A Kernel-Based Node Localization in Anisotropic Wireless Sensor Network

Wenxiu He, Fangfang Lu, Jingjing Chen, Yi Ruan, Tingjuan Lu, and Yi Zhang

1Zhijiang College, Zhejiang University of Technology, Hangzhou, China
2College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, China
3School of Science, Zhejiang University of Technology, Hangzhou, China
4Department of Information Technology, The 903th Hospital of PLA, Hangzhou 310004, China
5The 2nd Research Institute of Civil Aviation Administration of China, Chengdu, China

Correspondence should be addressed to Fangfang Lu; lufangfang@shiep.edu.cn

Received 3 April 2021; Accepted 10 June 2021; Published 21 June 2021

1.Introduction

Currently, wireless sensors localization is still the main problem concerning wireless sensor networks (WSN). The localization algorithms of WSN can be classified into the range-based measurement method and the range-free measurement method. The former one can get high accuracy but need range information [1–4], while the latter one gets low accuracy without any range information.

In order to improve the accuracy of range-free node localization, machine learning is introduced to the localization of WSN [5]. Preciously, artificial neural network (ANN) was used in range-free localization algorithms, and its accuracy and performance greatly promoted when compared with other traditional algorithms [6–12]. Besides, Phoemphon et al. [13, 14] have used a fuzzy logic to range-free localization in WSNs. This algorithm mainly focuses on the heterogeneous scenarios with lower complexity. A graph-based localization algorithm was presented using conventional neural networks (CNN) and support vector machine (SVM) [15] in the paper. Another localization method using SVM for large-scale WSNs was proposed in [16]. A fuzzy c-means training sample reduction (FCMTSR) method was used to reduce the training overhead and learning calculation. FCMTSR-SVM algorithm not only improves the localization accuracy but also reduces the training samples time. Based on SVM, a new algorithm called support vector machine-based range-free localization (RFSVM) has been presented in [17]. A transmit matrix was introduced to show the relation between hops and distances and to train the system, while SVM was used to find the unknown nodes in WSNs. Another range-free localization method has been developed using SVM classifier [18]. Recently, kernel-based approach is first proposed in isotropic and anisotropic WSN [19]. Active AVR is used in the localization and the simulating results show an improved performance in localization [20].

In this paper, in order to improve the accuracy of node localization in anisotropic WSN, we introduce kernel regression approach to node localization firstly. The contribution in this paper can be summarized to (1)
systematically review the literature regarding the approaches for wireless sensors localization and identity the existing issues in this field and (2) to propose a novel graph-based localization approach and introduce the design and implementation of the proposed approach in detail and (3) a set of experiments comparing proposed approach with existing ones are performed and discussed. The rest of the paper is organized as follows. Section 2 discusses the related theories. The proposed algorithm is discussed and algorithm simulation and analysis are presented in Section 3. The conclusions and future scope are highlighted in Section 4.

2. Materials and Methods

2.1. Relative Theories. SVM [21] is a universal machine learning method based on statistical learning theory framework. Vapnik and Lerner began to study machine learning problem with limited samples from 1960s [22] and then Vapnik and Chervonenkis [23] proposed VC theory and statistical learning theory, which were constructed on the principle of structural risk minimization. Later in 1982, Cortes and Vapnik [24] further proposed structural risk minimization (SRM) principle. It is pockmarking because it provides a same framework for solving the learning problem of limited samples and helps to solve a series of problems, including selecting neural network structure and local minimum [25, 26].

The methods based on SVM are widely used in pattern recognition, regression estimate, etc. [27–30]. The better efficiency and accuracy are achieved in pattern recognition field, handwritten numeral recognition, and image recognition [31].

SVM includes support vector classifier and support vector regression. It was first proposed to solve the problem of classification at the beginning, including linear classification and nonlinear classification. To solve the problem of nonlinear classification, SVM transforms the training set into a high-dimensional Hilbert space through a nonlinear transformation, and the linear classification can be obtained in the high-dimensional space through an optimal classification hyperplane, which can improve the nonlinear processing capability and reduce the dimension.

**Definition 1** (kernel function [13]). Supposing \( \chi \) is a subset of \( \mathbb{R}^n \). \( K(x, x') \) defined in \( \chi \times \chi \) is a kernel function if there is a mapping function \( \Phi \) from \( \chi \) to a Hilbert space \( H \) to make \( K(x, x') = \langle \Phi(x) \cdot \Phi(x') \rangle \), \( \langle \cdot \rangle \), indicating the inner product in space \( H \).

Mercer’s theorem shows the requirements of kernel function, which is also the requirement of a symmetric function \( K(x, x') \) correspondent to the inner product of a feature space.

**Theorem 1** (Mercer’s theorem [13]). Supposing \( \chi \) is a compact set of \( \mathbb{R}^n \). \( K \) is a symmetric function with continuous real value in \( \chi \times \chi \). Integral operator \( T_K: L_2(\chi) \longrightarrow L_2(\chi) \), then

\[
T_K f(x) = \int_\chi K(\cdot, x') f(x') dx'.
\]

Supposing \( \psi_j \in L_2(\chi) \) are the eigenfunctions of \( T_K \) when eigenvalue \( \lambda_j \neq 0 \) and \( \| \psi_j \| = 1 \), then

1. \( \lambda_j (T_K) \psi_j \in L_2(\chi) \)
2. \( \psi_j \in L_\infty(\chi) \) and \( \sup \| \psi_j \| < \infty \)
3. \( K(x, x') = \sum \lambda_j \psi_j(x) \psi_j(x') \) is workable to all \( (x, x') \) and is uniform convergence to all \( (x, x') \) sequences

Kernel function meeting the conditions of Mercer’s theorem is called Mercer kernel. Mercer’s theorem indicates that \( K(x, x') \) is a kernel function if function \( K \) meets the following formula:

\[
\int_{\chi \times \phi} K(x, x') f(x) f(x') dx dx' \geq 0, \quad \forall f \in L_2(\chi).
\]

**Function** \( K(x, x') \) can be written as an inner product in an eigenspace, as shown in the following formula:

\[
K(x, x') = \langle \Phi(x) \cdot \Phi(x') \rangle.
\]

Therefore,

\[
f(x) = \sum_{i=1}^{n} (\pi_i - \bar{\pi}_i)(\Phi(x) \cdot \Phi(x') + \bar{b})
\]

\[
= \sum_{i=1}^{n} (\pi_i - \bar{\pi}_i) K(x, x') + \bar{b}.
\]
Φ: $\phi \subset R^n \rightarrow L_2, x \mapsto \langle\lambda_1 \psi_1(x), \lambda_2 \psi_2(x), \ldots \rangle^T,$ \hspace{1cm} (5)

where $L_2$ is the Hilbert space.

Another equal condition of Mercer kernel is defined by Gram matrix based on definite-sample-definition kernel function. The condition is easy to validate when compared with Mercer kernel.

**Definition 2 (Gram matrix).** Matrix $K(l \times l)$ with line $l$ and column $l$ (element in line $i$ and column $j$ is $K_{ij} = K(x_i, x_j)$) is called Gram matrix of function $K$ with $x_1, x_2, \ldots, x_l$ for a given function $K: \chi \times \chi \rightarrow R$ and $x_1, x_2, \ldots, x_l \in \chi$.

**Theorem 2.** Necessary and sufficient condition of integral operator $T_K$ positive semidefinite is that Gram matrix of $K$ is positive semidefinite with any $x_1, x_2, \ldots, x_l \in \chi$, supposing $\chi$ is a compact set in $R^n$ and $K(x, x')$ is a continuous and symmetric function in $\chi \times \chi$.

Some properties of kernel function can be got from the abovementioned two theorems.

Ratiocination 1: the following functions are kernel functions, supposing $K_1(x, x')$ and $K_2(x, x')$ are kernel functions in $\chi \times \chi$, constant $a \geq 0$, and $p(x)$ is polynomial with all coefficients positive.

\[
K(x, x') = K_1(x, x') + K_2(x, x'),
\]

\[
K(x, x') = aK_1(x, x'),
\]

\[
K(x, x') = K_1(x, x') + K_2(x, x'),
\]

\[
K(x, x') = p(K_1(x, x')),
\]

\[
K(x, x') = \exp K_1(x, x').
\]

(6)

Some kernel functions frequently used can be got according to the above theorems.

(1) Polynomial-kernel function $K(x, x') = ((x \cdot x') + 1)^q$, where $q$ is the order of polynomial.

(2) Radial basis kernel function $K(x, x') = \exp(\|x - x'\|^2/2\sigma^2)$, where $\sigma$ is the width parameter of radial basis function (RBF).

(3) Sigmoid-kernel function $K(x, x') = \tanh[(x \cdot x') + c]$.

In addition, there are Fourier series kernel, B-spline kernel, tensor-kernel functions, etc.

2.2. Range-Free Node Localization Based on Kernel Function [19]. Supposing there is WSN $S = \{S_1, S_2, \ldots S_{M+N}\}$ with $M$
beacons and $N$ nodes ($M \ll N$) and the location of each node can be expressed as follows,

$$\text{pos}(S_i) = (x_i, y_i)^T, \quad i = 1, \ldots, M + N,$$  \hspace{1cm} (7)

here, the location of $M$ beacons $S_i \in A$ is known and the location of other $N$ nodes $S_j \in \Sigma$ is unknown. $A = \{S_i | i = 1, 2, \ldots, M\}$ and $\Sigma = \{S_j | j = M + 1, M + 2, \ldots, M + N\}$.

The only measurement information in the algorithm is approximate information that is indicated by hop number between all the nodes. The geometric distance between $S_i$ and $S_j$ can be defined as follows:

$$d(S_i, S_j) = \left\| \text{pos}(S_i) - \text{pos}(S_j) \right\| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \in \mathbb{R}.$$ \hspace{1cm} (8)

The approximate information between $S_i$ and $S_j$ $p(S_i, S_j) \in Z \in [0, 1, 2, \ldots]$ indicates the hop number between two nodes. Thus, the localization problem can be described as giving $\text{pos}(S_i)$, $d(S_i, S_j)$, and $p(S_i, S_j)$, estimating $\text{pos}(S_o)$.

Here, $S_i, S_j \in A$, $S_k, S_l \in A \cup \Sigma$. The basic idea of the approach is to transfer the localization problem to kernel regression problem. The location of WSN node $S_o \in \Sigma$ can be calculated by the following formula with the aid of approximate information, $p(S_i, S_j)$,

$$f(p) = w^T \phi(p) + b,$$

$$f_i(p) = \sum_{j=1}^{M} (\tilde{a}_{i,j} - \tilde{a}_{i,j})K(p, p_j) + b_i,$$ \hspace{1cm} (9)

where $K(p_j, p_k) \equiv \langle \phi(p_j), \phi(p_k) \rangle$ is the kernel function and $b_j = \sum_{j=1}^{M} (\tilde{a}_{i,j} - \tilde{a}_{i,j})$.

The algorithm consisted of the following three steps:

Step 1. Measurement. Each beacon $S_i \in A$ and each node $S_j \in \Sigma$ exchange their hop counting information and each beacon $S_i$ sends its location to other sensor nodes.

Step 2. Training (SVR). It estimates the distance model SVR of each node and broadcasts it. All other sensor nodes will receive M distance models.

Step 3. Localization. It calculates the distance of the node to all the beacons by using the distance models (SVR1, SVR2, \ldots, SVRm) in the above step and then calculates its location distributive by multilateral measurement method.

The communication cost of the algorithm is $O(MN)$ data packets.

3. Results and Discussion

The localization errors of RBF-kernel-based algorithm, polynomial-kernel-based algorithm, and DV-Hop algorithm are compared. DV-Hop algorithm is a classical algorithm in all the range-free algorithms [34]. Localization error is defined as the distance difference between the real location and estimated location of sensor node and RBF-kernel is the Gauss kernel as follows:

$$K(p_j, p_k) = \exp \left( -\frac{\|p_j - p_k\|^2}{2\sigma^2} \right).$$  \hspace{1cm} (10)

Gauss kernel-based localization through SVR can be denoted as G-LSVR. The extrapolation ability of a polynomial is good when the parameter of the polynomial-kernel function $q = 2$ or $4$ in [35]. Let the power of polynomial $q = 2$ and the polynomial-kernel be

$$\min_{\omega \in \mathbb{R}^n, \xi \geq 0} \frac{1}{2} \|\omega\|^2 + C \sum_{j=1}^{n} (\xi_j + \xi_j^*).$$  \hspace{1cm} (11)

Objective function in formula (12) includes $\|\omega\|^2$ and $\sum_{j=1}^{n} (\xi_j + \xi_j^*)$. $\|\omega\|^2$ indicates confidence interval, and $\sum_{j=1}^{n} (\xi_j + \xi_j^*)$ demonstrates empirical risk. $C$ is a constant, which balances empirical risk and confidence interval.

Polynomial-based localization through SVR can be denoted as P-LSVR.

Specifically, equations (10), (11), and (12) are valid only under the following conditions:

1. Only the connectivity between nodes and location of beacons is needed, while other range information, such as received signal strength indicator, angle, and time, is negligible.

2. The communication range of all sensor nodes and beacons is the same. The sensor nodes and beacons are distributed in either isotropic network or anisotropic network with a square area of $100 \times 100$, as shown in Figures 3 and 4.

(a) Isotropic network: all the sensor nodes are evenly distributed in the whole network and the node density in the network is the same.

(b) Anisotropic network: all the sensor nodes are discrete distributed in the whole network (taking Type-C network as an example).

The following three simulation environments are adopted and the communication range of all the sensor nodes is $R$:

1. Different node density:

(a) Sparse network: 20 beacons and 100 sensor nodes;

(b) Medium network: 20 beacons and 200 sensor nodes;

(c) Dense network: 100 beacons and 300 sensor nodes.

2. The number of sensor nodes in the network is 300 with the node density of beacons varying from 10% to 30%. The communication radius of all the sensor nodes is $R$. 4 Scientific Programming
The parameters have priority than simulations due to the overfitting of kernel-based algorithms. Regularized parameter C in formula (12) controls the balance between complexity and accuracy in G-LSVR for a better generalization. Parameter \( \sigma \) of Gauss kernel function is a constant and \( \sigma \) is crucial to the balance between definiteness and sensitivity. Thus, the value of \( C \) and \( \sigma \) should be determined first by simulations. Tables 1 and 2 show the localization error using algorithm G-LSVR with different \( C \) and \( \sigma \) in the WSN with 100 sensor nodes and 20 beacons. These data are calculated by the mean number of ten individual simulations. It is notable that the minimum localization error is obtained when \( C = 20 \) and \( \sigma = 20 \) in both isotropic WSN and anisotropic WSN.

In simulation environment (1), the communication range \( R \) of all the nodes is 20. The localization performance of the three different algorithms in isotropic WSN is shown in Table 3 and the localization performance in anisotropic WSN is shown in Table 4. It is clear that the three algorithms can get high localization accuracy in isotropic WSN and the performance of P-LSVR and G-LSVR is better than that of DV-Hop. At the same time, the localization error of DV-Hop is very large and the localization error of P-LSVR and G-LSVR is smaller than that of DV-Hop in anisotropic WSN obviously.

The localization error in dense network is minimum to any localization algorithms. It is the balance between node density and localization performance. But sometimes more beacons will cost more resource.

In simulation environment (2), the amount of sensor nodes is set to 300 and the number of beacons vary from 10% to 30%. The communication range of all the nodes is 20. The comparison of localization performance is shown in Figures 5 and 6 in isotropic WSN and anisotropic WSN, respectively. It is noticeable that the accuracy of kernel-based localization algorithm is proportional to the number of beacons. The localization performance of G-LSVR is better than that of P-LSVR, DV-Hop can get better performance in isotropic WSN, while worse performance in anisotropic WSN, in which DV-Hop uses a big hop number to denote long distance in isotropic.

In simulation environment (3), more realistic factors are concerned. Irregular wireless transmitting model [36], as shown in Figure 7, is used to simulate disturbance of communication range \( R \) because of multiaccess channels, disturbance, and other environmental factors. Degree of irregularity (DOI) is used to express the irregularity. Thus, the communication range can be denoted as

\[
(1 - DOI)R \leq \text{communication range} \leq R. \tag{12}
\]

Figures 8 and 9 present the localization performances of the three algorithms in isotropic and anisotropic WSN,
Table 3: Localization error comparison in isotropic WSN.

<table>
<thead>
<tr>
<th>Localization error (meter)</th>
<th>DV-Hop</th>
<th>P-LSVR</th>
<th>G-LSVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 beacons and 100 sensor nodes</td>
<td>12.532</td>
<td>9.472</td>
<td>9.253</td>
</tr>
<tr>
<td>20 beacons and 200 sensor nodes</td>
<td>10.321</td>
<td>9.052</td>
<td>8.975</td>
</tr>
<tr>
<td>100 beacons and 300 sensor nodes</td>
<td>8.939</td>
<td>8.479</td>
<td>7.896</td>
</tr>
</tbody>
</table>

Table 4: Localization error comparison in anisotropic WSN.

<table>
<thead>
<tr>
<th>Localization error (meter)</th>
<th>DV-Hop</th>
<th>P-LSVR</th>
<th>G-LSVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 beacons and 100 sensor nodes</td>
<td>30.352</td>
<td>9.673</td>
<td>9.541</td>
</tr>
<tr>
<td>20 beacons and 200 sensor nodes</td>
<td>25.475</td>
<td>9.284</td>
<td>9.025</td>
</tr>
<tr>
<td>100 beacons and 300 sensor nodes</td>
<td>24.229</td>
<td>8.732</td>
<td>8.645</td>
</tr>
</tbody>
</table>

Figure 5: Localization error comparison of different proportions of beacons in isotropic WSN.

Figure 6: Localization error comparison of different proportions of beacons in anisotropic WSN.
supposing that the number of sensor nodes in the whole WSN is 300, the proportion of beacons varies from 10% to 30%, communication range is 20, and DOI is set to 0.1∼0.3. It is clearly shown that the kernel-based localizationalgorithms P-LSVR and G-LSVR are more stable with better performance and smaller error.

The localization error of P-LSVR and G-LSVR is inversely proportional to the number of beacons in both isotropic WSN and anisotropic WSN. Owing to the sparse of SVR from the point of machine learning, the error of P-LSVR and G-LSVR is convergence. The error of DV-Hop is small in isotropic WSN, while it is big in anisotropic WSN. Therefore, DV-Hop is rarely used in practical applications.

4. Conclusions

This paper proposed a kernel-based range-free localization algorithm to transfer the problem of localization to the problem of kernel regression. The algorithm only needs simple approximate information and each sensor node estimates its own location distributivity. Radial basis kernel-based G-LSVR and polynomial-kernel-based P-LSVR proposed in this paper are compared with DV-Hop in both isotropic WSN and anisotropic WSN under different proportions of beacons, network scales, and disturbances of communication range. The kernel-based algorithms are better in localization accuracy and stability.

Data Availability

The data used to support the study are available from Dr. He via email, wenxiuhe@zjut.edu.cn.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Natural Science Foundation of Zhejiang Province under grant no. LGF21F020015.

References


