

Wireless Sensor Network Localization Based on a Mobile Beacon and TSVM

Jiang Xu^{1,2}, Huanyan Qian¹, Huan Dai², Jianxin Zhu³

¹School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China

²School of Computer Science and Engineering, Changshu Institute of Technology, Changshu, Jiangsu 215500, China

³Department of Self-Propelled Gun, Shenyang Academy, Shenyang, Liaoning 110162, China

Emails: skydestinyx@gmail.com hyqian@njust.edu.cn 184700185@qq.com
1374861408@qq.com

Abstract: *In this paper a new wireless sensor network localization algorithm, based on a mobile beacon and TSVM (Transductive Support Vector Machines) is proposed, which is referred to as MTSVM. The new algorithm takes advantage of a mobile beacon to generate virtual beacon nodes and then utilizes the beacon vector produced by the communication between the nodes to transform the problem of localization into one of classification. TSVM helps to minimize the error of classification of unknown fixed nodes (unlabeled samples). An auxiliary mobile beacon is designed to save the large volumes of expensive sensor nodes with GPS devices. As shown by the simulation test, the algorithm achieves good localization performance.*

Keywords: *WSNs, localization, mobile beacon, TSVM.*

1. Introduction

How to localize the position of each node? This is a very important research field for a wireless sensor network. In sensor networks, in order the data collected by nodes to make sense, it must be attached to their position information. Moreover,

coverage, deployment, target tracking and other related operations rely much on the efficient localization of nodes. After years of research and development, the scholars have put forward plenty of node localization algorithms. In accordance with the principle whether the practical distance between nodes is to be measured, the localization algorithms may be divided into range-based algorithms and range-free algorithms. In range-based localization algorithms, the most frequently applied distance measurement methods include Time Difference Of Arrival (TDOA) [1], Angle Of Arrival (AOA) [2] and Received Signal Strength Indication (RSSI) [3]. All of them require additional hardware support, and normally, the hardware is quite expensive, consuming much energy. The range-free localization algorithms include DV-Hop [4], MDS-MAP [5] and MDL [6]. In recent years, many scholars proposed also the localization algorithm based on machine learning [7-10]. This algorithm firstly builds a prediction model by training and learning in the deployment area, and then further estimates the position information of unknown nodes in the area.

Inspired by machine learning algorithms of Transductive Support Vector Machines (TSVM) [11] and Progressive Transductive Support Vector Machines (PTSVM) [12], in this paper an auxiliary mobile beacon marching forward, following a planned route will be employed to present a novel localization algorithm – MTSVM to estimate the position information of the unknown nodes in wireless sensor networks. The geographical area to be deployed with wireless sensor nodes is partitioned into several grids. On this basis the nodes will be classified into these grids. Thus, the problem of localization is naturally transformed into a problem of classification. The training data required by TSVM is derived from signal vectors between the nodes for communication. These signal vectors are comprised by few labeled samples and plenty of unlabeled samples. In this paper the signal vectors for communication between the auxiliary mobile beacon nodes and unknown fixed nodes are referred to as labelled samples. By contrast, the signal vectors for communication between the unknown fixed nodes are denoted as unlabeled nodes. Another important feature of TSVM is that in the learning process with mixed samples, the sample distribution information of a test set will be transferred from unlabeled samples to the final classifier. Since the number of unlabelled samples is larger, compared to labeled samples, they are able to better describe the data feature in the entire sample space, so as to endow the trained classifier with a better propagation performance. In this paper it is assumed that the nodes in the area are able to communicate via a signal. As for this, the algorithm is not fit for large scale networks.

The rest of this paper is organized as follows. Section 2 briefly introduces theories related to Support Vector Machines (SVM) [13] and TSVM. Section 3 gives detailed explanation of MTSVM algorithm. Section 4 shows the simulation test result. In Section 5 the research in this paper is summarized and the future work is prospected.

2. SVM and TSVM

2.1. SVM

Assuming that there are n training data samples x_i , $i = 1, 2, \dots, n$, $x_i \in R^m$, with their corresponding classes separately labeled as y_1, y_2, \dots, y_n , these samples are divided into two categories: H and $\neg H$ (non- H). If $x_i \in H$ then $y_i = 1$; or else, $y_i = -1$. Afterwards, it is required to predict whether a new data sample x belongs to H or not. SVM is just designed to solve this problem.

Defining a kernel function $K(x, x')$, this function has to comply with Mercer condition [14]. The central station then solves the following optimization problem:

$$(1) \quad \begin{aligned} & \min \frac{1}{2} \|w\|^2, \\ & \text{s.t.}: y_i[(w \cdot x_i + b)] \geq 1, \quad i = 1, 2, \dots, n. \end{aligned}$$

The decision function is defined as

$$f(x) = \text{sgn}(w \cdot x + b).$$

Under the linearly separable condition, all training samples shall comply with $|f(x)| \geq 1$. In the following discussion, the area complying with $|f(x)| < 1$ will be denoted as a margin band for the separating hyperplane.

Assuming that the training data samples are linearly non-separable, (1) may be modified as

$$(2) \quad \begin{aligned} & \min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \varepsilon_i, \\ & \text{s.t.}: y_i[(w \cdot x_i + b)] \geq 1 - \varepsilon_i, \quad i = 1, 2, \dots, n, \\ & \quad \varepsilon_i \geq 0, \quad i = 1, 2, \dots, n. \end{aligned}$$

With Wolfe theory the problem can be transformed to its dual problem.

$$(3) \quad \begin{aligned} & \max W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j), \\ & \text{s.t.}: \sum_{i=1}^n \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, n. \end{aligned}$$

Assuming that the optimal solution for the quadratic programming problem has been worked out, which is $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)^T$, it is hereby possible to further obtain $w^* = \sum_{i=1}^n \alpha_i^* y_i x_i$, $b^* = y_j - \sum_{i=1}^n y_i \alpha_i^* K(x_i, x_j)$, with the subscript accepted as $j \in \{j | \alpha_j^* > 0\}$. Then the decision function is obtained:

$$(4) \quad \begin{aligned} & f(x) = \text{sgn}(g(x)), \\ & g(x) = (w^* \cdot x) + b^* = \sum_{i=1}^n \alpha_i^* y_i K(x, x_i) + b^*. \end{aligned}$$

So far, drawing a support from the kernel function $K(x, x')$, the input space is successfully projected into a higher dimensional Hilbert space to work out the hyperplane $g(x)$. Moreover, the geometric interval of the training sample set in this hyperplane is maximized. In this paper, a radial basis function is used as a kernel function.

$$(5) \quad K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{\sigma^2}\right).$$

2.2. TSVM

Assuming that the following two sample sets are given: A group of independent identically distributed labeled training sample sets $(x_1, y_1), \dots, (x_n, y_n)$, $x_i \in \mathbb{R}^n$, $y_i \in (-1, 1)$ and another group of unlabelled sample sets with the same distribution $y_1^*, y_2^*, \dots, y_k^*$, the next step is to find a solution in order to classify the unlabeled samples set $y_1^*, y_2^*, \dots, y_k^*$, and to maximize the spacing of the jointed sequence $(x_1, y_1), \dots, (x_n, y_n)$, $(x_1^*, y_1^*), \dots, (x_k^*, y_k^*)$. Under a normal linearly non-separable condition, the training process of TSVM may be described as the following optimization problem:

$$(6) \quad \begin{aligned} & \text{Minimize over } (y_1^*, \dots, y_n^*, w, b, \varepsilon_1, \dots, \varepsilon_n, \varepsilon_1^*, \dots, \varepsilon_k^*), \\ & \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \varepsilon_i + C^* \sum_{j=1}^k \varepsilon_j^*, \\ & \text{s.t.: } y_i[w \cdot x_i + b] \geq 1 - \varepsilon_i \quad \forall i=1, \dots, n, \\ & \quad y_j^*[w \cdot x_j^* + b] \geq 1 - \varepsilon_j^* \quad \forall j=1, \dots, k, \\ & \quad \varepsilon_i > 0 \quad \forall i=1, \dots, n, \\ & \quad \varepsilon_j^* > 0 \quad \forall j=1, \dots, k, \end{aligned}$$

where C and C^* are the parameters to be assigned and adjusted by users, which play the same role as parameter C in (2). In the training process, 1-2 unlabeled samples with possibly major influence on the subsequent training process are selected, being endowed with the most possible label under the preset state. On this basis, the samples will be included in the label samples for another round of training. Normally, new included samples may affect a new round of training, leading to a slight offset to present a hyperplane. In this process it might be discovered that some previously made labels are unsuitable. Once this happens, the unfit labels shall be cancelled, and the samples shall be restored to an unlabelled state. By this meticulously designed incremental assignment and dynamic adjustment rule, the predicted hyperplane will become closer and closer to the optimal hyperplane in training, and finally figure out a local optimal solution for (6).

3. MTSVM

3.1. Problem description and model

It is assumed that m wireless sensor network nodes s_1, s_2, \dots, s_m with unknown positions are deployed in a 2D area $[0, D] \times [0, D]$ ($D > 0$). Then the 2D area is divided into n grids, as shown in Fig. 1 (the following part of this paper mainly discusses the classification of X axis, while Y axis is exactly the same). Our objective is to estimate the 2D coordinates of these unknown nodes.

Defining two class sets, with each class set containing T classes:

- Defining T classes $\{\text{TSVM}_{x_1}, \text{TSVM}_{x_2}, \dots, \text{TSVM}_{x_T}\}$ along axis X , while each class TSVM_{x_k} contains all the nodes when $X \geq iD/T$;
- Defining T classes $\{\text{TSVM}_{y_1}, \text{TSVM}_{y_2}, \dots, \text{TSVM}_{y_T}\}$ along axis Y , while each class TSVM_{y_k} contains all the nodes when $Y \geq iD/T$.

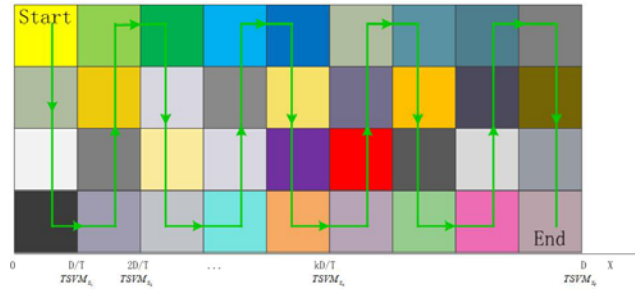


Fig. 1. The 2D area is divided into $T \times T$ grids. Green arrow indicates the moving direction of mobile beacon

Expression $\text{SS}(S_j, S)$, $j = 1, 2, \dots, m$, refers to the Received Signal Strength (RSS) of node S from a random node S_j ; s_j , $s_j = (\text{SS}(S_1, S_j), \text{SS}(S_2, S_j), \dots, \text{SS}(S_m, S_j))$, is defined as the signal vector, where y_j refers to the class label. In this paper a mobile node S_b is employed to assist the localization, which is deployed to march (lingering for a certain period in each grid) along the direction pointed by the green arrow in Fig. 1. The mobile beacons are usually outfitted with a GPS device, so that mobile beacon S_b may be able to generate n virtual beacon nodes S_{bi} and their separate position information. These virtual beacons are configured to receive RSS from all unknown fixed nodes in the network, leading to the signal vector s_{bi} , $s_{bi} = (\text{SS}(S_1, S_{bi}), \text{SS}(S_2, S_{bi}), \dots, \text{SS}(S_m, S_{bi}))$, as well as the class label y_{bi} , $y_{bi} \in \{1, 2, \dots, T\}$, for each class TSVM_{x_k} . The data information will be used as labeled data sample for TSVM. By contrast, the signal vector between unknown

fixed nodes, denoted as s_j , $s_j = (\text{SS}(S_1, S_j), \text{SS}(S_2, S_j), \dots, \text{SS}(S_m, S_j))$, as well as the corresponding label y_j^* will be taken as unlabeled data sample for TSVM.

At each step of the training, the present labeled samples will be used to do inductive learning to figure out the decision function for the present hyperplane, with the form shown in (4). In addition, the decision function value for all present unlabeled samples will also be calculated. On this basis, a new positive label will be made according to the following expression (7) and a new negative label will be made according to (8), i.e., labeling of two unlabelled sample nodes at once.

$$(7) \quad \max(f(x_i^*)), \text{ s.t. } 0 < f(x_i^*) < 1,$$

$$(8) \quad \min(f(x_i^*)), \text{ s.t. } -1 < f(x_i^*) < 0.$$

If there is no unlabeled sample that complies with (7) after training no new position label will be made in the current circulation. Similarly, if there is no unlabeled sample that complies with (8), no new negative label will be made in the current circulation. This process will be continued, until no unlabeled sample appears within the margin band area of the present optimal hyperplane. Then it would be considered that the problem has got its optimal solution. All the rest unlabeled samples will be classified and labeled by a current decision function. After that the calculation is ended and the result is obtained. This process is defined as a pair-based labeling method. In the labeling process, it is possible that after training, the labeled no-label sample value is different from the label value obtained by classification with the present hyperplane. This may be caused by an error labeling at the early stage of the iteration process. Under such circumstance, the sample will be reset as unlabeled, so as to continue to go through the iteration process. In this way this sample may get a more reliable new label in future trainings, and this approach is referred to as a label reset.

If an unknown node is estimated belonging to TSVM_{x_k} and not belonging to $\text{TSVM}_{x_{k+1}}$, the horizontal coordinate of the unknown node may hereby be denoted by $\left(k + \frac{1}{2}\right)D/T$. If the classification estimation is correct, the maximum localization error of the unknown node will be $D/\sqrt{2T}$.

3.2. Classification strategy and algorithm description

It may be seen from the above discussion that the classification problem in this paper belongs to a multi-class classification problem. Common multi-class classification methods include one-to-rest, one-to-one and a decision binary tree method. In accordance with the feature of the training samples, a multiple class to multiple class classification method is proposed in this paper, in order to construct T two-class classifiers. Each classifier will be used to separate the samples on the left side from the samples on the right side. On this basis we are to estimate the attribution of unknown nodes.

The detailed classification strategy is described in two steps as follows:

Step 1. Defining a training set

$$P = \{(x_1, y_1), \dots, (x_n, y_n)\},$$

where $x_i = s_{bi} \in R^m$, $y_i \in \{1, \dots, T\}$, $i = 1, 2, \dots, n$.

Step 2. For $k = 1, 2, \dots, T$, the following calculation will be made. According to Fig. 1, the training samples on the right of class TSVM _{x_k} are taken as positive samples, and the samples on the left will be considered as negative samples. A Decision Function $f^k(x)$ is applied to decide whether the input x belongs to TSVM _{x_k} or not:

$$(9) \quad f^k(x) = \text{sgn}(g^k(x)) = \text{sgn}(g^k(x) = \sum_{i=1}^n y_i \alpha_i^k K(x, x_i) + b^k).$$

MTSVM localization algorithm based on TSVM is described in five steps as follows:

Step 1. As described in 3.1, after a mobile beacon gets TSVM training data, the parameters C and C^* will be configured, while SVM training program will be executed, and figure out α^* and the corresponding b^* for class TSVM _{x_i} . Then, the mobile beacon will transmit the data to the entire network. All unknown nodes will store the data information, and then calculate the decision function of each class.

Step 2. The auxiliary mobile beacon retreats from the network. The unknown nodes establish mutual communication and all the nodes may get a signal vector s_j . Plugging the signal vector into (9), it is easy to figure out the classification result of all unknown fixed nodes with regard to all the classes. Then the nodes are labeled. Afterwards, the aforementioned pair-based labeling method is applied to make a new positive label and a new negative label for unlabeled samples in the present margin band.

Step 3. Repeating the training for all the samples, and calculating the discrimination function output for all unlabeled samples. If the label value of a certain preliminarily marked unlabeled sample is inconsistent with respect to current discrimination function output value, the label for the sample shall be cancelled according to a previously discussed label reset rule.

Step 4. The pair-based labeling method is applied to seek for unlabeled no-label samples in the present margin band, which complies with the new labeling condition. If there is such unlabeled sample, it must be labeled and then turn to Step (3). If there is no such unlabeled sample, the present hyperplane decision function will be used to classify and label the rest of all unlabeled samples and output the result.

Step 5. If an unknown fixed node S_j belongs to TSVM _{x_k} and does not belong to TSVM _{x_{k+1}} , then the horizontal coordinate (X axis) of S_j is estimated to be $(k + \frac{1}{2})D/T$. The vertical coordinate of S_j may be figured out in the same way.

4. Simulation

Assume that a wireless sensor network $\{S_1, S_2, \dots, S_m\}$ comprised by m unknown fixed nodes is randomly deployed in a $60m \times 60m$ 2D area, with $m = 500$ and $m = 300$. The ID of the nodes are separately $1, 2, \dots, m$, while the communication radius is $r = 60$ m. Then the 2D area is divided into $T \times T$ square grids, separately denoting $T = 6, T = 8, T = 10, T = 12, T = 15$, and $T = 20$ (i.e., the side length of the square grid is separately configured as 10 m, 7.5 m, 6 m, 5 m, 4 m and 3 m). The objective of the node localization is to work out the estimated position (\hat{x}_i, \hat{y}_i) of the unknown fixed node, in order to make (\hat{x}_i, \hat{y}_i) closer to the real coordinates (x_i, y_i) of the unknown fixed node as much as possible. In this paper MATLAB is applied to invoke LIBSVM software to implement the algorithm, and Average Localization Error (ALE) is adopted to measure the performance of the algorithm. The formula is as follows:

$$(10) \quad \text{ALE} = \frac{\sum_{i=1}^m \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}}{m},$$

where (\hat{x}_i, \hat{y}_i) denotes the estimated coordinates of i -th node, and m stands for the number of unknown fixed nodes. Fig. 2 shows the distribution of nodes when $m = 500$ and $T = 6$, and $T = 10$.

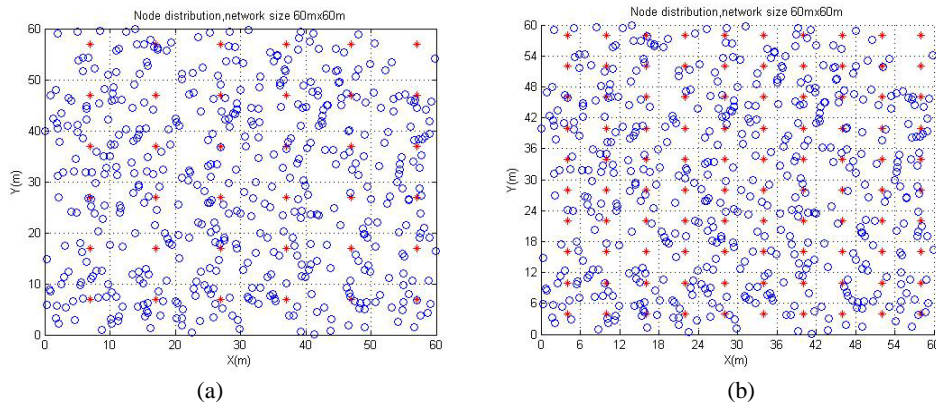


Fig. 2. Node distribution under two different classification patterns when $m = 500$. Blue “o” stands for the unknown nodes, red “*” stands for generated virtual beacon nodes: $T = 6$ and $m = 500$ (a); $T = 10$ and $m = 500$ (b)

It can be seen from Fig. 3 that as the number of classes increases, the average localization error tends to decrease.

However, when $T \geq 15$, the descending tendency is no longer obvious. This is mainly caused by the classification error. On the other hand, according to the Fig. 3, the average localization error, when $m = 500$ is slightly lower than that when

$m = 300$. However, the difference is not significant. This may help to demonstrate that the higher density of nodes leads to higher localization precision. Besides, on the premise with dense nodes in the network, the localization effect of MTSVM is more stable and MTSVM is insensitive to nodes distribution and number variation.

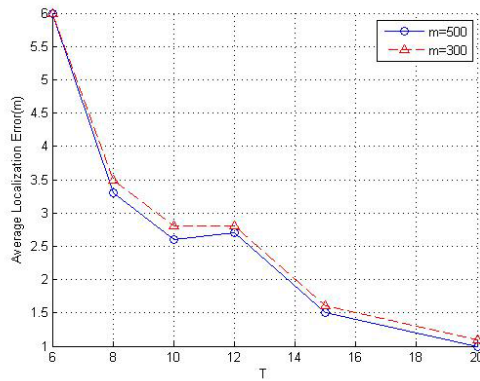


Fig. 3. MTSVM average localization error

Fig. 4 shows the average localization error of a traditional SVM and MTSVM under the environment when $m = 500$ and $m = 300$. As shown in the graph, the localization error of MTSVM is obviously lower than SVM's. This must be due to the high classification accuracy of MTSVM.

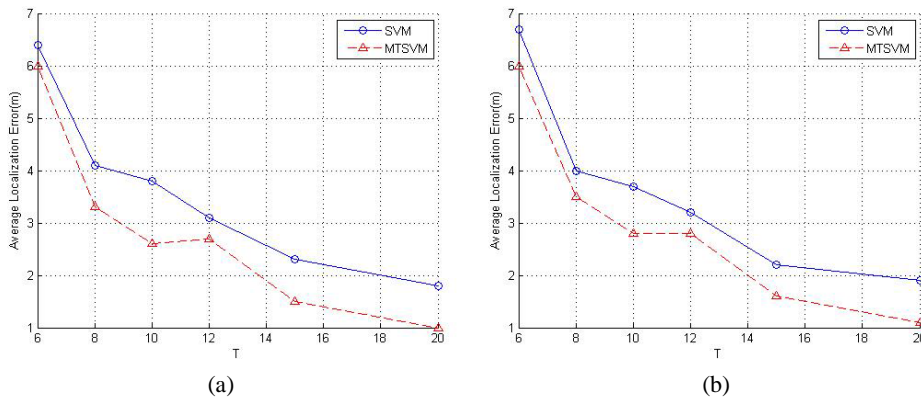


Fig. 4. ALE comparison of SVM vs. MTSVM: average localization error when $m = 500$ (a); average localization error when $m = 300$ (b)

5. Conclusion

In connection with the defects of the wireless sensor networks, such as limited energy and high beacon node cost, in this paper a mobile beacon is adopted to generate a virtual beacon. Inspired by the learning algorithm, TSVM-based MTSVM algorithm is proposed to estimate the position, achieving high localization accuracy. However, since direct communication between nodes is required to get

the signal vector, MTSVM is only applicable to dense wireless sensor networks. In addition, when there are too many unlabeled samples (unknown fixed nodes), frequent pairing labeling and re-training after the label reset have increased the complexity of MTSVM algorithm. This is one disadvantage of MTSVM. One possible improvement is to accelerate the training speed by using some faster learning algorithms, such as the incremental and decremental learning methods [15]. Furthermore, the way to apply a learning algorithm in node localization under a large scale network environment is our next.

Acknowledgments: This work is supported by NSFC (Grant No 61300186), and University Science Research Project of Jiangsu Province (Grant No 13KJB510001).

References

1. Girod, L., D. Estrin. Robust Range Estimation Using Acoustic and Multimodal Sensing. – In: Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2001, 1312-1320.
2. Niculescu, D., B. Nath. Ad Hoc Positioning System (APS) Using AOA. – In: Proc. of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, Rutgers University, Piscataway, 30 March-3 April 2003, 1734-1743.
3. Krumm, J., G. Cermak, E. Horvitz. Right SPOT, a Novel Sense of Location for a Smart Personal Object. – In: Proc. of Ubicomp, 2003, 36-43.
4. Niculescu, D., B. Nath. DV Based Positioning in Ad Hoc Networks. – J. Telecommun. Syst., Vol. **22**, 2003, 267-280.
5. Shang, Y., W. Ruml, Y. Zhang, M. P. J. Fromherz. Localization from Mere Connectivity. – In: Proc. of 4th ACM Int. Symp. on Mobile Ad Hoc Networking & Computing, 2003, 201-212.
6. Lee, H., S. Lee, Y. Kim, H. Chong. Grouping Multi-Duolateration Localization Using Partial Space Information for Indoor Wireless Sensor Networks. – IEEE Trans. on Consumer Electronics, Vol. **55**, 2009, No 4, 1950-1958.
7. Brunato, M., R. Battisti. Statistical Learning Theory for Location Fingerprinting in Wireless LANs. – Comput. Networks, Vol. **47**, 2005, 825-845.
8. Nguyen, X., M. I. Jordan, B. Sinopoli. A Kernel-Based Learning Approach to Ad Hoc Sensor Network Localization. – ACM Trans. Sensor Networks, Vol. **1**, 2005, 134-152.
9. Pan, J. J., J. T. Kwok, Y. Chen. Multidimensional Vector Regression for Accurate and Low-Cost Location Estimation in Pervasive Computing. – IEEE Trans. on Knowledge and Data Engineering, Vol. **18**, 2006, 1181-1193.
10. Tran, D. A., T. Nguyen. Localization in Wireless Sensor Networks Based on Support Vector Machines. – IEEE Trans. Parallel Distrib. Syst., Vol. **19**, 2008, 981-994.
11. Joachims, T. Transductive Inference for Text Classification Using Support Vector Machines. – In: Proc. of 16th Int. Conf. on Machine Learning (ICML), San Francisco, 1999, 200-209.
12. Chen, Y., G. Wang, S. Dong. A Progressive Transductive Inference Algorithm Based on Support Vector Machine. – Journal of Software, Vol. **14**, 2003, No 3, 451-460.
13. Cortes, C., V. Vapnik. Support-Vector Networks. – Machine Learning, Vol. **20**, 1995, 273-297.
14. Scholkopf, B., A. J. Smola. Learning with Kernels-Support Vector Machines, Regularization, Optimization and Beyond. Cambridge, MIT Press, 2002.
15. Cauwenberghs, G., T. Poggio. Incremental and Decremental Support Vector Machine Learning, Advances Neural Information Processing Systems. Cambridge, MIT Press, 2001, 409-415.