PSO-Least Squares SVM for Clustering in Cognitive Radio Sensor Networks

Jerzy Martyna

PSO-Least Squares SVM for Clustering in Cognitive Radio Sensor Networks

Jerzy Martyna

Institute of Computer Science, Faculty of Mathematics and Computer Science
Jagiellonian University, ul. Prof. S. Lojasiewicza 6, 30-348 Cracow, Poland

Abstract. In this paper, a solution for a cluster formation in cognitive radio networks is presented. The solution features a network-wide energy consumption model for these networks. The particle swarm optimisation (PSO) and least squares support vector machines (LS-SVMs) have been transformed into our clustering problem. The obtained results show that the given hybrid AI system provides a good estimate of a cluster formation. Through extensive simulations, we observed that the PSO-LS-SVM method can be effectively used under various spectrum characteristics. Moreover, the formed clusters are reliable and stable in a dynamic frequency environment.

Key words: least squares support vector machines, particle swarm optimisation, cognitive radio technology, wireless sensor networks, clustering problem

1   Introduction

Cognitive radio (CR) technology belongs to a key technology that enables the improvement of the spectrum use in a dynamic manner. The Federal Communications Commission (FCC) reported that a large portion of the assigned spectrum is used sporadically, leading to a significant amount of the unused spectrum [9].

CR devices allow one to sense the environment (cognitive capability), analyse and study the detected information, and adapt to the environment (reconfigurable capabilities). Moreover, CR technology can be integrated into wireless sensor networks [2]. In this integration, nodes can communicate by tuning their radios to vacant radio channels through a collision-free band instead of through the heavily crowded unlicensed bands. The cognitive radio sensor networks [1, 16] (CRSN) allows us to implement multimedia applications with high bandwidth requirements. Moreover, the CRSN gathers all of the sensed data of observed phenomena.

A two-tiered CRSN is composed of a two-level hierarchy. The first level consists of ordinary sensors according to the application-specific objectives. All ordinary sensors are grouped into clusters. All cluster head sensor nodes that are responsible for data aggregation and sending it to the base station (BS) belong to the second level. It is obvious that each clustering technique used in CRSN

must reduce the number of exchanged messages between the ordinary sensors and the cluster head nodes, and minimise the distance between the cluster head nodes to reduce transmission power.

Many solutions for the realisation of clustering in the CRSN have been proposed in the recent literature. Among others, in the paper by Younis et al. [17], an election clustering method based on nodes residual energy as a primary clustering parameter and node proximity toits neighbours as secondary parameter was proposed. Unfortunately, it was designed for fixed channel settings and can not be used in the CRSN. A deterministic clustering scheme was proposed in the paper by Bradonji and Lazos [3]. In this approach, clusters are realised by the maximising the summation of idle channels with the number of nodes within the cluster. According to the authors' statement, this solution is stable, but is a NP-hard and violates design constraints for energy and computation. A specific aware clustering method has been presented in the paper by Zhang et al. [18]. The proposed method demands introducing the k-mean clustering algorithm. In this method each node is treated as cluster head node and then merges cluster heads within each iteration until the number of cluster heads reaches an optimal number that was obtained a priori by the theoretical analysis. Applying artificial intelligence (AI) can be an alternative approach to solving the clustering problem in the CRSN. Cognitive radio sensor networks, that deal with the intelligent assignment and use of the radio spectrum, are the natural study research area for the use of AI techniques.

Support vector machines (SVMs) have been developed as an alternative that avoids artificial neural network (ANN) limitations [14], [15]. SVM compute globally optimal solutions, unlike tend to fall into local minima [4]. Least squares support vector machines (LS-SVMs), introduced by Suykens et al. [11], [12], simplify the training process of standard SVM in a great extent by replacing the inequality constraints with equality ones. A particle swarm optimisation (PSO) [5] belongs to stochastic optimisation technique influenced by the social behaviour of bird flocking or fish schooling. The use of a PSO makes it possible to optimise the performance of the classifier LS-SVM method for clustering in the CRSN networks. Use of the PSO-LS-SVM methods permit to build a novel bybrid artificial intelligence (AI) system.

The main goal of this paper is to introduce a new hybrid AI system based on PSO-LSSVM techniques for clustering in CRSNs. In the proposed method, the structure of the CRSN is mapped into the network energy consumption model. Thus, a spectrum-aware clustering process is efficiently realised. We show that suggested method achieves good performance efficiency: the clusters are organized in such a way that the total communication power is minimised.

This paper is organised as follows. In Section 2, we present an energy-efficient spectrum aware consumption model for CRSNs. Section 3 introduces the hybrid AI system based on the PSO and LS-SVM methods. In Section 4, we evaluate the effectiveness of the proposed method through intensive simulations. Section 5 concludes the paper.

2   Model of the network energy consumption

In this section, we present the network energy consumption model. We take into account data transmission, data signalling and data sensing in the CRSN.

The data transmission involves intracluster and intercluster communications. In the first case, all CRSN nodes belonging to the cluster send their data to the cluster heads [18]. Thus, the sum of transmission power of all cluster members is given by:

$$P_{intra} = \sum_{k=1}^{K} \sum_{i=1, i \neq j}^{N_k} P_i(n_i^k) \tag{1}$$

where N_k is the number of CRSN nodes in the k-th cluster, $1 \leq k \leq K$. P_i is the power needed to reading data transmission from the CRSN node to the cluster head node. The distance between cluster members and their cluster head are short within the cluster and, assuming a free-space channel model as an appropriate channel model, we can rewrite the sum transmission power as:

$$P_{intra} = C_0 P_r \sum_{k=1}^{K} \sum_{i=1, i \neq j}^{N_k} d^2(n_i^k, n_j^k) \tag{2}$$

where P_r is the minimal received power required for the cluster head to correctly decode the transmitted information, C_0 is constant loss factor, and $d(n_i^k, n_j^k)$ is the Euclidean distance between the i-th and j-th nodes in the k-th cluster.

As the cluster head node, we can select the number of nodes with maximal residual energy. As proven by Zhang [19], intracluster energy achieves the upper bound if each node will have an equal probability of becoming cluster head. Hence, the sum for intracluster power consumption is given by:

$$P_{intra} = C_0 P_r \sum_{k=1}^{K} \sum_{j=1}^{N_k} \frac{1}{N_k} \sum_{i=1, i \neq j}^{N_k} d^2(n_i^k, n_j^k) =$$

$$2C_0 P_r \sum_{k=1}^{K} \sum_{i=1}^{N_k} d^2(n_i^k, \text{center}(k)) \tag{3}$$

where $\text{center}(k) = (\frac{1}{N_k} \sum_{i=1}^{N_k} x_i^k, \frac{1}{N_k} \sum_{i=1}^{N_k} y_i^k)$ is the center of the k-th cluster. The total network wide energy consumption is formulated as follows:

$$E_{network} = P_{sen} t_{sen} + P_{sig} t_{sig} + (K C_0 P_r d_{max}^2 +$$

$$2C_0 P_r \sum_{k=1}^{K} \sum_{j=1}^{N_k} d^2(n_i^k, \text{center}(k))) t_{data} \tag{4}$$

where P_{sen}, P_{sig} is the energy devoted to sensing and signalling, t_{sen}, t_{sig}, t_{data} are the sensing, signalling, and data transmission times, respectively, d_{max} is the transmission range of the CRSN node.

3 Hybrid AI System Based on the PSO and LS-SVM Methods for Clustering in CRSN System

3.1 Particle Swarm Optimization

The particle swarm optimization (PSO) is a swarm intelligence method that models social behaviour of organisms such as bird flocking and fish schooling [5] to guide swarms of particles toward the most promising regions of the search space. PSO is conceptually similar to the crossover operation used by genetic algorithms. The main difference of particle swarm optimization method from the evolutionary computing is that flying potential through hyperspace are accelerating toward "better" solutions, while in evolutionary computation schemes operate on potential solutions which are represented as locations in hyperspace [6].

Each particle represents a candidate position (i.e., solution). A particle is considered as a point in a D-dimension space, and its status is characterized according to its position and velocity. Thus, the D-dimensional for the particle i at the t-th iteration can be represented as $x_i^t = \{x_{i1}^t, x_{i2}^t, \ldots, x_{iD}^t\}$. The velocity (i.e., distance change) for particle i at iteration t can be defined by $v_i^t = \{v_{i1}^t, v_{i2}^t, \ldots, v_{iD}^t\}$. Let $p_i^t = \{p_{i1}^t, p_{i2}^t, \ldots, p_{iD}^t\}$ be the best solution that particle i has obtained until iteration t and $p_g^t = \{p_{g1}^t, p_{g2}^t, \ldots, p_{gD}^t\}$ be the best solution obtained from p_i^t in the population at iteration t. To search for the optimal solution, each particle changes its velocity according to the cognition and social parts as follows:

$$V_{id}^t = V_{id}^{t-1} + c_1 r_1 (P_{id}^t - x_{id}^t) + c_2 r_2 (P_{gd}^t - x_{id}^t), \quad d = 1, 2, \ldots D \qquad (5)$$

where c_i indicates the cognition learning factor; c_2 indicates the social learning factor, and r_1, r_2 are random numbers uniformly distributed in $U(0, 1)$. Each particcle moves to a new potential solution based on the equation

$$X_{id}^{t+1} = X_{id}^t + V_{id}^t, \quad d = 1, 2, \ldots, D \qquad (6)$$

The iteration is terminated if the number of iteration achieves the given of number of maximum iteration.

3.2 LS-SVM Classifiers

Least squares support vector machines (LS-SVM) [11], [12] are least squares versions of support vector machines (SVM) [14], [15], which are set of related supervised learning methods that analyse data and recognize patterns, and which are used for classification and regression analysis.

Let $\{(x_i, y_i)\}_{i=1}^{N}$ be a given training set, with the input data $x_i \in R^n$ and output data $y_i \in R$ with class labels $y_i \in \{-1, +1\}$ and the linear classifier

$$y(x) = sign[w^T x + b] \tag{7}$$

When the data of the two classes are separable we have the original SVM classifier [14], [15] that satisfies the following conditions:

$$\begin{cases} w^T \phi(x_i) + b \geq +1 \text{ if } y_i = 1 \\ w^T \phi(x_i) + b \leq -1 \text{ if } y_i = -1 \end{cases} \tag{8}$$

These two sets of inequalities can be combined into one single set as follows

$$y_i[w^T \phi(x_i) + b] - 1 \geq 0, \quad i = 1, 2. \ldots, N \tag{9}$$

where $\phi \colon R^n \to R^m$ is the feature mapping the input space to a usually high dimensional feature space. The data points are linearly separable by a hyperplane defined by the pair $(w \in R^m, b \in R)$. Thus, the classification function is given by

$$f(x) = sign\{w^T \phi(x) + b\} \tag{10}$$

Instead of estimating with the help of the feature map we work with a kernel function in the original space given by

$$K(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j) \tag{11}$$

In order to allow for the violation of Eq.(9), we introduce slack variables ξ_i such that

$$y_i[w^T \phi(x_i) + b] \geq 1 - \xi_i, \quad \xi_i > 0, \quad i = 1, 2, \ldots, N \tag{12}$$

The following minimization problem is accounted for as follows:

$$\min_{w,b,\xi} J(w, b, \xi) = \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} \xi_i$$
$$\text{subject to} \quad y_i[w^T \phi(x_i) + b] \geq 1 - \xi_i, \quad \xi_i > 0,$$
$$i = 1, 2, \ldots, N, \quad C > 0 \tag{13}$$

where C is a positive constant parameter used to control the tradeoff between the training error and the margin.

The dual problem of the system (13), obtained as a result of Karush-Kuhn-Tucker (KKT) condition [7], leads to a well-known convex quadratic programming (QP). The solution of the QP problem is slow for large vectors and it is difficult to implement in the on-line adaptive form. Therefore, a modified version of the SVM called the Least Squares SVM (LS-SVM) was proposed by Suykens et al [11], [13].

In the LS-SVM method, the following minimization problem is formulated

$$\min_{w,b,e} J(w,b,e) = \frac{1}{2}w^T w + \gamma \frac{1}{2}\sum_{k=1}^{W} e_k^2$$
$$\text{subject to} \quad y_k[w^T \phi(x_k) + b] = 1 - e_k, \ k = 1, 2, \ldots, N \quad (14)$$

The corresponding Lagrangian for Eq. (13) is given by

$$L(w,b,e;\alpha) = J(w,b,e) - \sum_{k=1}^{N} \alpha_k\{y_k[w^T \phi(x_k) + b] - 1 + e_k\} \quad (15)$$

where the α_k are the Langrange multipliers. The optimality condition leads to the following $(N+1) \times (N+1)$ linear system

$$\begin{bmatrix} 0 & Y^T \\ Y & \Omega^* + \gamma^{-1}I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix} \quad (16)$$

where
$Z = [\phi(x_1)^T y_1, \ldots, \phi(x_N)^T y_N]$
$Y = [y_1, \ldots, y_N]$
$\mathbf{1} = [1, \ldots, 1]$
$\alpha = [\alpha_1, \ldots, \alpha_N]$
and $\Omega^* = ZZ^T$. Due to the application of Mercer's condition [8] there exists a mapping and an expansion

$$\Omega_{kl}^* = y_k y_l \phi(x_k)^T \phi(x_l) = y_k y_l K(x_k, x_l) \quad (17)$$

Thus, the LS-SVM model for the function estimation is given

$$y(x) = \sum_{k=1}^{N} \alpha_k y_k \cdot K(x, x_k) + b \quad (18)$$

where parameters α_k and b are based on the solution to Eqs. (15) and (16).

3.3   Mixtures of kernels

Each kernel function is characterized by its advantages and disadvantages. For instance, a radial basis function (RBF) kernel $K(x; x_i) = exp\{- \mid x - x_i \mid^2 / \sigma^2\}$, where σ is the width of the radial basis function, is a typical local kernel in which only the data that are close have an influence on the kernel values. The polynomial kernel (see [14]) such as $K(x; x_i) = [x \cdot x_i + 1]^q$, where q is the kernel parameter which defines the degree of the polynomial to be used, guarantees the influence of all the data points that are far away from each other. Therefore, the mixture of these kernels gives a better performance than any single kernel.

As was defined by Smits et al. [10], an exemplary mixture of the RBF and polynomial kernels is given by

$$K_{mix} = \rho \cdot K_{poly} + (1 - \rho)K_{RBF} \tag{19}$$

where ρ is the mixing coefficient treated as a constant scalar.

3.4   LS-SVM transformed into a clustering problem in CSRN

The input data are described by a coordinate (r, z) and the output data is the energy value. For any data block, the input points are in the form $\{(r_o + dr, z_o + dz) : \mid dr \mid < m, \mid dz \mid < n\}$. All such sets of points can be translated to the same set $\{(dr, dz) : \mid dr \mid < m, \mid dz \mid < n\}$ by subtracting (r_o, z_o) from all the vectors, m and n are the half numbers of the horizontal and vertical pixels of the blocks. The LS-SVM method can be transformed into a clustering problem by the use of the same set of input vectors but different sets of labels.

We can transform the Eq. (16) into a system

$$\begin{bmatrix} 0 & 1^T \\ 1 & \Omega \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ Y \end{bmatrix} \tag{20}$$

where $Y = [y_1; \ldots,; y_N]$, $\Omega = K(x_i, y_i) + \gamma^{-1}I$, $1^T = [1_1; \ldots; 1_N]$, $\alpha = [\alpha_1; \ldots; \alpha_N]$. Thus, the Ω is given by

$$\Omega = K(x_i, x_j) + \gamma^{-1}I \tag{21}$$

The solution of Eq. (21) gives the values

$$\begin{cases} b = \frac{1^T \Omega^{-1} Y}{1^T \Omega^{-1} 1} \\ \alpha = \Omega^{-1}(Y - b1) \end{cases} \tag{22}$$

By setting $A = \Omega^{-1}$ and $B = \frac{1^T \Omega^{-1}}{1^T \Omega^{-1} 1}$ we obtain

$$\begin{cases} b = BY \\ \alpha = A(Y - b1) \end{cases} \tag{23}$$

where A and B are precalculated matrixes that depend only on the input vector (x_k) but not on the vector y_k.

The sensors are usually correlated due to the high probability that the adjacent pixels will contain the sensors in the cluster. We assume that the sensor field is two-dimensional and the image energy distribution of the sensor field on the surface, known as the point spread funcion (PSF), can be approximated by the Gaussian PSF. On the other hand, the center point of the PSF corresponds to the measured sensor position.

In our approach based on the mixtures of kernels, we take into consideration two sets of indexes of the neighbourhood of the sensor image that satisfies the condition of linearly separable patterns. We recall that the linear separability

requires that in order to be classified, the patterns must be separated from each other to ensure that decision surfaces should consist of hyperplans.

The LS-SVM with the RBF and polynomial kernels transformed into a sensor clustering problem has the fitted energy intensity surface function over the constant vector space as follows

$$g(r,z) = \sum_{k=1}^{N} \alpha_k \{ (\rho[(rr_k + zz_k) + 1]^q + (1-\rho) \cdot e^{-(|r-r_k|^2 + |z-z_k|^2)/\sigma^2} \} + b \quad (24)$$

where (r,z) are the coordinates of the pixels. The function $g(r,z)$ gives the corresponding energy intensity value, b and α are obtained as a solution of Eq. (23).

3.5   A multi-class formulation of the LS-SVM transformed into a sensor clustering problem

In comparison with the standard SVM method, the LS-SVM has a lower computational complexity and memory requirements. Nevertheless, in certain situations, such as the classification of several characters, clusters, etc., a multi-class classifiction is very suitable.

In the multi-class formalism we now use multiple output values y^i with $i = 1, \ldots, n_y$, where n_y defines the number of output values [15]. Thus, in the primal weight space the multi-class classification system possesses the following binary classifiers

$$\begin{cases} y^{(1)}(x) & = sign[w^{(1)^T} \phi^{(1)}(x) + b^{(1)}] \\ y^{(2)}(x) & = sign[w^{(2)^T} \phi^{(2)}(x) + b^{(2)}] \\ \vdots & \quad \vdots \\ y^{(n_y)}(x) & = sign[w^{(n_y)^T} \phi^{(n_y)}(x) + b^{(n_y)}] \end{cases} \quad (25)$$

with mappings on a high dimensional feature space $\phi^{(i)}(.) : R^n \rightarrow R^{n_h}$, $i = 1, 2, \ldots, n_y$, with dimensions $n_{h_1}, n_{h_2}, \ldots, n_{h_{n_y}}$.

By the extension of Eq. (23) to a multi-class problem, we obtain

$$\begin{cases} \bar{b} = B\overline{Y} \\ \bar{\alpha} = A(\overline{Y} - \bar{b}1) \end{cases} \quad (26)$$

where matrix $\overline{Y}$ is given by

$$\overline{Y} = \begin{bmatrix} y_1^{(1)}, \ldots, y_N^{(1)} \\ y_2^{(2)}, \ldots, y_N^{(2)} \\ \ldots \\ y_1^{(n_y)}, \ldots, y_N^{(n_y)} \end{bmatrix} \quad (27)$$

and vector $b_M = [b^{(1)}, \ldots, b^{(n_y)}]$, where N is the number of pixels in one processing block.
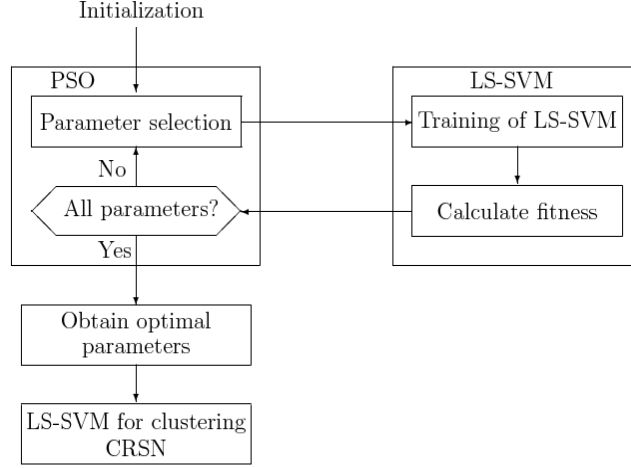
Fig. 1. Block diagram of PSO-LS-SVM-based algorithm.

4   Experimental Results

An experiment to verify clustering efficiency of PSO-LS-SVM approach was performed for randomly deployed Pus and CRSN nodes in a 200 m x 200 m square area. We assumed that each PU randomly operates in four channels in which the neighbouring Pus are inactive. In the simulation, we supposed that the distance between a transmitting and receiving radio pair is normally distributed with a mean of 30 and variance of 15.

We used a link gain of $h = h_i/d$, where d is the distance from transmitter to receiver. If the distance is was less than 10, we set link gain to one.

The proposed PSO-LS-SVM approach for clustering in CRSN is composed of two main phases. PSO algorithm was applied as a technique to improve the setting of the parameter values of LS-SVM. Clustering using LS-SVM method consists of two phases: training phase and calculate fitness phase. A block diagram of the algorithm is depicted in Fig. 1.

The PSO-LS-SVM algorithm was developed based on Matlab R2014b and LS-SVMlab 1.8 for the optimization of model. Through training, the parameter values of the PSO and LS-SVM approach were set as follows. The cognition learning factor c_1 and the social learning factor c_2 were set to 1.8. The number of particles and generations were found to be 8 and 150. Since the PSO-LS-SVM approach has a larger solution space, the number of solution evaluated was raised to 1000. The optimized parameters for LS-SVMs inlude kernel parameters and regularization parameter. If radial basis function (RBF) or polynomial (Pol) function is chosen as a kernel function, we used the vector $\nu = (\gamma, \sigma_1, \sigma_2, \ldots, \sigma_{N_{input}})$ and $\nu = (\gamma, \sigma, \delta)$, respectively, where γ is the regularization parameter, δ and $\sigma_i (i = 1, 2, \ldots, N_{input})$ are kernel parameters, N_{input} is the dimension number of input patterns.
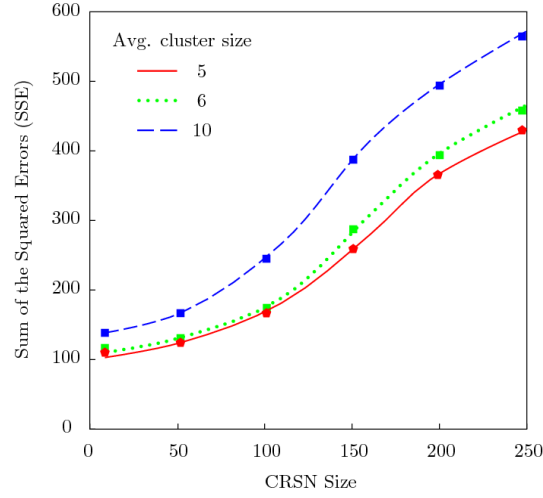
Fig. 2. Average sum of the squared errors (SSE) versus size of the CRSN.

To perform the clustering evaluation, six PUs are randomly deployed in our sensor field. Furthermore, the sum of the squared errors (SSE) of the clustering results for various numbers of clusters was taken into account for each scenario. In Fig. 2, we compare the average SSE with respect to CRSNs of different sizes for all of the studied number of clusters. The experiment shows that the method for a given CRSN has clustering effectiveness with the minimal average SSE, when the cluster number is equal to five or six.

Fig. 3 plots the average cluster size versus the number of nodes in the CRSN and given number of active PUs. As we can see, all PUs behave similarly in terms of the number of clusters and the cluster size for different numbers of PUs for various CRSN sizes. However, the small number of PUs gives a greater average cluster size.

To compare the efficiency of LS-SVM and PSO with various kernels, we used the Rand index. Rand index measures the percentage of decisions that are correct, namely

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \tag{28}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

Table 1 presents the values of Rand index for the PSO-LS-SVM approach under different circumstances. As can be seen from Table 1, the efficiency of PSO-LS-SVM with the mixture of kernels is better than other models.
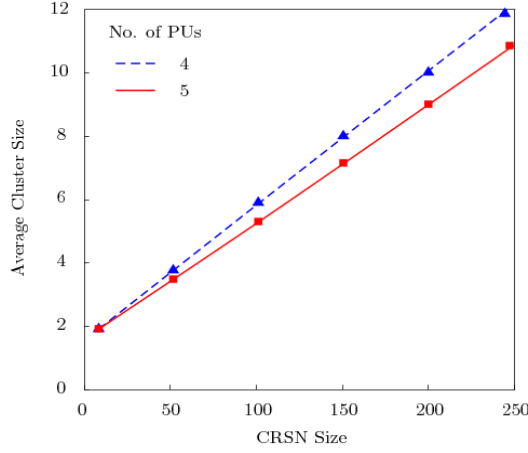
Fig. 3. Average cluster size versus size of the CRSN.

Table 1. Efficiency comparison between various models obtained by
PSO-LS-SVM.

Model	RI (%)
LS-SVM^{Poly}	61.8
LS-SVM^{RBF}	65.4
LS-SVM^{mix}	67.5
PSO-LS-SVM^{Poly}	83.6
PSO-LS-SVM^{RBF}	88.2
PSO-LS-SVM^{mix}	89.3

5   Conclusion

In this paper, a new clustering method for cognitive radio sensor networks is introduced. The applied methodology used in clustering has been found in the network energy consumption model for CRSNs. Three parts of this model were used: multi-task sensing, signalling and data transmission. Using the energy distribution of the sensor field surface and applying the PSO-LS-SVM system, we have been able to solve the clustering problem in the CRSN.

The presented hybrid artificial intelligence system is shown to satisfy CRSN energy constraints through extensive simulation experiments. Our PSO-LS-SVM method showed that the proposed solution offers several advantages with respect to classical clustering schemes, such as k-means clustering algorithm, etc. Moreover, this method incorporates the spectrum-sensing capability and spectrum-aware sensing constraint in the nodes and their localisation in the network. The presented method achieves comparable effectiveness with the other clustering method in cognitive radio sensor networks. Future research directions include

the development of clustering methods for CRSN with energy constraints by allowing each node to select its role without the need for cooperation between nodes.

References

[1]   Akan, O.B, Karli, O.B., Ergul, O.: Cognitive Radio Sensor Networks, IEEE Networks 22, 34-40 (2009)

[2]   I.F. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: A Survey, Computer Networks 38, 393-422 (2002)

[3]   Bradonji, M., Lazos, L.: Graph-based Criteria for Spectrum-aware Clustering in Cognitive Radio Networks, Ad Hoc Networks 10, 75-94 (2012)

[4]   Cherkassky, V., Ma, Y.: Practical Solution of SVM Parameters and Noise Estimation for SVM Regression, Neural Networks 17, 113-126 (2004)

[5]   Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proc. of IEEE Int. Conf. on Neural Networks 4, 1942-1948 (1995)

[6]   Kennedy, J., Spears, W. M.: Matching Algorithms to Problems, An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimodal Problem Generator. In: Proc. Int. Conf. on Evolutionary Computation. Anchorage, AK, USA (1998)

[7]   Kuhn, H., Tucker, A.: Nonlinear Programming. In: Proc. of the 2nd Berkeley Symp. on Mathematical Statistics and Probabilistics, University of California Press, pp. 481-492 (1951)

[8]   Mercer, J.: Functions of positive and negative type and their connection with the theory of integral equations, Phil. Trans. of the Royal Society A 209, 44458 (1909)

[9]   U.S. Frequency Allocation, http://www.ntia.doc.gov/osmhome/allochrt.pdf (2002)

[10] Smits, G.F., Jordan, E.M.: Improved SVM Regression Using Mixtures of Kernels. In: Proceedings of IJCNN 02 on Neural Networks 3, pp. 2785-2790 (2002)

[11] Suykens, J.A.K., Vandewalle, J.: Least Squares Support Vector Machine Classifier, Neural Process. Lett. 9, 293-300 (1999)

[12] Suykens, J.A.K., Lukas, L., Vandewalle, J.: Sparse Approximation Using Least Squares Support Vector Machines. In: Proc. of the IEEE Int. Symp. on Circuits and Systems (ISCAAS 2000), 2, pp. 757-760 (2000)

[13] Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machines, World Scientific, Singapore (2002)

[14] Vapnik, V.N.: The Nature of Statistical Learning Theory, Springer, Berlin (1995)

[15] Vapnik, V.N.: Statistical Learning Theory, Wiley, New York (1998)

[16] Yau, K.-L., Komisarczuk, P., Teal, P.D.: Cognitive Radio-based Wireless Sensor Networks: Conceptual Design and Open Issues. In: IEEE 34th Conf. on Local Computer Networks, pp. 955 - 962 (2009)

[17] Younis, O., Fahmy, S.: Heed: a Hybrid, Energy-efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks, IEEE Trans. on Mobile Computing 3, 366-379 (2004)

[18] Zhang, H., Zhang, Z., Dai, H., Yin, R., Chen, X.: Distributed Spectrum-aware Clustering in Cognitive Radio Sensor Networks, Global Telecommunications Conference (GLOBECOM 2011) 1 - 6 (2011)

[19] Zhang, H., Zhang, Z.Y., Yuen, Ch.: Energy-efficient Spectrum-aware Clustering for Cognitive Radio Sensor Networks, Chinese Science Bulletin, 57, pp. 3731-3739 (2012)