Enhanced Estimation of Configuration Capabilities in Cognitive Radio

Evgenia Adamopoulou, Konstantinos Demestichas, and Michael Theologou

***Abstract*--Cognitive radio is a highly promising answer to the complexity and heterogeneity characterizing the Beyond 3G wireless scenery. In this context, the present paper advances from the field of interference sensing to the fields of (basic) reasoning and robust reasoning. Interference sensing is concerned with the acquisition of interference related measurements for frequency bands of interest. The paper describes how a cognitive radio system can reason on these measurements, in order to obtain estimations for the capabilities of alternate configurations, especially in terms of achievable transmission capacity and coverage. Subsequently, it focuses on robust reasoning, i.e. on enhancing these estimations by employing machine learning, which constitutes an important aspect of cognitive radio. Several relevant solutions are sketched and explained, with a view to providing a complete picture.**

***Index Terms*--Cognitive radio, Interference sensing, Machine learning, Robust reasoning.**

INTRODUCTION

TODAY'S wireless access landscape comprises various access technologies: (a) 2G, 2.5G and 3G cellular systems (e.g., GSM, GPRS, UMTS); (b) wireless local/ metropolitan/ personal area networks; and (c) broadcast networks, such as Digital Audio Broadcasting (DAB) and Digital Video Broadcasting (DVB). The evolution of the abovementioned wireless communication systems over the past years demonstrates a clear trend towards architectures that will support multiple access technologies, and multimode mobile terminal devices, i.e. capable of alternately operating in the diverse radio segments available in the infrastructure. This trend is often referred to as 'systems Beyond 3G' (B3G), and its main notion is that a network operator can rely on multiple Radio Access Technologies (RATs) for achieving the desired Quality of Service (QoS), i.e. performance (e.g., bit-rate, delay, jitter), availability (blocking probability), reliability (e.g., handover blocking probability), as well as security/safety, in a cost-efficient manner. The need

Manuscript received May 23, 2007.
E. Adamopoulou, K. Demestichas, and M. Theologou are with the School of Electrical and Computer Engineering, National Technical University of Athens, Greece (phone: +30-210-772-1493, contact e-mail: eadam@cn.ntua.gr).

for this stems from the fact that each RAT is best suited for handling certain – but not all – situations, especially in terms of desirable transmission capacity, coverage, mobility support and cost. Thus, in order to be competitive and raise customer satisfaction, a network operator will need to combine the benefits offered by different RATs.

Although significant steps have been made towards the joint utilization of the heterogeneous access technologies, the primary resource that all wireless technologies are built upon, i.e. the electromagnetic radio spectrum, is limited by its nature. Moreover, its fragmentation into separate frequency bands has resulted in the over-utilization of some of them, while others remain largely unoccupied [1]. In all, the underutilization of the radio spectrum is due to the following reasons: (a) only parts of the spectrum can be used by mobile communication systems, while the remaining parts have been reserved for other uses (e.g., air navigation and GPS from 0.96 to 1.2 GHz); (b) efficient wideband operation at 2.5 GHz and beyond is still relatively expensive; and (c) a license is needed to operate in most of the available bands, thus unlicensed users cannot currently utilize the bands for which they do not have legacy rights, even if they do not interfere.

Consequently, the problem that needs to be confronted is the complexity and heterogeneity of the B3G environment, in combination with the scarcity and underutilization of the RF spectrum. *Cognitive radio* [2]-[3] appears as a highly promising solution to this combined problem. Cognitive radio systems are able to sense their RF environment and react, either proactively or reactively, to external stimuli. By the term ‘react’ it is implied that the systems have the ability to reconfigure the algorithms and parameters of their operation, in order to better adapt to environment conditions. Thus, in principle, the operation of a cognitive radio system includes two stages: (a) *observe*, and (b) *decide*. The focus of this paper concentrates on stage (a).

The remainder of this paper is structured as follows: The next section sets the scene by summarizing the main principles of cognitive radio and by examining in more detail its operation scheme. In what follows, the ways in which a cognitive radio may retrieve interference measurements for frequency bands of interest are outlined. In the sequel, we explain in detail how a cognitive radio system can reason on the collected interference measurements, in order to extract estimations for the capabilities of alternate configurations, especially in terms of achievable transmission capacity and coverage range. We then investigate how these estimations can be enhanced by employing machine learning methods. The motivation for this stems from the fact that a cognitive element's environment changes over time, thus it is difficult to estimate the capabilities of alternate configurations with a high degree of assurance.

COGNITIVE RADIO OPERATION

MAIN PRINCIPLES

Cognitive radio is based on three main principles:

(a) Reconfigurability: This property of cognitive radios refers to their ability to dynamically modify their configuration. A *configuration* denotes a combination of RAT and spectrum. Different RATs support different (also, possibly more than one) parameters and algorithms for modulation, coding and error control. Reconfigurability can efficiently be realized through the use of network (and terminal) elements (transceivers) that can dynamically alter the parameters of their operation, in order to improve the offered QoS. Reconfigurations are software-defined, i.e. they are done by activating the appropriate software at the transceiver. As an example of this, let us assume a B3G service area (i.e., a cell) comprising three reconfigurable network elements (transceivers) e_1, e_2 and e_3, each one capable of alternately operating in one of n configurations,

namely c_1, c_2, …, c_n. During a specific time period, the best configuration, in terms of transmission capacity, coverage, mobility support and cost, might be configuration (c_1, c_3, c_2). However, environmental conditions and requirements change dynamically, so in a next time period configuration (c_1, c_4, c_2) might be the most suitable one. In this example, element e_2 was reconfigured to operate under configuration c_4 instead of c_3.

(b) Cognition: It is exactly this stochastic nature of the environment conditions that raises the need for the existence of the second main attribute of cognitive radio systems, i.e. cognition. Cognition refers to the ability of sensing the RF environment and extracting knowledge (data and experience) about the achievable capabilities of alternate configurations.

(c) Self-management: Although not inherent, this attribute can be incorporated into a cognitive radio system, with a view to rendering it more scalable. In this sense, each transceiver should be able of *self*-adapting to its environment, without the need of being instructed by a central management entity with higher rationality. This concept, which is aligned to the *autonomic* computing paradigm, provides significant reduction of a system's complexity, since it does not call for a centralized management entity.

FUNCTIONAL TASKS

A typical cognitive radio operation includes two stages: *observe* and *decide*. The first stage can be divided into two separate functional tasks: *sensing* and *reasoning*. Conclusively, three fundamental functional tasks can be distinguished within a cognitive radio operational cycle: *(i) Sensing*: This involves tuning to a frequency and measuring the interference levels perceived. *(ii) Reasoning*: This encompasses the estimation of the capabilities of a candidate configuration, based on the levels of interference measured in (i). *(iii) Decision-reaching*: Based on the estimations of task (ii), the most suitable configuration is selected, taking under consideration current traffic demands, as well as mobility and positioning patterns. The focus of this paper concentrates on stage (a), i.e. functional tasks (i) and (ii).

SENSING

APPROPRIATE METRICS

Interference is one of the most limiting factors in the operation of cognitive radio. The *sensing* task refers to the procedure of measuring the perceived levels of interference in a certain frequency band, in the area of interest. In order to quantify the effects of interference into signal transmission, two appropriate metrics can be employed: (a) the signal-to-interference-plus-noise ratio (SINR), represented by the ratio between the received wanted carrier signal power, and the total received interference power; and (b) the interference temperature (IT), represented by the ratio between the interference power, and the spectrum bandwidth multiplied by the Boltzmann's constant. Either of the two metrics can be utilized by a cognitive radio system. The following two sub-sections provide an overview on how these two metrics can be computed.

SINR

In general, two strategies can be employed for achieving SINR estimations. The first is based on the transmission of pilot symbols (training sequences), whereas the second tries to derive the channel characteristics directly from the data symbols, i.e. the received signal, without the use of training sequences. The two strategies are often referred to in the literature as 'non-blind' and 'blind', respectively. A training sequence is a priori known to the receiver, thus the task of SINR calculation is made easier, since the receiver knows which symbols it is supposed to receive. Consequently, the use of training sequences allows for greater accuracy, but also introduces a significant overhead, which could be used instead for the transmission of additional data sequences. *In the cognitive radio context*, *the sensing task can be based solely on pilot transmission*, since we are interested in the potential capabilities of alternate configurations. On the other hand, of course, interference estimation in the frequency band that is currently used for service delivery can follow either of the two strategies.

As becomes obvious, the execution of an SINR sensing process requires proper coordination between the network transceiver and the terminals. In particular, this process involves instructing (part of) the local terminals to (a) temporarily switch to a different frequency band, (b) perform SINR measurements, based on pilot signals transmitted by the network transceiver in this band, and (c) report these measurements back to the network side, after first re-switching to the original frequency. The measurements can then be exploited by the network side (the reconfigurable transceiver), in conjunction with the attributes and characteristics of each *candidate RAT*, in order to reach estimations about each RAT's anticipated capabilities (expected performance) in the examined frequency band. Messages of steps (a) and (c) are transmitted using the initial frequency, while the pilot messages of step (b) are transmitted in the alternate frequency band under study. This raises significant synchronization problems, since both the network transceiver and the terminals have to switch frequencies at the same time. However, in the near future, a major part of the research and standardization activities is expected to be targeting at producing the complete specifications of this sensing process.

Interference temperature

IT estimation is more straightforward, compared to SINR estimation, as it only involves tuning to the frequency of interest and calculating the received signal power. By received signal power we mean the received power of the unwanted signals, i.e. the interference, since no desired signal is transmitted yet in this case. The IT sensing process is much simpler (compared to the process for SINR), since measurements can randomly be taken by local terminals, e.g. during periods of inactivity. Terminals can randomly select the frequencies in which they will perform measurements, and then configure back to the original frequency band, in order to report them to the network side. IT estimations are easy to calculate and provide useful information: According to the Federal Communications Commission's model [4], for a given frequency band in a given geographic location, an 'interference temperature limit' is defined by some regulatory agency. A potential transmitter must then ensure that by transmitting it does not raise the current IT above the specified IT limit [5], meaning that the sum of the current IT and the additional IT, caused by the prospective transmission, should not exceed the IT limit.

Reasoning

After receiving the SINR (or IT) measurements from the terminals, the network transceiver's management module is responsible for extracting conclusions about the capabilities of each candidate RAT in the frequency band under study. Due to the dynamic and stochastic nature of the environment conditions, it would be wiser to consider these estimations as *instantaneous estimations* (i.e., not fully reliable), which should be further enhanced by an appropriate mechanism (machine learning technique). In what follows, we concentrate on how to reach instantaneous estimations about the transmission capacity (i.e., maximum achievable throughput) and the coverage range of a candidate configuration (combination of RAT and spectrum), based on the results of the sensing process. As input we assume a set of SINR values. The analysis for the case of IT values remains the same.

Transmission Capacity

Depending on the candidate RAT, the measured SINR values can be mapped to maximum achievable bit-rates (throughputs), taking into account the specific characteristics (e.g., physical modes) of the RAT (Figure 1). This can be clarified by considering the example of the IEEE 802.11b WLAN technology. 802.11b clients can operate at 11 Mbps (using QAM-64), but will scale back to 5.5 (QAM-16), 2 (QPSK), or 1 Mbps (BPSK), if signal quality becomes an issue. Since the lower data rates use less complex and more redundant methods of encoding the data, they are less susceptible to corruption due to interference and signal attenuation.

The network transceiver's management module uses, for each candidate RAT, a reference adaptive scheme, in order to map each input SINR value to a corresponding bit-rate value. For example, in the case of 802.11b systems, the rates 1, 2, 5.5 and 11 Mbps correspond to SINR values of: <4 dB, 4-7.5 dB, 7.5-11 dB, and >11 dB, respectively [6]. Subsequently, the transceiver's management module statistically processes the bit-rate values in order to extract a metric that is referred to as *effective transmission capacity*, as depicted in Figure 1. This metric expresses the anticipated average transmission capacity, with the term "average" implying that the metric has been calculated based on SINR reports from all the terminals involved in the sensing process. This computation may vary from being very simple (a typical average value) to more complicated (like Kalman filtering). The goal is to take into account all measurements, not only the worst or the best cases, in order to gain a more realistic picture concerning the candidate configuration's capabilities in terms of transmission capacity. Back to the 802.11b example, if at a given moment 20% of the users can achieve a maximum bit-rate of 11 Mbps, 0% 5.5 Mbps, 30% 2 Mbps, and 50% 1 Mbps, then, by averaging this input, the effective transmission capacity can be estimated as $0.2 \cdot 11 + 0.3 \cdot 2 + 0.5 \cdot 1 = 3.3$ Mbps.

Coverage

The objective here is similar, i.e. to extract an instantaneous estimation about the *effective* *coverage range* of a candidate RAT in the frequency band under consideration. This metric expresses the anticipated average coverage range. Let it be noted that the effective coverage range does not represent the actual coverage area of the candidate configuration, but is rather a metric (grade) of the configuration's effectiveness in terms of coverage. The proposed process is depicted in Figure 2, and can be divided into two phases: In phase (a), the signal attenuation factor n (i.e., the path loss exponent) in the transceiver's area is calculated. In phase (b), an instantaneous estimation about the effective coverage range is reached.

In phase (a), the following input from each involved terminal is utilized: (i) the power of the desired signal (i.e., the power of the received training sequence, not including the interference power); and (ii) the terminal's location (distance from the transceiver). Concerning (ii), only GPS can provide adequate accuracy in location detection; however, the current trend for mobile terminals is to incorporate GPS receivers. After taking into account the input from all the involved terminals, the average value of the path loss exponent n is determined, as illustrated in Figure 2.

In phase (b), the following input is utilized: (i) the path loss exponent as determined in phase (a); (ii) the minimum required SINR value in order to ensure acceptable quality (this value is RAT-dependent); (iii) the transmission power that has been used while taking the measurements (this value may also be RAT-dependent); and (iv) the average (mean) interference power value, based on the reports of the involved terminals. By processing this input, we can reach an instantaneous estimation about the effective coverage range.

INCORPORATING MACHINE LEARNING

THE NEED FOR ROBUSTNESS

The instantaneous estimations reached through the procedures of the previous section are by definition *instantaneous*. This is because these estimations change frequently and randomly, being influenced by the *varying environment conditions*, including the random and temporary shadowing of terminal units, as well as the configuration selections of "near-by" autonomous network transceivers. As a response to this problem, functional task (ii) (reasoning) needs to be enhanced by a technique that will extract *robust estimations* by post-processing the instantaneous estimations. Such a technique must be capable of making intelligent decisions, rather than blindly considering the recent instantaneous estimations as valid and accurate. In particular, the goal is twofold: (a) To smooth out any fluctuations originating from momentary conditions which do not reflect the actual situation of the RF environment; this can be achieved by taking into account past knowledge (experience). Fluctuations of this kind can be considered as noise that must be avoided. (b) To render the system adaptable to the permanent changes, by properly taking under consideration the recent instantaneous estimations.

Machine learning algorithms are well-suited to evaluation of new, possibly unseen situations based on past knowledge [7]. Their characteristics are well-matched to the challenge of robust reasoning in cognitive radio systems. In the sub-sections that follow, a number of machine learning methods that can be used in the context of extracting robust estimations are described. The objective is to present the complete landscape of machine learning from the viewpoint of cognitive radio's reasoning task. A solution to the problem of extracting robust estimations is sketched, for each machine learning method that is presented. Focus is given on how to model the problem at hand when using each specific method.

In the following, we consider an arbitrary candidate configuration and examine how to reach robust estimations for its effective transmission capacity (the processes are similar in nature for the effective coverage range). We assume that the effective transmission capacity can take values from a set of discrete reference values. To make the examples more illustrative, we assume there are 6 different values in this set (for the configuration under consideration), e.g. (in Mbps) v_1=1, v_2=3, v_3=5, v_4=7, v_5=9 and v_6=11 Mbps. At any given moment, the actual effective transmission capacity may be in the neighbourhood of any of these reference values. To illustrate this, every reference value has an associated *ranking attribute*, which represents the degree of proximity to the actual value. We define the set R as the set reflecting the current rankings of the reference values. For example, if $R=\{2, 1, 3, 4, 5, 6\}$, then reference value v_1 is considered at the moment as the second closest (since r_1=2) to the actual effective transmission capacity, value v_2 is the first closest (r_2=1), value v_3 is the third closest (r_3=3), and so forth. From another viewpoint, the ranking indicates how probable it is for the corresponding reference value to be nearest the actual capacity value.

Commonly, a machine learning method will use a set of training examples as input in order to lean how to behave. Figure 3 depicts a sample set of training examples, in the context of robust reasoning. As illustrated, each training example consists of a set of instance attributes and a set of target attributes. For the problem under consideration, there are two types of instance attributes: (a) the ranking attributes (set R), and (b) the most recent instantaneous estimate of the effective transmission capacity (denoted for convenience as e). Attribute e may take any of the transmission capacity reference values (i.e., v_1 through v_6). The pair $<R, e>$ constitutes the set of instance attributes. The current ranking of the reference values (R), together with the most recent instantaneous estimation of the effective transmission capacity (e), must be jointly taken under consideration in order to reach a target classification, i.e. a target ranking T (possibly different from the current one) of the reference values. Thus, set T constitutes the set of target attributes; e.g., if t_1=2, then reference value v_1 should now be considered as the second closest to the actual effective transmission capacity.

Decision Tree Learning

Decision tree learning is one of the most widely used and practical methods for inductive inference. It serves as a method for approximating discrete-valued functions, while being robust to noisy training data [7]. The learned function is represented by a decision tree. New instances are classified by being sorted down the decision tree, from the root to some leaf node. Each leaf node provides a potential classification of an instance. Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute in the given example.

Figure 4 illustrates a sample of a typical learned decision tree. This decision tree takes as input a new instance $<R, e>$ and suggests a classification of target attribute t_1 into one of its possible discrete values (i.e., 1 through 6). Similar trees can be formed, as well, for the other target attributes (t_2, t_3, etc.). By passing the new instance through all the decision trees, the set T of target attributes is formed. Inconsistencies (i.e., two target attributes being set to the same value) are handled by making some random choice. A different approach would be to use a single decision tree (instead of 6 different trees), the leaf nodes of which would correspond to possible values of set T. However, this would require at least 6! = 720 leaf nodes to be present in the tree, in order to cover all possible values of set T. The proper training of this tree would require a prohibitively large number of training examples.

The key question is *how* to construct a decision tree that will be able to successfully classify future instances, based on a limited set of training examples. Most algorithms that have been developed for learning decision trees are variations on a core algorithm, namely the ID3 algorithm [8], which employs a top-down, greedy search through the space of possible decision trees. The ID3 algorithm begins with the question "which attribute should be tested (hence, placed) at the root of the tree?". To answer this question, each attribute is evaluated using a statistical test to determine how well it *alone* classifies the training examples. The best attribute is selected and used as the test at the root node of the tree. A descendant of the root node is then created for each possible value of this attribute, and the training examples are sorted to the appropriate descendant node. The entire process is then repeated using the training examples associated with each descendant node to select the best attribute to test at that point in the tree. In each step, the attribute that is most useful for classifying examples must be selected. The measure that is typically used to guide this selection is the *information gain*, a statistical property that reflects how well a given attribute separates the training examples according to their target classification.

Instead of using a static, initial set of training examples, another solution would be to follow a more dynamic approach. The idea is to regard each new classified instance as a training guideline, and retrain the system, i.e. reconstruct the decision trees. As an example, assume there is an initial set of 10 training examples that is used in order to produce an initial set of 6 decision trees, as specified in the previous. Let us now assume that a new instance is classified, with the help of the 6 decision trees. Once classified, this instance may be used as a new training example, leading to the construction of 6 new decision trees. At this point, there are two approaches: (a) either to infinitely enlarge the training set, or (b) to replace an old training example with the new one. Case (b) is computationally more efficient and regards recent instances as more useful than the past ones, by employing a "sliding window" of training examples. Although such a solution seems to compromise accuracy, it constitutes a valid approach, since it takes into account the time dependency of the sensing process, which leads to (a) the incorporation of additional training examples as time passes, and (b) the preference of recent instances over past ones.

Artificial Neural Networks

Artificial neural networks (ANNs) provide a general, practical method for learning either real-valued or discrete-valued functions from examples [7]. Figure 5 illustrates a sample of a typical ANN. Every ANN consists of three or more layers of units (or nodes): an input layer, one or more hidden layers and an output layer. Each node produces a single output. The lines entering a node represent its inputs. For the problem at hand, the following design can be followed:

Input encoding. The following input units are required: (a) the value of the attribute e (i.e., the most recent instantaneous estimation), and (b) the values of the attributes of set R (i.e., the rankings of the reference values). Hence, 7 input units are required in total.

Output encoding. The output layer should consist of 6 units, one for every target attribute of set T. The type of units that will be used is an important design choice. One solution is the *sigmoid unit*, which takes as input a vector of real-valued inputs, calculates a linear combination of these inputs, then computes a non-linear scaling (threshold function) ranging from 0 to 1. By using this range of values, the 6 possible values of a target attribute should be encoded as depicted in Figure 5. A useful property of the sigmoid unit is that its non-linear threshold function is easily differentiable. Learning a sigmoid unit involves choosing values for the weights of the inputs, in such a way that the training examples are successfully classified by the ANN.

Network graph structure. It is common to use either one or two hidden layers of sigmoid units in typical ANNs. In the current design, a single hidden layer may be employed. Another key issue is how many hidden units to include in the ANN. For the problem at hand, an appropriate choice would be to select at most 6 different hidden units, i.e. at most as many hidden units as capacity reference values. Of course, more hidden units can be used, which may, in general, lead to somewhat higher test accuracy. However, sufficient test accuracy can also be achieved by employing only a small number of hidden units. Another advantage is that smaller numbers of hidden units require smaller training times.

Learning algorithm. The algorithm that is typically used for learning ANNs is the Backpropagation algorithm [9], which is able to learn the weights for a multilayer network, given that the latter possesses a fixed set of units and interconnections. Backpropagation receives as input a set of training examples and attempts to minimize the squared error between the network output values and the target values for these outputs. This is done by slowly adjusting the weights towards the direction that reduces the aforementioned squared error.

Adding recurrence. Up to this point, an important limitation of the constructed ANN is that the prediction of T at time $t+1$ depends only on input R and e at time t and cannot capture possible dependencies of $T(t+1)$ on earlier values of R and e. Of course, this difficulty could be remedied by including not only $R(t)$ and $e(t)$, but also $R(t-1)$ and $e(t-1)$, at the input layer of the ANN. However, if we wish the network to consider an arbitrary window of time in the past when predicting $T(t+1)$, then a different solution is required. The recurrent network of Figure 5 provides one such solution. Here, a new unit b has been added to the hidden layer, as well as a new input unit *c(t)*. The value of *c(t)* is defined as the value of unit *b* at time *t-1*; i.e., the input value *c(t)* to the network at one time step is simply copied from the value of unit *b* on the previous time step. Notice that this implements a recurrence relation, in which *b* represents information about the history of network inputs. Because *b* depends not only on *R(t)* and *e(t)* but also on *c(t)*, it is possible for *b* to summarize information from earlier values of *R* and *e* that are arbitrarily distant in time. Interestingly, recurrent networks such as this one can be trained by using a simple variant of Backpropagation.

Bayesian Reasoning

Bayesian reasoning provides a probabilistic approach to inference. It is based on the assumption that the quantities of interest are governed by probability distributions and that optimal decisions can be made by reasoning about these probabilities together with observed data [7]. The most recent approach to Bayesian reasoning is the development of *Bayesian belief networks*. A Bayesian belief network, or simply Bayesian network (BN), describes the probability distribution governing a set of variables, by specifying a set of conditional independence assumptions along with a set of conditional probabilities. Associated with each node of a BN is a conditional probability table (CPT), which specifies the conditional distribution for the corresponding variable given its immediate parents in the graph.

Since the problem at hand is time-dependent, a simple and computationally efficient strategy can be employed: The idea is to keep the structure of the BN relatively simple, and move the complexity towards the learning algorithm. To that end, the simple model of Figure 6 can be utilized, and the focus may be given on how to update the CPTs, i.e. how to make them reflect the actual current conditions. Learning BNs (i.e., adapting the values of the CPTs) is in the spotlight of much current research [7]. For the specific problem, a process such as the one presented in Figure 6 can be followed. Whenever a new instantaneous estimation e becomes available, this process: (a) measures the instantaneous estimation's distance from each reference value v_i, and (b) utilizes this distance, in conjunction with the current probability distribution of each target attribute t_i, in order to calculate the new (adapted) probability distribution of t_i. By following this process, the probability values are adapted in such a way that captures the influence of both past and recent knowledge.

CONCLUSIONS

Despite being a relatively recent concept, cognitive radio is developing extremely rapidly and is considered today as a highly promising way towards handling the complexity of the B3G wireless landscape. Cognitive systems dynamically reconfigure the algorithms and parameters they use, in order to adapt to the changing environment conditions. The present paper extensively investigated the problem of extracting estimations about the capabilities of candidate configurations, especially in terms of achievable transmission capacity and coverage range. It described in detail the processes that a real-life cognitive radio system may follow, in order to reach some estimates of these capabilities. It then focused on the problem of enhancing these estimations, by incorporating machine learning. Relevant solutions were sketched and explained in a tutorial manner. In the future, more research attention is expected to be attracted by machine learning in the context of cognitive radio, since cognition and machine learning are tightly interconnected. This paper will hopefully serve as a guideline for such future works.

Current work conducted by the authors is focused on extracting and studying results from a BN-based machine learning technique that has been implemented on the basis of the specifications presented in the *Bayesian Reasoning* sub-section. Current results show that, by following this method, the system is able to avoid overestimating, as well as underestimating, a candidate configuration's achievable effective transmission capacity, by performing adaptations that are based upon not only recent but also past knowledge.

Future researchers may be interested in implementing some of the machine learning algorithms described in this paper. In this case, they must keep in mind that they should be able to evaluate the performance of their machine learning methods in the following ways: (a) how *successfully* the method handles noisy input data, i.e. instantaneous estimations that do not truly reflect the situation of the RF environment; (b) how *fast* the method adapts to new, not temporary situations. In the context of cognitive radio, the presented machine learning algorithms can be applied to any problem related to the determination of the capabilities of candidate configurations, including, but not limited to, the determination of the effective transmission capacity, the effective coverage range, the achievable delay and jitter, the blocking probability, etc. Future research can also be directed towards proposing modifications and enhancements of the presented machine learning algorithms, e.g. by incorporating more input parameters, as well as towards developing machine learning models targeted for the third functional task of cognitive ratio operation (*decision-reaching*), i.e. targeted for reaching optimal reconfiguration decisions based on the extracted *robust* capability estimations, as well as other parameters, such as cost, time restrictions, and policies.

ACKNOWLEDGEMENT

This work has been performed in the framework of the ARIADNE project (03ED235), partially funded by the General Secretariat of Research and Technology (GSRT) of the Greek Ministry of Development.

REFERENCES

[1] S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications", *IEEE JSAC*, Vol. 23, No. 2, Feb. 2005, pp. 201-220.

[2] J. Mitola and G. Maguire Jr., "Cognitive Radio: Making Software Radios More Personal", *IEEE Personal Communications Magazine*, Vol. 6, No. 6, Aug. 1999, pp. 13-18.

[3] J. Mitola, "Cognitive radio: An integrated agent architecture for software defined radio", Doctor of Technology, Royal Institute of Technology (KTH), Stockholm, Sweden, 2000.

[4] Federal Communications Commission, "Establishment of interference temperature metric to quantify and manage interference and to expand available unlicensed operation in certain fixed mobile and satellite frequency bands", *ET Docket 03-289*, Notice of Inquiry and Proposed Rulemaking.

[5] P. J. Kolodzy, "Interference temperature: a metric for dynamic spectrum utilization", *International Journal of Network Management, Special Issue: Management of Interference in Wireless Networks, John Wiley & Sons*, Vol. 16, No. 2, March 2006, pp. 103-113.

[6] J. Del Prado and S. Choi, "Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement", ICC'03, Anchorage, Alaska, May 2003, pp. 1108-1113.

[7] T. Mitchell, "Machine Learning", *McGraw Hill*, ISBN 0-07-042807-7, 1997.

[8] J. R. Quinlan, "Induction of decision trees", *Machine Learning*, Vol. 1, No. 1, 1986, pp. 81-106.

[9] R. Williams and D. Zipser, "Gradient-based learning algorithms for recurrent networks and their computational complexity", *In Y. Chauvin & D. Rumelhart (Eds.), Backpropagation: Theory, architectures, and applications. Hillsdale, NJ: Lawrence Erlbaum Associates*, 1995, pp. 433-486.

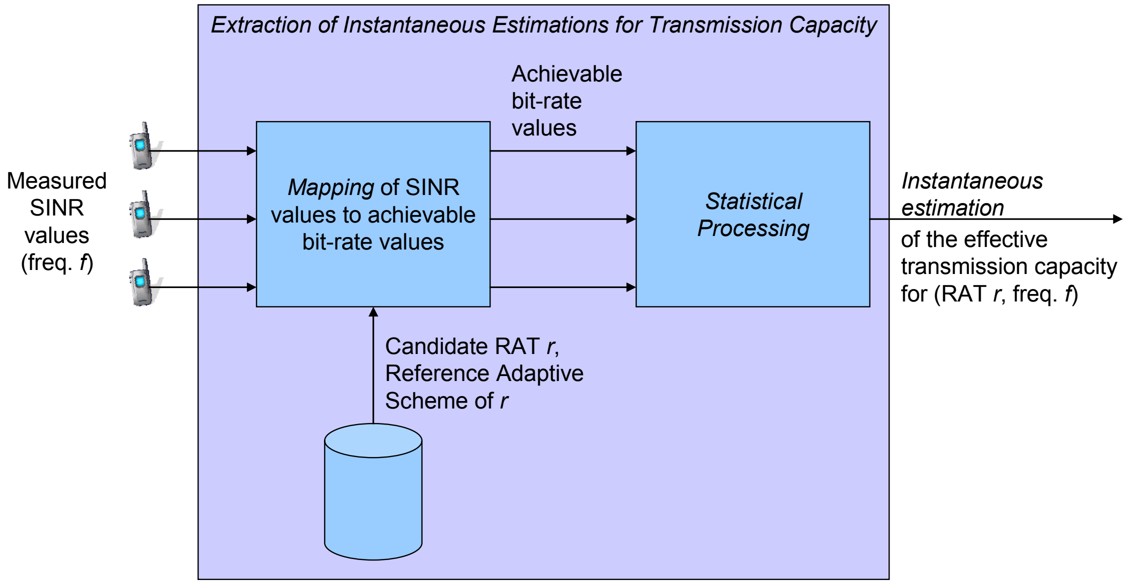


Figure 1: Process of extracting instantaneous estimations for the effective transmission capacity of a candidate configuration

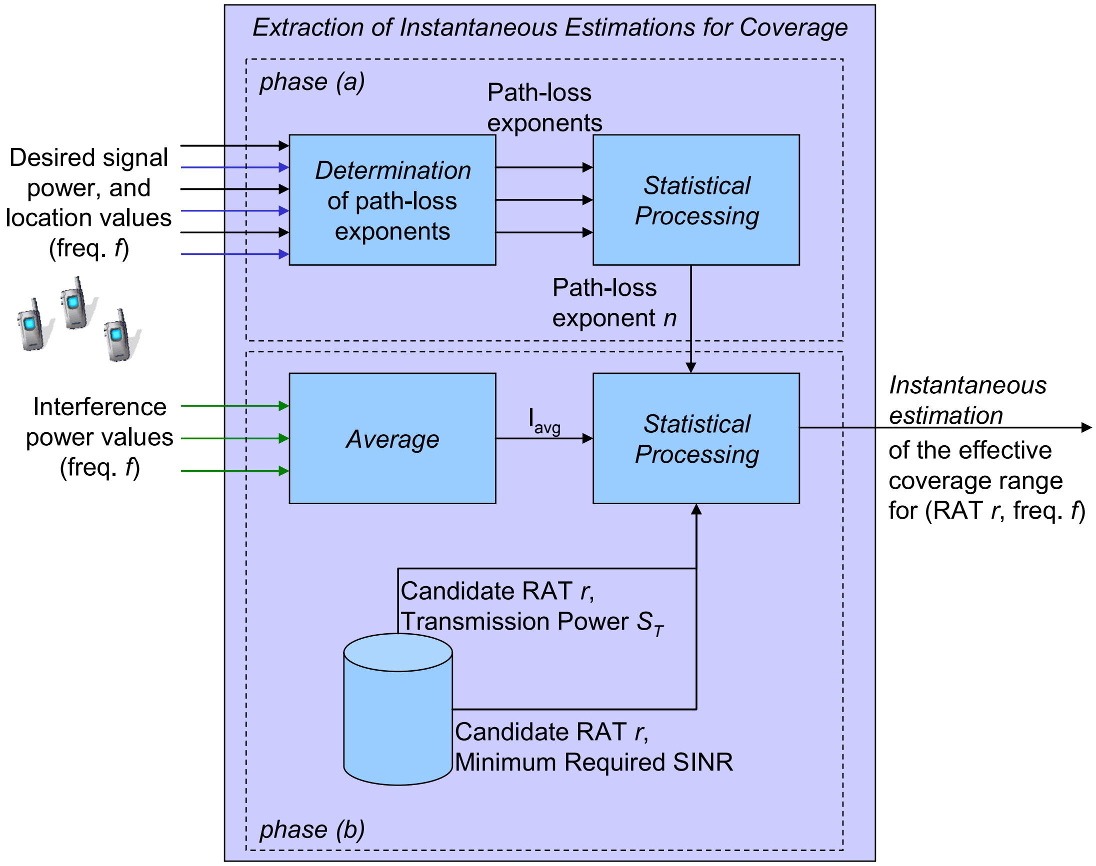
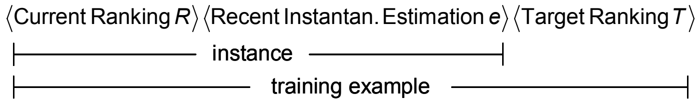


Figure 2: Process of extracting instantaneous estimations for the effective coverage range of a candidate configuration



R						***e***	***T***					
2	1	3	4	5	6	v_1	1	2	3	4	5	6
3	1	2	5	4	6	v_3	3	2	1	4	5	6
6	5	4	2	1	3	v_5	6	5	4	2	1	3
6	4	5	3	1	2	v_6	6	4	5	3	2	1
…	…	…	…	…	…	…	…	…	…	…	…	…

Figure 3: Sample set of training examples

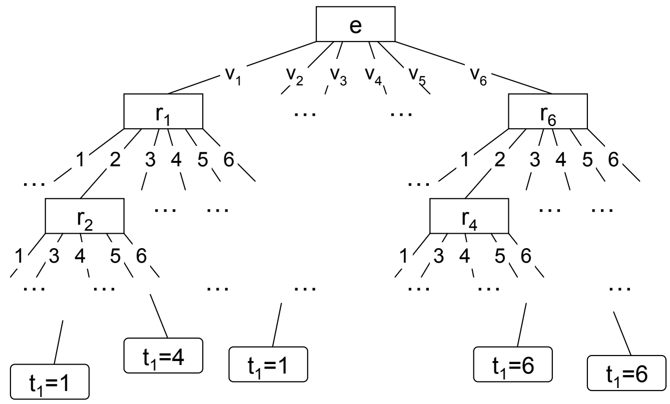


Figure 4: Sample of a typical learned decision tree

Output encoding of the ANN

Rank	1	2	3	4	5	6
Output Encoding	0.143	0.286	0.429	0.571	0.714	0.857

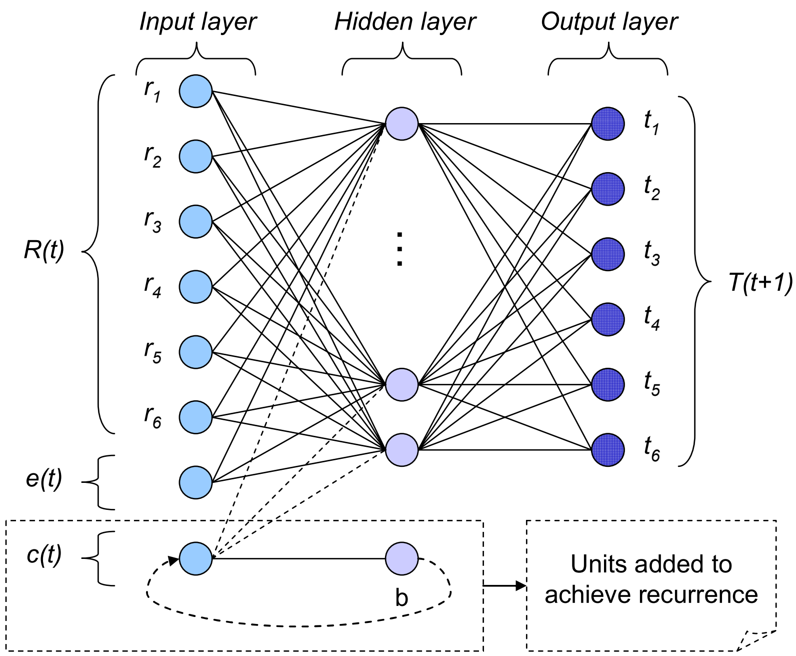
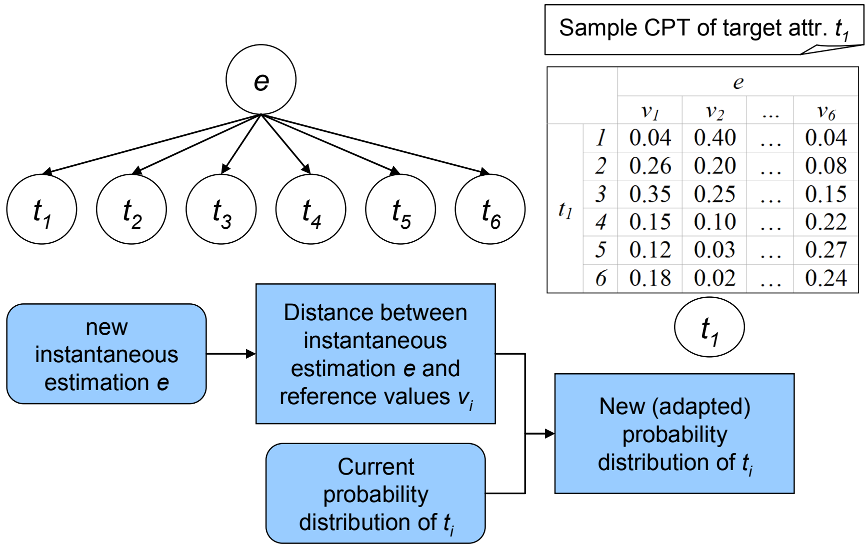


Figure 5: Output encoding and structure of a sample ANN; Units of the hidden and output layer are sigmoid units.



		e			
		v_1	v_2	…	v_6
t_1	1	0.04	0.40	…	0.04
	2	0.26	0.20	…	0.08
	3	0.35	0.25	…	0.15
	4	0.15	0.10	…	0.22
	5	0.12	0.03	…	0.27
	6	0.18	0.02	…	0.24

Figure 6: Structure of the BN for robust reasoning, and process for learning the BN