packets (received and sent) and Bytes (received and sent) with one QoS metric, the bitrate in order to predict it. This is done by using the unsupervised training technique of SOM. However, as also mentioned in section 1, SOM is a technique for representation and classification of multidimensional data into 2D maps. So how could it be useful in our case?

To begin with, measurements that have taken place in a real working environment within the premises of our University were used to create different data files. Each file comprised a different test case including different combinations of our parameters. Each column, apart from the last one, referred to a different parameter of the data sample while each row corresponded to one different data sample (see **Fig. 3**). Finally, the last column of the data file was the measured value of the bitrate which was related to the data sample and was taking part only for distinguishing them. It is essential to mention that no normalization of the data samples has taken place.

In the sequence the created data file and SOM toolbox v.2 of MATLAB [14] were used to train the SOM. For facilitating the analysis, SOM toolbox offers the ability to use labels in order to distinguish the data samples. In our case, each label corresponds to the measured value of the bitrate of the data sample. However, the fact that more than one data samples may have the same BMU (m_c) leads to the fact that each cell of the map may have more than one labels appearing more than once. As SOM toolbox offers enough different ways for labeling the map, we used three of them ending up with three different versions of our program. The first way (VOTE) is to put on each cell only the most frequently appearing label, the second one (ADD1) is to put all labels while the third one (FREQ) is to put all labels, like in case of ADD1, but in descending order with respect to their appearance frequency and followed by the number of appearances.

1	5					
2	#n RSSI	IPKTS	OPKTS	IBYTES	OBYTES	
3	-35	926	1750	32630	880650	54
4	-32	908	1680	31932	845424	54
5	-32	888	1680	31272	845424	54
6	-35	888	1679	31272	845468	54
7	-36	890	1683	31338	845598	54
8	-36	960	1818	33812	915702	54
9	-36	890	1682	31338	845598	54
10	-36	928	1766	32756	887366	54
11	-34	920	1731	32328	873760	48
12	-33	926	1751	32586	880650	54
13	-33	924	1750	32564	880650	54
14	-32	894	1684	31494	845688	54

Fig. 3. Matlab Data File: the number of the first line refers to the number of the parameters of the configuration, here equal to 5 (RSSI, Input PacKeTS, Output PacKeTS, Input BYTES, Output BYTES), and the last column refers to the bitrate which was used as label. Each line is a data sample and each column is a different parameter of the configuration.

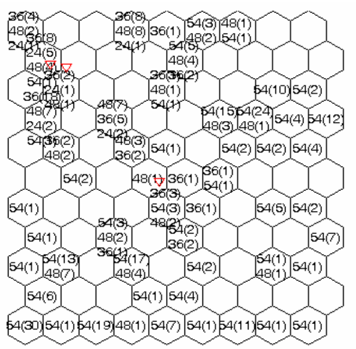

Fig. 4. Labeled SOM when using the FREQ version

At this point, the output of our program is a labeled SOM map (**Fig. 4**) and our program may represent a new data sample on the map but cannot predict its bitrate. In order to train our program how to predict the bitrate of a data sample we transformed our visualization into mathematical functions. According to the ability of SOM technique cells which have the same bitrate are expected to comprise a cluster whose center may be calculated by the equations

$$x = \sum_{i}^{n} w_i x_i / n \qquad (1) \qquad \text{and} \quad y = \sum_{i}^{n} w_i y_i / n \qquad (2),$$

where n is the number of cells which belong to the cluster, x_i and y_i are the coordinates of the cell i and w_i is the weight according to which the cell i participates to the calculation. In the first two versions (VOTE, ADD1) w_i is always set equal to 1 while in the last version w_i is set to be calculated by the following function:

$$w_i = k/r \qquad (3),$$

where k is the number of instances of the specific bitrate in the cell i and r is the sum of the instances of all bitrates of the cell.

In order to define the bitrate of a data sample we need to find to which cluster the BMU of the data sample belongs. The BMU is set to belong to the cluster that its center is closest according to the Euclidean Distance. The bitrate of the data sample will be the one that represents the cluster. Finally, for evaluating our process and reaching conclusions our program is able to compare the predicted values of the bitrate of each data sample with its real measured value.

3 Test Cases and Results

A number of test cases that correspond to variations of input parameters of the proposed method have been set up in order to reach useful conclusions. In particular, the focus is placed on exploring the following: a) which the best choice between the three versions of the method (VOTE, ADD1 and FREQ) is, b) what variables of our data samples are going to be used, c) how many data samples are needed for the training phase and d) what the training algorithm and the values of its parameters should be. For evaluation and comparison reasons, the higher the percent of the correct prediction is, the better the choice. As a result, the metric used was the number of data samples whose bitrate was correctly predicted (expressed in percent). The different test cases are presented and compared to each other below.

3.1 Comparison of the Labelling Versions

Having analyzed the three versions, we need to compare them in order to use the best one according to their results. As mentioned above, the VOTE version uses only the most frequently appeared label when calculating the centres of the clusters. In this case, it is possible that a label doesn't appear in the created SOM even if it has been used as label in a data sample. As a result, labels with fewer instances may not appear in the created SOM. This causes the elimination of one or more clusters as there is no centre of them. In addition to the above, the program terminates a little after the calculation of the centres as according to its programming it needs all four centres. Finally, even if the program wouldn't stop, the data, which are used for evaluation and belong to the eliminated cluster, would be correlated with a wrong label and cluster.

Trying to find a solution to the existing problem of VOTE version, ADD1 version was created. In ADD1 version, all possible labels of each cell participate equally independently of their instances. We executed both versions using same data files in both training and evaluating phases. The result of the tests led to the conclusion that ADD1 version solved the problem of VOTE version but, in cases where VOTE version worked properly, ADD1 version had lower percent of correct predictions.

The above conclusion led us in a new version, the FREQ version. Contrarily to ADD1 version, labels participate in the calculation of the centres of the clusters unequally as a weighted average of their frequency. Having created this version all that was left to be done was its comparison with the VOTE and ADD1 versions. In order to do so, we executed all versions using the same data files in both training and evaluating phase. It is worth mentioning that in all three versions the training parameters were the same. The results of FREQ version were all better (with higher percent of correct predictions of the bitrate) and thus FREQ version is the one that was used in the rest of the experiments.

3.2 Selection of the Variables of a Data Sample

The next step of our research concerned the variables of the data samples that suit better for predicting the bitrate. In order to do so we created many different cases and used the FREQ version of our program and the same training variables. These cases used different data files for both training and evaluation phases. The difference between them lied in the number and the type of the variables of the data samples.

At the created cases there were 9 variables of a data sample that were used in different combinations, namely: noise, RSSI (Received Signal Strength Identifier), number of input and output packets, number of input and output error, number of input and output bytes and bitrate.

Comparing these cases we concluded the following:

- The result does not depend on the fact that data samples are or are not ordered according to the bitrate. The results in both cases are the same.
- The existence or not of bitrate as variable of the data sample does not influence the results always in the same way. In some cases the result was reduced while in other cases it was increased. Moreover, the highest percent of correct predictions was not one of these whose data file contained the bitrate as a variable.
- Finally, the case with the highest percent of correct predictions, equal to 71.4%, was the one whose variables were the number of input and output packets and RSSI. Those variables are the ones that are used in the rest of our paper.

3.3 Selection of the Number of Data Samples

Having selected the variables of a data sample, we needed to decide the number of data samples to participate in the training process of SOM. In order to do so we created cases which included the variables in which we resulted from the analysis in section 3.2 (number of input and output packets and RSSI) but different number of data samples. For taking results we used once more the same training parameters and the FREQ version.

According to the results, the number of data samples affected the results of our predictions but not always in the same direction. These results are depicted in the following **Fig. 5**:

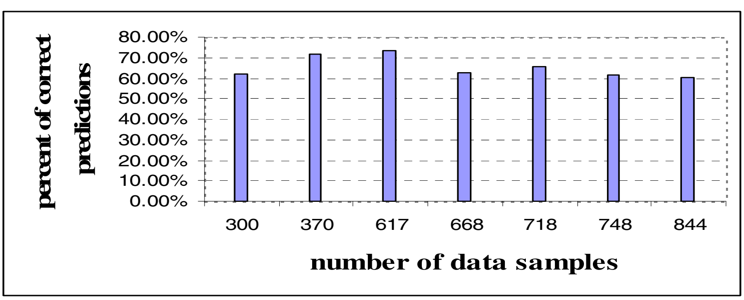

Fig. 5. Diagram of the percent of correct predictions of the bitrate according to the number of the used data samples

Finally, the maximum result was 73.6% and appeared when the number of data samples was 617. This number was also the number of the data samples which comprised our cases for the rest of our tests.

3.4 Selection of the Training Algorithm and its Parameters

Our next concern was to decide between the two training algorithms and finding the most suitable values for their parameters. In order to make such a decision we firstly defined the most suitable values for each training algorithm separately and then we compared them to each other.

In order to decide which were the most suitable values for the parameters of the batch training algorithm we tried different test cases changing only one parameter at a time. The parameters were tested randomly. Comparing the results it was obvious that the best choice in the case of batch training algorithm is shown in **Table 1**:

Table 1. Values of the parameters for the batch training algorithm

Neighborhood function: Gaussian			
Rough Phase		Fine-tuning Phase	
Initial radius	5	Initial radius	1
Final radius	1	Final radius	1
Training length	6	Training length	48

Fig. 6 depicts the diagram of the predicted values of the bitrate, the diagram of the real measured values of the bitrate and a comparison among the two above in case of batch training algorithm.

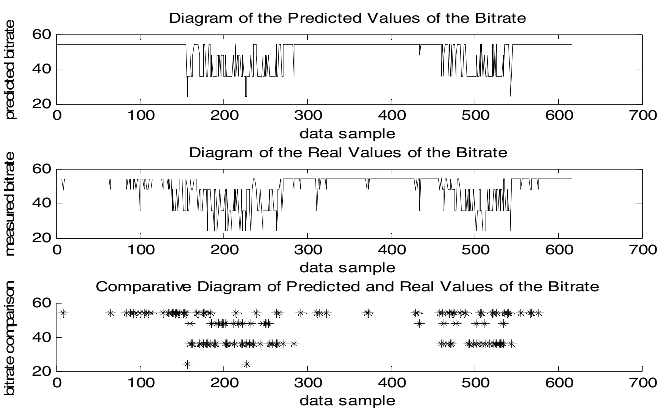

Fig. 6. Batch Training Algorithm: Diagram of the predicted bitrate, Diagram of the measured bitrate and a Comparison of the above. The symbol * depicts only the data samples which have different predicted and real values.

In case of sequential training algorithm, we created different test cases as well. Although the technique was the same it's worth mentioning an important difference: in sequential training algorithm the samples do not enter the training phase at the same time. As a result, the order that they enter the system leads to different results. In order to avoid such a situation we selected the entrance of the samples to be ordered according to the data file.

Finally, the best set of values for the sequential training algorithm is shown on **Table 2**:

Table 2. Values of the parameters for the sequential training algorithm

Neighborhood function: Gaussian			
Length type: epochs			
Learning function: inv			
rough phase		Fine-tuning phase	
Initial radius	3	Initial radius	1
Final radius	1	Final radius	1
Training length	4	Training length	21
Initial alpha	0.5	Initial alpha	0.05

As previously, a diagram of the predicted bitrate, a diagram of the measured bitrate and a comparison of the above diagrams is depicted in **Fig. 7**.

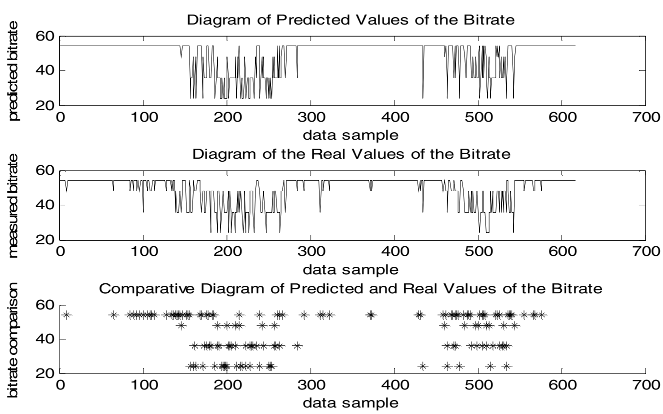

Fig. 7. Sequential Training Algorithm: Diagram of the predicted bitrate, Diagram of the measured bitrate and a Comparison of the above. The symbol * depicts only the data samples which have different predicted and real values.

The comparison of the set of the values of the batch training algorithm with the one of the sequential training algorithm reveals that the first result, equal to 74.4%, is a little lower than the second one, equal to 75.4%, giving the impression that the best choice is the sequential training algorithm. However, the time that is needed to complete the training phase of the SOM is sometimes crucial so we measured it as well. According to the program, batch training algorithm is quicker, requiring about 3 to 4 seconds to complete the training phase while sequential one requires about the double time (7-8 seconds). As a result, and because of the fact that the difference between the two results is rather small, the choice between the two algorithms is subjective and depends on the existence of the requirement of a quick training or not.

4 Conclusions

Rapid evolution of wireless communications demands the use of systems capable of intelligently adapting to the highly varying and disparate modern environments. In these terms, Cognitive Radio Systems have been a very promising technology but the cognition process, which they utilize in order to monitor, evaluate and select a radio configuration to operate with, is often time-consuming, thus leading to the necessity of a learning technique for speeding it up. In this paper we used an unsupervised leaning technique, Self-Organizing Map, in order to train a CRS to predict the bitrate that can be achieved under a combination of parameters obtained as a result of a specific radio configuration and based on its past experience. Going through numerous test cases we achieved to predict correctly the bitrate at 75.4% of the tested data samples. Such a method is expected to assist CRS to choose among the different candidate configurations by taking into account the predictions of the bitrate that can be achieved.

Acknowledgments. This work is performed in the framework of the European-Union funded project OneFIT (Opportunistic NEtworks and cognitive management systems for efficient application provision in the Future InterneT). The project is supported by the European Community's Seventh (7th) Framework Program (FP7). The views expressed in this document do not necessarily represent the views of the complete consortium. The Community is not liable for any use that may be made of the information contained therein.

References

1. Mitola III, J., Maguire Jr., G.: Cognitive radio: making software radios more personal. IEEE Personal Commun. 6(4), 13–18 (1999)
2. Haykin, S.: Cognitive radio: brain-empowered wireless communications. IEEE Journal on Selected Areas in Communications 23(2), 201–220 (2005)
3. Demestichas, P., Katidiotis, A., Tsagkaris, K., Adamopoulou, E., Demestichas, K.: Enhancing Channel Estimation in Cognitive Radio Systems by means of Bayesian Networks. Wireless Personal Communications 49(1), 87–105 (2009)
4. Katidiotis, A., Tsagkaris, K., Demestichas, K.: Performance Evaluation of Artificial Neural Network-Based Learning Schemes for Cognitive Radio Systems. In: Computers & Electrical Engineering. Elsevier, Amsterdam (in press)
5. Tsagkaris, K., Katidiotis, A., Demestichas, P.: Neural Network-based Learning schemes for Cognitive Radio systems. Computer Communications 31(14), 3394–3404 (2008)
6. Liang, M., Shen, J., Wang, G.: Identification of illicit drugs by using SOM neural networks. Journal of Physics D: Applied Physics 41 (2008)
7. Tokutaka, H., Yoshihara, K., Fujimura, K., Obu-Cann, K., Iwamoto, K.: Application of self-organizing maps to chemical analysis. Applied Surface Science, 144–145, 59–63 (1999)

8. Kaski, S., Hankela, T., Lagus, K., Kohonen, T.: WEBSOM – Self-organizing maps of document collections. Neurocomputing 21, 101–117 (1998)
9. Kohonen, T., Somervuo, P.: Self-Organizing Maps of Symbol Strings with Application to Speech Recognition (1997)
10. Matsuura, Y., Tuoya, S.M., Tokutaka, H., Ohkita, M.: The Identification of a Cancer Cell Gene by using SOM. In: 16th International Conference on Genome Informatics, Yokohama Pacifico, Japan, December 19-21 (2005)
11. Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E.S., Golub, T.R.: Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. Proc. Natl. Acad. Sci. USA 96, 2907–2912 (1999)
12. Kohonen, T.: Self-Organizing Maps, 2nd edn. Series in Information Sciences, vol. 30. Springer, Heidelberg (1997)
13. Kohonen, T.: The Self-Organizing map. Neurocomputing 21, 1–6 (1998)
14. Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J.: SOM Toolbox for Matlab 5 (2000)