A Survey of Machine Learning Algorithms and Their Applications in Cognitive Radio

Mustafa Alshawaqfeh[1], Xu Wang[1], Ali Rıza Ekti[1(✉)], Muhammad Zeeshan Shakir[2], Khalid Qaraqe[2], and Erchin Serpedin[1]

[1] Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA
{mustafa.shawaqfeh,xu.wang,arekti}@tamu.edu, serpedin@ece.tamu.edu
[2] Department of Electrical and Computer Engineering, Texas A&M University at Qatar, Doha, Qatar
{muhammad.shakir,khalid.qaraqe}@qatar.tamu.edu

Abstract. Cognitive radio (CR) technology is a promising candidate for next generation intelligent wireless networks. The cognitive engine plays the role of the brain for the CR and the learning engine is its core. In order to fully exploit the features of CRs, the learning engine should be improved. Therefore, in this study, we discuss several machine learning algorithms and their applications for CRs in terms of spectrum sensing, modulation classification and power allocation.

Keywords: Cognitive radio · Machine learning · Learning engine · Spectrum sensing · Modulation classification

1 Introduction

The evolution of wireless communications systems and many other devices is continuously subject to two major development trends: a) improvement of existing capabilities, and b) extension and insertion of new features into the existing structures. In what concerns the first trend, one can notice that insertion of new features arises from the fact wireless systems progress very fast in accordance with the market demands. Therefore, wireless systems always require new services and applications. One of the most striking examples for such situations is cell phones. Earlier cell phones were used only for voice transmissions along with limited text messaging applications however contemporary cell phones are capable of transmitting multimedia along with an operating system running on. In what concerns the second trend, a continuous improvement of existing capabilities is a necessity since incorporating new features adds new dimensions that help improve the existing capabilities.

The above mentioned considerations suggest that adaptation and optimization should always be employed as key enabling technologies for the continuous update of communication systems to dynamically changing conditions.

M. Weichold et al. (Eds.): CROWNCOM 2015, LNICST 156, pp. 790–801, 2015.
DOI: 10.1007/978-3-319-24540-9_66

In this regard, the purpose of this study is to provide a conceptual description of machine learning algorithms used in the design of wireless communication systems in the light of a recently emerging technology called cognitive radio (CR) [1–5]. The idea of CR was first presented by Joseph Mitola III. and Gerald Q. Maguire, Jr. in [3] "The point in which wireless personal digital assistants and the related networks are sufficiently and computationally intelligent about radio resources and related computer-to-computer communication to detect user communications needs as a function of use context, and to provide radio resources and wireless services most appropriate to those needs" [1]. There are many advantages offered by CRs in wireless communications. A CR is basically an intelligent wireless device which is aware of the environment and spectrum and is able to adapt/optimize itself easily to the characteristics of the communication channel to satisfy the user needs. The environment of a CR may include radio frequency (RF) spectrum, user behavior, transmission characteristics and parameters, multi-access interference, localization and data rates of users. The key strengths of machine learning algorithms are their adaptive nature with respect to the dynamic changes of the channel and communication system parameters. In addition, the ability to work without prior knowledge about the communication environment represents another important feature of CRs. These considerations recommend machine learning as a promising technology for CRs.

In this paper, applications of machine learning for *learning engine*, *spectrum sensing*, *modulation classification* and *power allocation* in CRs are studied along with currently available methods and approaches to better adapt and optimize the overall system performance. The rest of the paper is organized as follows. The learning engine is presented in Section 2. An overview of key machine learning techniques that can be implemented into the learning engine is presented in Section 3. A review of machine learning applications in spectrum sensing, modulation classification and power allocation for CRs are presented in Section 4, Section 5 and Section 6, respectively. Concluding remarks are provided in Section 7.

2 Learning Engine

The cognitive engine is the brain of a CR system and it enables the system to react intelligently to changes in the environment. Basically, as shown in Figure 1, the CR extends a software-defined radio by adding an independent cognitive engine, which consists of a learning engine and reasoning engine [6]. The learning engine lies in the core of the cognitive engine and it aims to build a model or an objective function based on the inputs that are to be used in taking the right decisions and making the correct predictions.

In the context of CRs, no simple relationship between the system inputs and the objective function is available due to the high complexity and degree of freedom of the software-defined radio (SDR). In this case, several channel statistics, such as transmit power, modulation scheme and sensing scheme, need to be adjusted simultaneously [7]. In such scenarios, adopting a policy-based

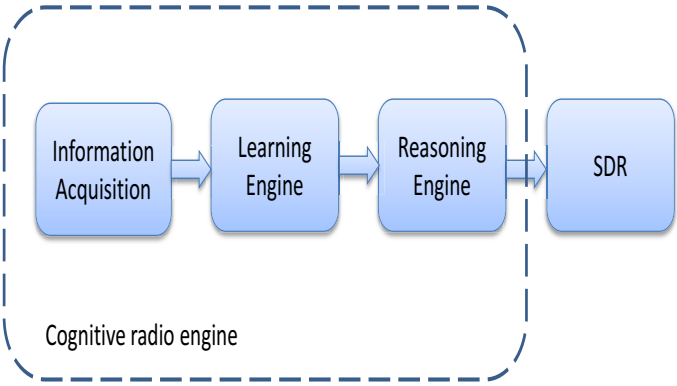

Fig. 1. The structure of cognitive radio engine.

decision making strategy is infeasible due to the large number of states that the cognitive radio networks (CRNs) and its radio frequency (RF) environment assume. In addition, even if the resources are available, considering all the possible states and actions is impossible given the dynamic and random nature of CRNs. Thus, the learning engine is crucial in the operation of the CR engine. A learning engine is adopted to estimate the channel statistics. The results are incorporated into a predictive calculus-based reasoning engine to make decisions and achieve certain objectives.

Several learning algorithms can be used to implement the learning engine. For the sake of brevity, Table 1 lists some of recent works involving the applications of machine learning algorithms in CR. The recent literature shows extensive use of different learning algorithms in CRs which will be discussed in Section 3. However, several factors influence the selection of the learning algorithm to implement the learning engine. For example, one important factor is the availability of prior knowledge about the environment. Supervised learning methods are applicable only if prior information about the environment is known to train the agent. On the other hand, unsupervised learning methods are appealing for scenarios with lack of prior information. The computational complexity of the algorithm is the main limiting factor especially for CRs with limited resources. In general, CRNs and their RF environment exhibit the following characteristics [7]: (*i*) incomplete observation information about the state variable, (*ii*) incorporation of CRs into CRNs and (*iii*) unknown RF environment. Consequently, the learning engine must be designed by taking into account the above characteristics such that the learning method efficiently and optimally adapt to the changes and the incompleteness of the observed information and RF environment.

Table 1. Classification of Papers Exploiting Machine Learning Algorithms

	Supervised Learning		Unsupervised Learning	Reinforcement Learning
	SVM	KNN		
Spectrum Sensing	[8,9]	[8,9]	[8,10]	[11–13]
Modulation Classification	[14,15]	[16]		
Power Allocation				[17–19]

3 Machine Learning

In literature, machine learning techniques can be categorized into three different types, namely, supervised learning, unsupervised learning and reinforcement learning (RL).

3.1 Supervised Learning

Supervised learning is a machine learning approach that infers an objective function from a labeled training data. Thus, this method requires prior information about the environment. The training data consists of input-output pairs. An inferred function is derived based on the samples to map the future input. For instance, the training samples (x_i, y_i) are given and it is assumed that (x_i, y_i) are drawn from some distribution $P(\mathbf{x})$. Classification is the main function for supervised learning and its goal is to find a classifier function f such that it fits and characterizes the training examples. The classifier is used to map and classify the newcoming data. One well-known example of supervised learning methods is referred to as the support vector machine (SVM) and it was first developed in [20]. The original SVM approach builds a linear classifier that maps the input vectors to a high-dimensional space. A nonlinear SVM classification method was proposed by Boser et al [21] using the kernel trick. SVM is exploited in a wide range of machine learning applications due to its accurate predictions, fast evaluation of the targeted function and the robustness against noise and errors. For more information about SVM, the reader is referred to [22,23].

3.2 Unsupervised Learning

In contrast to the supervised learning, the unsupervised learning applies to an environment in which the prior knowledge is unknown. Specifically, the unsupervised learning extracts hidden features from the unlabeled data. Since the samples from unsupervised learning are unlabeled, unsupervised learning receives neither targeted outputs nor environmental rewards. This fact distinguishes the unsupervised learning from the supervised learning and the reinforcement learning. The main functions for unsupervised learning are clustering, dimensionality reduction and blind signal separation [24,25]. In principle, a clustering algorithm aims to group objects into clusters such that the elements in the same cluster are similar to each other and different from the elements placed in any other clusters. There are several clustering algorithms such as K-means or centroid-based clustering [26,27] and mixture models.

3.3 Reinforcement Learning

Reinforcement learning is an online learning method which lies in the middle between supervised and unsupervised learning. The general idea behind the reinforcement learning is to maximize a specific reward function. According to [28], the reinforcement learning consists of three main components: a *policy*, a *reward function* and *value function*. Let S be the set of all possible states of the environment, and A be the set of all possible actions and n denote the time index. A policy $\pi : S \times A \rightarrow S$ is the rule that defines the selection of next state s_{n+1} based on the current state-action pair (s_n, a_n). The policy can be deterministic or stochastic. In a deterministic policy, the agent selects the actions in a deterministic fashion based on the current state. The reward function $r_n : A \times S \rightarrow \Re$ is a scalar function that maps each state-action pair (s_n, a_n) into a single real number, *reward*, that indicates the reward obtained by selecting the action a_n at state s_n to move into state s_{n+1}. According to the knowledge of the reward function, reinforcement learning is classified into a model-based learning if the reward is known and a model-free learning otherwise. Generally, the reward functions may be stochastic. The reward function determines the immediate or short term reward of an action. However, the agent is interested in the long-run total reward which is defined by the *value function* or *return*. Starting from state s_n, the *return* is the random variable R_n defined as:

$$R_n = \begin{cases} \sum_{k=0}^{\infty} \gamma^k r_{n+k+1} : \text{non-episodic model} \\ \sum_{k=0}^{N} r_{n+k+1} \quad : \text{episodic model,} \end{cases} \tag{1}$$

where $\gamma \in [0, 1]$ is the *discount factor*. The goal of the reinforcement algorithm is to find a policy that maximizes R_n. In principle, the optimal policy can be found by exhaustive search of the policy space. This solution is computationally infeasible due to the large (or even infinite) number of policies to be checked. Hence, the core of reinforcement learning algorithms is to find an efficient method to calculate or approximate the function value.

One appealing method is to estimate the function value. Estimation of function values in more details is commonly carried out within a Markov Decision Process (MDP), which represents a general framework for reinforcement learning. MDP is a reinforcement learning environment in which states satisfy Markov property. Markov property means that deciding the next state s_{n+1} depends only on the current state s_n and action a_n. In other words, the current state and actions contain all the required information about future state. Mathematically, this condition can be expressed as follows:

$$\begin{aligned} &Pr\{s_{n+1}, r_{n+1} | s_n, a_n, r_n, s_{n-1}, \ldots, s_0, a_0, r_0\} \\ &\quad = Pr\{s_{n+1} = s, r_{n+1} = r | s_n, a_n, r_n\}. \end{aligned} \tag{2}$$

The Markovian assumption simplifies the analysis by allowing prediction of future rewards based only on the current state and action. A finite MDP means that state and action spaces are finite. A natural way to estimate the value function is to take the sample mean of the received rewards. Since the rewards depend

on the selected action, the estimated value function depends on the selected policy. Define the *state-value function for* π *policy* (V^{π}) as the expected value of return given that agent is in the s_n state and follows the π policy. For MDP, $V^{\pi}(s_n)$ is defined as:

$$V^{\pi}(s_n) = E_{\pi}[R_n|s_n, \pi]. \tag{3}$$

Similarly, the *action-value function for* π *policy*, $Q^{\pi}(s_n, a_n)$, is defined as the expected return starting from state s_n and taking the action a_n and following the policy π. In MDP, $Q^{\pi}(s_n, a_n)$ can be defined as:

$$Q^{\pi}(s_n, a_n) = E_{\pi}[R_n|s_n, a_n, \pi]. \tag{4}$$

It is shown in [28] that the optimal action-value function $Q^*(s_n, a_n)$ satisfies:

$$Q^*(s_n, a_n) = \max_{\pi} Q^{\pi}(s_n, a_n) \cdot \sum_{s_{n+1} \in S} \Big(Pr[s_{n+1}|s_n, s_n] \; [r_n + \gamma \max_{a_{n+1} \in A} Q^*(s_{n+1}, a_{n+1})] \Big). \tag{5}$$

One way to maximize the action-value functions is the Q-learning algorithm [29]. Q-learning follows a fixed state transition and does not require prior information about the environment. The update for the one-step version is given by:

$$Q_{n+1}(s_n, a_n) = Q_n(s_n, a_n) + \alpha[r_{n+1} + \gamma \max_{a_{n+1} \in A} Q_n(s_{n+1}, a_{n+1}) - Q_n(s_n, a_n)]. \tag{6}$$

The reinforcement learning is subject to a trade-off between exploration and exploitation. This trade-off manifests through the fact that at each stage, the agent has to decide whether to exploit the current highest reward action or to explore new actions for higher rewards. Two action selection methods for controlling the trade off between exploration and exploitation are the ϵ-*greedy* and *softmax* action [28,30]. In ϵ-greedy, the next action is selected either at random with uniform probability ϵ or by selecting the optimal action $a^* = \max_a Q(a, s)$ with probability $1 - \epsilon$. In the softmax method, the action a is selected with probability

$$\frac{exp\{Q(s_n, a_n)/\tau\}}{\sum_{a_{n+1} \in A} exp\{Q(s_n, a_n)/\tau\}}, \tag{7}$$

where τ is a positive weight factor for each action and is referred to as the temperature factor.

Reinforcement learning algorithms differ by how they efficiently compute the value function. Reinforcement learning algorithms can be also divided into *single agent reinforcement learning* (SARL) and *multiple agent reinforcement learning* (MARL). In SARL, the learning process is local at each agent in the sense that rewards for each agent does not depend on the other agents. In MARL, the reward depends on both, the environment and all agent policies and actions. This dependence on other agents' policies complicates the learning process. The interested reader is referred to [28,31,32] for detailed information about the reinforcement learning.

4 Machine Learning for Spectrum Sensing

The main challenge of CRNs is to opportunistically utilize the unused spectrum of the primary system. Also, the CR should be designed in a way to protect the primary users from any interference or quality of service (QoS) degradation. To achieve this goal, the CR must present the ability to detect the occupancy of RF transmission activities in the primary system.

Various methods have been proposed for spectrum sensing [33] such as *matched filter*, *energy detection* and *cyclostationary detection*. The matched filter [34] is known to be optimal for detecting deterministic unknown signals in additive white Gaussian noise (AWGN). However, the matched filter approach is a coherent method, and impractical for scenarios where the CR compete for large number of bands. Implementation of a matched filter for such scenarios requires to equip the CR device with a large number of synchronization circuits to match the different bands. However, such an approach is not efficient. The basic idea of energy detector [35,36] is to measure the energy of the received signal, and then to compare it to a threshold to decide the occupancy of the sensed primary band. The main advantages of an energy detector are its simplicity, low cost and the ability to work without any prior knowledge about the waveform of the primary system. However, an energy detector is very sensitive to channel impairments since it is unable to distinguish between the primary signal and noise or any type of interference. Cyclostationary detection is based on the fact that many digital and analog modulated signals have special statistical features because of the inherent periodicity of these signals statistics [37]. In contrast, the noise does not present in general such features. One way to exploit the cyclostationary features is to use the spectral-correlation density (SCD) function. A cyclostationary statistics based detection approach is more immune to stationary noise and interferences. Moreover, cyclostationary provides inherent signal identification since different signals differ in their SCD function. However, these benefits come at the cost of more complexity.

Assessing the RF-spectrum is a high dimensional complex problem due to the large number of parameters involved. Using dynamic programming methods is computationally infeasible especially if the CR devices present power limitations. Machine learning provides an asymptotically close-to-optimal and computationally efficient alternative [30]. Therefore, many papers propose machine learning-based techniques for spectrum sensing.

Spectrum sensing is a typical classification (or clustering) problem in the sense that it is required to identify whether the sensed band belongs to the available class (or cluster). A misdetection occurs if the selected channel is considered to be idle, while it is in reality used by the primary system. Hence, the primary system will be subject to an interference and a collision may occur. On the other hand, a false alarm occurs if the channel is available for the CR but the classifier decides that it is used. Consequently, a degradation of spectrum utilization occurs.

The authors in [8] implemented cooperative spectrum sensing (CSS) using several machine learning techniques. These techniques are the K-means clus-

tering and Gaussian mixture model (GMM) from the unsupervised learning category and the support vector machine (SVM) and the weighted K-nearest-neighbor (KNN) from the supervised learning category. The CSS considered here is a centralized based cooperative sensing. All the energy levels estimated at the CR devices are collected at a CR device (e.g., the central device). This vector of energy levels acts as a feature input for the classification and clustering algorithm to decide whether the channel is available for the CR or not. The channel is idle if it is not utilized by any primary user. Similarly, the CSS scenario is studied in [9] using SVM and KNN.

The authors in [38] proposed a centralized CSS method in which each CR reports its measurements to a central node (another CR device). Then, the linear fusion rule is used to decide the availability of the channel. To enhance the sensing performance, the topology of the CRN is taken into account because measurements carried out by CRs closer to the primary users are more reliable than far away transceivers. The impact of the location information is reflected into the values of the linear coefficients which are determined by the Fisher linear discriminant analysis.

In many cases, the spectrum of interest is very wide and/or non-contiguous. Hence, a single CR device may not sense the whole the spectrum at once. An alternative solution is to assign a subset of k CRs to sense each subband [39]. One issue with the fixed number assignment is that monitoring some subbands with k CRs is more than what is needed to achieve the sensing requirements. And hence, more power consumption is required for the CRs. In [11], a reinforcement learning method with ϵ-greedy action selection is employed to optimize the multiband spectrum sensing and reduce the energy consumption in the CRN. This is achieved by exploiting the occupancy statistic of each subband and then assigning the minimum number of CRs that achieves the required misdetection probability.

5 Machine Learning for Modulation Classification

In general, modulation classification algorithms assume two steps. The first step performs the feature extraction. Examples of features are spectral correlation and cumulants. The second step carries out the classification task (via Naive Bayes, SVM) or clustering task (via KNN, mixture models).

The authors of [16] proposed a two stage classification algorithm using Genetic Programming (GP) and K-Nearest Neighbor (KNN) approach. The proposed algorithm can identify BPSK, QPSK, 16QAM and 64QAM modulation schemes, and exploit the forth and sixth order cumulants of the received signals as features. The first stage divides the signal into three classes: BPSK, QPSK and QAM (both 16 and 64). To differentiate between 16QAM and 64QAM, the third class output is fed into the second stage classifier that distinguishes between 16QAM and 64 QAM.

In [14], two modulation classification approaches are presented. Both of them exploit the SVM classifier. However, they differ in the selection of the feature

vector and modulation schemes. The first approach aims to distinguish among 16 QAM, 32 QAM and 64 QAM and uses the demodulation error (i.e, the distance between the received symbol and its nearest neighbor in each constellation) as a feature vector. The second approach aims to distinguish among AM, BPSK, QPSK and BFSK and uses the cyclic spectral correlation as a feature.

The previous works classify only digital modulated signals. In [15], a SVM classification method is proposed to classify two analog modulated signals (AM and FM) in addition to five digitally modulated ones (BPSK, QPSK, GMSK, 16-QAM and 64QAM). The authors use a combination of spectral and higher order cumulants as features. Then, these features are fed into a SVM classifier to identify the modulation scheme.

6 Machine Learning for Power Allocation

As mentioned in Section 3, Q-learning is a simple and efficient way to implement reinforcement learning. The operation of Q-learning requires the definition of a reward function. In power allocation problem, defining the reward function can be easily done in terms of the transmission powers and channel gains. The authors in [17] use centralized Q-learning to address the channel and power allocation problem in CRNs. They consider a scenario where all the transmissions of the CRs are controlled by a cognitive base station. Therefore, the cognitive base station is the learning agent and provides channel and power allocation services to the CRs. In this work, the number of transmission activities of the CRs is modeled as a Poisson process. The state is defined as

$$\begin{aligned}\mathbf{s}_n =&[\text{incoming user index, user(s) on transmission,}\\ &\text{received power on each channel}]^T\end{aligned}$$

and the reward function is defined by:

$$r_n = \sum_{i=1}^{N} \log_2 \left(1 + \frac{P_i f(i) h_i(f(i))}{N_0 + \sum_{j \neq i} P_j f(j) h_j(f(j)) \psi(i,j)}\right)$$

where $f(i)$ and P_i are the channel and power level used by the i'th user, respectively. N_0 denotes the noise power and N stands for the number of users. Function $\psi(i,j)$ is determined by:

$$\phi(i,j) = \begin{cases} 1 & , f(i) = f(j) \\ 0 & , else. \end{cases}$$

A decentralized Q-learning algorithm for power allocation is considered in [18]. The reward criterion is defined by:

$$r_n = \sum_{i=1}^{N} \left(SINR_i^s - SINR_{Th}^s\right)^2,$$

where $SINR_i^p$ is the primary network $SINR$ at the I_i'th cell, $SINR_i^s$ denotes the secondary network $SINR$ at the I_i'th cell and N stands for the number of cells.

In [19], a decentralized MARL is considered to control the transmit power and spectrum used by CRs in order to reduce the interference at the primary users. In order to overcome the increased computational complexity of the function value in reinforcement learning for large CRNs, the authors apply an approximation to the value function using a Kanerva-based approximation function. In this paper, the environment state at time index n is defined as $\mathbf{s}_n = [\mathbf{sp}_n, \mathbf{pw}_n]^T$ where $\mathbf{sp}_n$ denotes the vector of spectra and $\mathbf{pw}_n$ stands for a vector of power values across all agents.

7 Conclusion

There is a growing interest in machine learning techniques in assessing the features of CRNs. Therefore, in this study, we investigated the usefulness of machine learning techniques for spectrum sensing, modulation classification and power allocation in CRNs.

Acknowledgments. This publication was made possible by NPRP grant 4-1293-2-513 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

1. Mitola, J.: Cognitive Radio–An Integrated Agent Architecture for Software Defined Radio. Royal Institute of Technology (KTH) (2000)
2. Chen, K.-C., Prasad, R.: Cognitive Radio Networks. John Wiley & Sons, June 2009
3. Mitola, J., Maguire, G.: Cognitive Radio: Making software radios more personal. IEEE Personal Communs. **6**(4), 13–18 (1999)
4. Wang, J., Ghosh, M., Challapali, K.: Emerging cognitive radio applications: A survey. IEEE Communs. Magazine **49**(3), 74–81 (2011)
5. Wang, B., Liu, K.R.: Advances in cognitive radio networks: A survey. IEEE J. Selected Topics Signal Process **5**(1), 5–23 (2011)
6. Clancy, C., Hecker, J., Stuntebeck, E., O'Shea, T.: Applications of machine learning to cognitive radio networks. IEEE Wireless Communs. **14**(4), 47–52 (2007)
7. Bkassiny, M., Li, Y., Jayaweera, S.K.: A survey on machine-learning techniques in cognitive radios. IEEE Commun. Surveys & Tuts. **15**(3), 1136–1159 (2013)
8. Thilina, K.M., Choi, K.W., Saquib, N., Hossain, E.: Machine learning techniques for cooperative spectrum sensing in cognitive radio networks. IEEE Journal on Selected Areas in Communications **31**(11), 2209–2221 (2013)
9. Thilina, K.M., Choi, K.W., Saquib, N., Hossain, E.: Pattern classification techniques for cooperative spectrum sensing in cognitive radio networks: SVM and W-KNN approaches. In: 2012 IEEE Global Communications Conference (GLOBECOM), pp. 1260–1265 (2012)

10. Ding, G., Wu, Q., Song, F., Wang, J.: Decentralized sensor selection for cooperative spectrum sensing based on unsupervised learning. In: 2012 IEEE International Conference on Communications (ICC), pp. 1576–1580, June 2012
11. Oksanen, J., Lundén, J., Koivunen, V.: Reinforcement learning method for energy efficient cooperative multiband spectrum sensing. In: 2010 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), pp. 59–64 (2010)
12. Lo, B.F., Akyildiz, I.F.: Reinforcement learning-based cooperative sensing in cognitive radio ad hoc networks. In: 2010 IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), pp. 2244–2249 (2010)
13. Di Felice, M., Chowdhury, K.R., Kassler, A., Bononi, L.: Adaptive sensing scheduling and spectrum selection in cognitive wireless mesh networks. In: 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN), pp. 1–6 (2011)
14. Freitas, L.C., Cardoso, C., Muller, F.C., Costa, J.W., Klautau, A.: Automatic modulation classification for cognitive radio systems: results for the symbol and waveform domains. In: IEEE Latin-American Conference on Communications, LATINCOM 2009, pp. 1–6 (2009)
15. Petrova, M., Mähönen, P., Osuna, A.: Multi-class classification of analog and digital signals in cognitive radios using support vector machines. In: 7th International Symposium on Wireless Communication Systems (ISWCS), pp. 986–990 (2010)
16. Aslam, M.W., Zhu, Z., Nandi, A.K.: Automatic modulation classification using combination of genetic programming and KNN. IEEE Transactions on Wireless Communications **11**(8), 2742–2750 (2012)
17. Yao, Y., Feng, Z.: Centralized channel and power allocation for cognitive radio networks: a Q-learning solution. In: IEEE Future Network and Mobile Summit, pp. 1–8
18. van den Biggelaar, O., Dricot, J., De Doncker, P., Horlin, F.: Power allocation in cognitive radio networks using distributed machine learning. In: IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), pp. 826–831 (2012)
19. Wu, C., Chowdhury, K., Di Felice, M., Meleis, W.: Spectrum management of cognitive radio using multi-agent reinforcement learning. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Industry track, pp. 1705–1712. International Foundation for Autonomous Agents and Multiagent Systems (2010)
20. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
21. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proc. ACM Fifth Annual Workshop Computational Learning Theory, Pittsburgh, PA, USA, July, 1992, pp. 144–152 (1992)
22. Byun, H., Lee, S.-W.: Applications of support vector machines for pattern recognition: a survey. In: Lee, S.-W., Verri, A. (eds.) SVM 2002. LNCS, vol. 2388, pp. 213–236. Springer, Heidelberg (2002)
23. Wang, G.: Applications of support vector machines for pattern recognition: a survey. In: Proc. IEEE Fourth International Conference on Networked Computing and Advanced Information Management, pp. 123–128 (2008)
24. Hastie, T., Tibshirani, R., Friedman, J., Franklin, J.: The elements of statistical learning: Data mining, inference and prediction. The Mathematical Intelligencer **27**(2), 83–85 (2005)
25. Qiu, R.C., Hu, Z., Li, H., Wicks, M.C.: Cognitive Radio Communication and Networking: Principles and Practice. John Wiley & Sons (2012)

26. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Applied Statistics, 100–108 (1979)
27. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(7), 881–892 (2002)
28. Sutton, R.S., Barto, A.G.: Introduction to Reinforcement Learning. MIT Press (1998)
29. Watkins, C.J., Dayan, P.: Q-learning. Machine Learning **8**(3–4), 279–292 (1992)
30. Biglieri, E.: Principles of Cognitive Radio. Cambridge University Press (2012)
31. Busoniu, L., Babuska, R., De Schutter, B.: A comprehensive survey of multiagent reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews **38**(2), 156–172 (2008)
32. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. Journal of Artificial Intelligence Research, 237–285 (1996)
33. Zeng, Y., Liang, Y.-C., Hoang, A.T., Zhang, R.: A review on spectrum sensing for cognitive radio: challenges and solutions. EURASIP Journal on Advances in Signal Processing (2010)
34. Cabric, D., Mishra, S.M., Brodersen, R.W.: Implementation issues in spectrum sensing for cognitive radios. In: Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 772–776. IEEE (2004)
35. Sonnenschein, A., Fishman, P.M.: Radiometric detection of spread-spectrum signals in noise of uncertain power. IEEE Transactions on Aerospace and Electronic Systems **28**(3), 654–660 (1992)
36. Tandra, R., Sahai, A.: Fundamental limits on detection in low snr under noise uncertainty. In: 2005 International Conference on Wireless Networks, Communications and Mobile Computing, vol. 1, pp. 464–469. IEEE (2005)
37. Kim, K., Akbar, I., Bae, K., Um, J.-S., Spooner, C., Reed, J.: Cyclostationary approaches to signal detection and classification in cognitive radio. In: 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2007, pp. 212–215 (2007)
38. Choi, K.W., Hossain, E., Kim, D.I.: Cooperative spectrum sensing under a random geometric primary user network model. IEEE Transactions on Wireless Communications **10**(6), 1932–1944 (2011)
39. Oksanen, J., Koivunen, V., Lundén, J., Huttunen, A.: Diversity-based spectrum sensing policy for detecting primary signals over multiple frequency bands. In: 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 3130–3133 (2010)