Semisupervised Location Awareness in Wireless Sensor Networks Using Laplacian Support Vector Regression

Jaehyun Yoo and H. Jin Kim

School of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, Republic of Korea

Correspondence should be addressed to H. Jin Kim; hjinkim@snu.ac.kr

Received 22 October 2013; Accepted 17 March 2014; Published 15 April 2014

Academic Editor: Hwa-Young Jeong

Copyright © 2014 J. Yoo and H. J. Kim. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Supervised machine learning has been widely used in context-aware wireless sensor networks (WSNs) to discover context descriptions from sensor data. However, collecting a lot of labeled training data in order to guarantee good performance requires much cost and time. For this reason, the semisupervised learning has been recently developed due to its superior performance despite using only a small amount of the labeled data. In this paper, we extend the standard support vector regression (SVR) to the semisupervised SVR by employing manifold regularization, which we call Laplacian SVR (LapSVR). The LapSVR is compared with the standard SVR and the semisupervised least square algorithm that is another recently developed semisupervised regression algorithm. The algorithms are evaluated for location awareness of multiple mobile robots in a WSN. The experimental results show that the proposed algorithm yields more accurate location estimates than the other algorithms.

1. Introduction

Location awareness is one of the most important issues for context-aware wireless sensor networks (WSNs) because many applications need to know where the data has been obtained. For example, in applications such as target tracking [1–5], network routing [6, 7], human-centric service [8–10], and monitoring system [11, 12], the measurement data is useless without location information of sensing device.

Machine learning methods have been adopted to learn location context from raw sensor data. The standard SVR learning [13, 14], manifold learning [15], and neural network learning [16, 17] can be employed to obtain precise estimate of the location. However, all these methods require a large amount of the labeled training data for high accuracy. Also, a significant effort such as cost, time, and human skill is needed in order to generate a large set of the labeled data. To make location-aware algorithm viable with a negligible effort, this paper adopts a semisupervised learning which uses a small number of the labeled data.

Based on the semisupervised learning framework suggested in [18], various algorithms such as [19–21] have been developed by defining different loss functions and regularization functions. In this work, we develop Laplacian support vector regression (LapSVR) which extends the standard SVR to the semisupervised learning framework. This approach uses the standard SVR to define the loss function and kernel-based regularization. However, for utilization of unlabeled data, the manifold regularization based on the concept of the graph Laplacian [22] is performed.

We evaluate the LapSVR to estimate locations of multiple mobile robots in the WSN. The deployed anchor nodes broadcast RF signal whose signal strength (RSSI) is measured by the mobile robots. The mobile robots independently localize themselves by using the trained regression model and current RSSI measurements. The LapSVR is compared with the standard SVR and the semisupervised Laplacian least square algorithm [23]. The location awareness performance is compared according to the number of labeled data, unlabeled data, and anchor nodes. From the experimental results, we confirm that the LapSVR estimates the location most accurately and is least sensitive to the variation of the number of the anchor nodes.

This paper is organized as follows. In Section 2, we provide details on Laplacian support vector regression algorithm. Section 3 describes the estimation process of mobile robots using the proposed algorithm. Section 4 reports empirical results with the description of the experimental setting. Finally, concluding remarks are given in Section 5.
2. Laplacian Support Vector Regression

This section describes a semisupervised regression approach named Laplacian support vector regression (LapSVR). We use the standard SVR algorithm to define the loss function and kernel-based regularization. Also, the graph Laplacian is employed for manifold regularization. We describe these ingredients of the formulation in the following.

2.1. Semisupervised Learning Framework. We define a set of \( l \) labeled samples \( \{x_i, y_i\}_{i=1}^l \) and a set of \( u \) unlabeled samples \( \{x_i\}_{i=l+1}^{l+u} \), where \( x_i \in X \subseteq \mathbb{R}^\xi \) and \( y_i \in \mathbb{R} \). Semisupervised learning aims to establish a mapping \( f : X \rightarrow \mathbb{R} \), by the following regularized minimization functional:

\[
f^* = \arg \min \left \{ \frac{1}{l} \sum_{i=1}^l V(x_i, y_i, f) + \gamma_A \|f\|^2_A + \gamma_I \|f\|^2_I \right \},
\]

where \( V \) is a loss function, \( \|f\|^2_A \) is the norm of the function in the reproducing kernel Hilbert space (RKHS), and \( \|f\|^2_I \) is the norm of the function in the low-dimensional manifold. \( \gamma_A \) and \( \gamma_I \) are the regularization weight parameters.

2.2. Decision Function. By the representer theorem [25], the solution of (1) can be defined as an expansion of kernel function over the labeled and the unlabeled data, given by

\[
f(x) = \sum_{i=1}^{l+u} \alpha_i K(x_i, x) + b
\]

where the kernel function \( K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \), where \( \phi(\cdot) \) is a nonlinear mapping to the RKHS.

Also, the regularization term \( \|f\|^2_A \) which is associated with the RKHS is defined as

\[
\|f\|^2_A = (\Phi \alpha)^T (\Phi \alpha) = \alpha^T K \alpha,
\]

where \( \Phi = [\phi(x_1), \ldots, \phi(x_{l+u})] \) and \( \alpha = [\alpha_1, \ldots, \alpha_{l+u}]^T \), and \( K \) is the \((l+u) \times (l+u)\) kernel matrix whose element is \( K(x_i, x_j) \).

2.3. Manifold Regularization. According to the manifold regularization (or manifold assumption), the data points are samples from a low-dimensional manifold which is embedded in a high-dimensional space. This is represented by the graph Laplacian and the kernel function:

\[
\|f\|^2_I = \frac{1}{l+u} \sum_{i=1}^{l+u} W_{ii} (f(x_i) - f(x_i))^2 \leq \frac{1}{l+u} \sum_{i=1}^{l+u} W_{ii} f^T L f,
\]

where \( L \) is the graph Laplacian, \( f = [f(x_1), \ldots, f(x_{l+u})]^T \), \( W \) is the adjacency matrix of the data graph, and \( D \) is the diagonal matrix given by \( D_{ii} = \sum_{j=1}^{l+u} W_{ij} \). In this paper, the edge weights \( W_{ij} \) are defined as Gaussian function of Euclidean distance, given by

\[
W_{ij} = \exp \left( \frac{-\|x_i - x_j\|^2}{\sigma_w^2} \right),
\]

where \( \sigma_w^2 \) is the kernel width parameter.

2.4. Loss Function for Regression. We employ the \( \varepsilon \)-insensitive loss function:

\[
V(x_i, y_i, f) = \begin{cases} 0 & \text{if } |f(x_i) - y_i| < \varepsilon \\|f\|_I^2 & \text{otherwise} \end{cases}
\]

Also, slack variables \( \xi_i, \xi_i^* \) are used with the \( \varepsilon \)-insensitive loss function in order to cope with infeasible constraints of the optimization problem that uses only the \( \varepsilon \) constraint, described in [24]. Figure 1 depicts an example of one-dimensional linear regression function with \( \varepsilon \)-tube. The samples outside the \( \varepsilon \)-tube are called support vectors. Slack variables \( \xi_i, \xi_i^* \), which establish trade-off between model complexity and the support vectors, are determined by solving the optimization problem in the following section.

2.5. Complete Formulation. In this subsection, the complete formulation of the LapSVR is given based on the ingredients described in Sections 2.1–2.4. Substituting (2)–(6) into (1) with the slack variables \( \xi_i, \xi_i^* \), the primal problem is defined as

\[
\min_{\alpha \in \mathbb{R}^{l+u}, \xi, \xi^* \in \mathbb{R}^l} \frac{l}{l+u} \sum_{i=1}^{l+u} (\xi_i + \xi_i^*) + \gamma_A \alpha^T K \alpha + \frac{1}{l+u} \sum_{i=1}^{l+u} \gamma_I \alpha^T K L K \alpha
\]

subject to:

\[
\sum_{j=1}^{l+u} \alpha_j K(x_i, x_j) - b \leq \varepsilon + \xi_i^*,
\]

\[
\sum_{j=1}^{l+u} \alpha_j K(x_i, x_j) + b - y_i \leq \varepsilon + \xi_i^*,
\]

\[
\xi_i, \xi_i^* \geq 0, \quad i = 1, \ldots, l.
\]

After the introduction of four sets of \( l \) multipliers \( \beta, \beta^*, \eta, \eta^* \), the Lagrangian \( H \) can be obtained in the following:

\[
H(\alpha, \beta, \eta, \xi, \eta^*) = \frac{1}{2} \alpha^T (2\gamma_A K + \frac{2}{l+u} \gamma_I K L K) \alpha + \frac{1}{l+u} \sum_{i=1}^{l+u} (\xi_i + \xi_i^*)
\]
\[
- \sum_{i=1}^{l} \beta_i \left( \epsilon + \xi_i - y_i + \sum_{j=1}^{l+u} \alpha_j k(x_i, x_j) + b \right)
\]
\[
- \sum_{i=1}^{l} \beta_i^* \left( \epsilon + \xi_i^* + y_i - \sum_{j=1}^{l+u} \alpha_j k(x_i, x_j) - b \right)
\]
\[
- \frac{1}{l} \sum_{i=1}^{l} (\eta_i \xi_i + \eta_i^* \xi_i^*) .
\]

In order to convert the primal problem to the dual representation, we take derivatives

\[
\frac{\partial H}{\partial b} = \sum_{i=1}^{l} \left( \beta_i - \beta_i^* \right) = 0,
\]
\[
\frac{\partial H}{\partial \xi_i} = \frac{1}{l} - \frac{\beta_i^*}{1} - \frac{1}{l} \eta_i = 0.
\]

Using the above equations, we can rewrite the Lagrangian as a function of \( \alpha, \beta, \) and \( \beta^* \) only, given by

\[
H(\alpha, \beta, \beta^*) = \frac{1}{2} \alpha^T \left( 2y_A K + \frac{2}{(l+u)^2} \delta KLK \right) \alpha
\]
\[
- \sum_{i=1}^{l} \left( \beta_i - \beta_i^* \right) \sum_{j=1}^{l+u} \alpha_j k(x_i, x_j) ^* \right) - \epsilon \sum_{i=1}^{l} \left( \beta_i + \beta_i^* \right) + \sum_{i=1}^{l} y_i \left( \beta_i - \beta_i^* \right)
\]
\[
- \frac{1}{2} \alpha^T \left( 2y_A K + \frac{2}{(l+u)^2} \delta KLK \right) \alpha - \alpha^T K I_L^T B
\]
\[
- \epsilon \sum_{i=1}^{l} \left( \beta_i + \beta_i^* \right) + \sum_{i=1}^{l} y_i \left( \beta_i - \beta_i^* \right) ,
\]

where \( B = [\beta_1 - \beta_1^*, \ldots, \beta_l - \beta_l^*]^T \) and \( I_L = [I_{l \times l}, 0_{l \times u}] \) where \( I_{l \times l} \) is the \( l \times l \) identity matrix and \( 0_{l \times u} \) is the \( l \times u \) zero matrix.

Taking derivatives with respect to \( \alpha \), we obtain

\[
\frac{\partial H}{\partial \alpha} = \left( 2y_A K + \frac{2}{(l+u)^2} \delta KLK \right) \alpha - \alpha^T K I_L^T B = 0.
\]

From (11), a direct relationship between \( \alpha \) and \( B \) is obtained:

\[
\alpha = \left( 2y_A K + \frac{2}{(l+u)^2} \delta KLK \right)^{-1} I_L^T B ,
\]

where \( I \) is the \((l+u) \times (l+u)\) identity matrix. Substituting (12) back into Lagrangian (10), we arrive at the convex optimization problem:

\[
\max_{\beta \in \mathbb{R}^l, \beta^* \in \mathbb{R}^l} -\frac{1}{2} \beta^T J_L K \left( 2y_A I + \frac{2}{(l+u)^2} \delta KLK \right)^{-1} J_L^T B - \frac{1}{l} \sum_{i=1}^{l} \left( \beta_i + \beta_i^* \right) + \sum_{i=1}^{l} y_i \left( \beta_i - \beta_i^* \right)
\]

subject to:

\[
\sum_{i=1}^{l} (\beta_i - \beta_i^*) = 0
\]
\[
0 \leq \beta_i^*, \beta_i^* \leq \frac{1}{l}, \quad i = 1, \ldots, l.
\]

The optimal solution \( B^* \) of the above quadratic problem is linked to the optimal \( \alpha^* \) in (12). Then, the optimal regression function \( f^*(x) \) can be obtained by substituting \( \alpha^* = [\alpha_1^*, \ldots, \alpha_{l+u}^*]^T \) into (2).

2.6. Computation of \( b \). Now, the remaining work to establish \( f^*(x) \) is computation of the bias term \( b \). As described in [24], \( b \) is calculated by using Karush-Kuhn-Tucker (KKT) conditions which state that the product between dual variables and constraints has to be zero at the point of the solution. From (8), we obtain

\[
\beta_i \left( \epsilon + \xi_i - y_i + \sum_{j=1}^{l+u} \alpha_j k(x_i, x_j) + b \right) = 0,
\]
\[
\beta_i^* \left( \epsilon + \xi_i^* + y_i - \sum_{j=1}^{l+u} \alpha_j k(x_i, x_j) - b \right) = 0,
\]

for any labeled data; that is, \( i \in \{1, \ldots, l\} \).

Equations (14) lead to three conclusions. First, only the samples \((x_i, y_i)\) with corresponding \( \beta_i^* = 1/l \) lie outside the \( \epsilon \)-tube. Second, \( \beta_i = 0 \); that is, a set of dual variables \( \beta_i^* \) is never simultaneously nonzero. Finally, for \( \beta_i \in (0, 1/l) \), we have

\[
\xi_i = 0, \quad \left( \epsilon + \xi_i - y_i + \sum_{j=1}^{l+u} \alpha_j k(x_i, x_j) + b \right) = 0,
\]

and for \( \beta_i^* \in (0, 1/l) \) we have

\[
\xi_i^* = 0, \quad \left( \epsilon + \xi_i^* + y_i - \sum_{j=1}^{l+u} \alpha_j k(x_i, x_j) - b \right) = 0.
\]

Hence, we can compute \( b \) using any data point satisfying \( \beta_i \in (0, 1/l) \) or \( \beta_i^* \in (0, 1/l) \), given by

\[
b = \begin{cases} 
  y_i - \sum_{j=1}^{l+u} \alpha_j k(x_i, x_j) + \epsilon & \text{for } \beta_i \in \left(0, \frac{1}{l}\right) \\
  y_i - \sum_{j=1}^{l+u} \alpha_j k(x_i, x_j) - \epsilon & \text{for } \beta_i^* \in \left(0, \frac{1}{l}\right).
\end{cases}
\]

In reality, the value of \( b \) can be slightly different depending on which data point is used in (17). Averaging over all \( b \) obtained from support vectors may improve robustness.

3. RSSI-Based Location Awareness

In RSSI-based location awareness, mobile nodes of unknown locations collect the RSSI measurements emitted by \( n \) anchor nodes. We assume that the anchor nodes are stationary with known location in \( \chi \in \mathbb{R}^2 \) where \( \chi \) is the area of interest. Each
For accuracy analysis, the true locations of the mobile robots only the RSSI measurements, given by the mobile node at the specific location the algorithm of one mobile node. In the training phase, we put the mobile node independent of the others, we will describe unlabeled training data sets. Because each robot estimates its own location and obtain the RSSI values \( Z_i = [z_{i1}, \ldots, z_{ip}, \ldots, z_{in}]^T \in \mathbb{R}^n \), where \( z_{ix} \) and \( z_{iy} \) are the \( x \)- and \( y \)-coordinate of the mobile node, and \( z_{ip} \) is the RSSI value from the \( p \)th anchor node for \( i = 1, \ldots, l \). The total \( l \) labeled training data sets to learn the \( x \) and \( y \) are defined as \( \mathcal{L}_x = [Z_i, s_x]_{i=1}^l \) and \( \mathcal{L}_y = [Z_i, s_y]_{i=1}^l \), respectively. On the other hand, \( u \) unlabeled data consists of only the RSSI measurements, given by \( \mathcal{U} = [Z_i]_{i=1}^u \).

In the test phase, with the trained semisupervised SVR model, the mobile node determines its own location \( \hat{s}(t) = [\hat{s}_x(t), \hat{s}_y(t)]^T \) using the RSSI set \( Z(t) \in \mathbb{R}^n \) measured at time \( t \).

We summarize the detail of the semisupervised SVR for RSSI-based location awareness Algorithms 1 and 2.

**Algorithm 1** (training phase). Consider the following.

**Step 1.** Collect the training data set \( \mathcal{L}_x, \mathcal{L}_y, \) and \( \mathcal{U} \).

**Step 2.** Obtain the kernel matrix \( K \) in (2).

**Step 3.** Build weight matrix \( W \) and the graph Laplacian \( L = D - W \) in (4).

**Step 4.** Choose the regularization weights \( \gamma_x \) and \( \gamma_y \) in (1).

**Step 5.** Solve the QP problem twice in order to obtain \( \alpha^*_x \) and \( \alpha^*_y \) in (13).

**Step 6.** Compute \( b_x \) and \( b_y \) in (17).

**Algorithm 2** (test phase). Consider the following.

**Step 1.** At time \( t \), the mobile node collects the RSSI measurements \( Z(t) \) from the \( n \) anchor nodes.

**Step 2.** Obtain the regression estimation with \( \hat{s}_x(t) = \sum_{i=1}^{l+u} \alpha^*_x K(Z_i, Z(t)) + b_x \) and \( \hat{s}_y(t) = \sum_{i=1}^{l+u} \alpha^*_y K(Z_i, Z(t)) + b_y \). Then decide the location estimate \( \hat{s}(t) = [\hat{s}_x(t), \hat{s}_y(t)]^T \).

### 4. Experimental Results

In the experimental setup, the total of thirteen anchor nodes using IEEE 802.15.4 and TinyOS is deployed. The nodes use commercial CC2420 radio chip which operates in the 2.4 GHz ISM band with a data rate of 256 kbps and provides a received signal strength indicator (RSSI) for each received packet. Two mobile robots, which are equipped with one receiver node, respectively, estimate their own position using the RSSI measurements obtained from the anchor nodes distributed in the \( 3 \times 4 \) m workspace, shown in Figure 2. For accuracy analysis, the true locations of the mobile robots are measured by the Vicon Motion System [26] composed of eight cameras that track reflective markers attached to the mobile robots.

For obtaining labeled training data, we collect the RSSI data sets of the mobile robot placed at the different locations, repeatedly. On the other hand, the unlabeled data sets are obtained as we let the mobile robots move automatically and collect the RSSI measurements. Figure 3 is an illustration of the obtained data set. Note that the real locations of the unlabeled data are not included in the unlabeled data set, although Figure 3 shows the locations of the unlabeled data.

#### 4.1. Characteristics of RSSI

Radio signal is easily influenced by environmental noise. Also, its measured value affects the performance of the location awareness. Here, we examine the characteristic of RSSI measurement.

We record RSSI measurements of one moving receiver, emitted by the fixed thirteen anchor nodes. The distance between the anchor nodes and receiver varies from 0.2 m to 3 m. Figure 4 shows the boxplot of the obtained measurements. The central red bar is the median and the edges
We use as the kernel function in (2) the radial basis function (RBF) given by
\[ K(x_i, x_j) = \exp(-\|x_i - x_j\|^2/\sigma^2), \]
where \( \sigma \) is the kernel width parameter. In (4), the graph Laplacian consisted of \((l + u)\) samples using \(k\)-nearest neighbors, and the edge weights \(W_{ij}\) are defined as Gaussian function of Euclidean distance. In all experiments, we set \( \sigma = 0.5 \) and \( k = 10 \).

4.2. Model and Parameter Setting. We use as the kernel function in (2) the radial basis function (RBF) given by
\[ K(x_i, x_j) = \exp(-\|x_i - x_j\|^2/\sigma^2), \]
where \( \sigma \) is the kernel width parameter. In (4), the graph Laplacian consisted of \((l + u)\) samples using \(k\)-nearest neighbors, and the edge weights \(W_{ij}\) are defined as Gaussian function of Euclidean distance. In all experiments, we set \( \sigma = 0.5 \) and \( k = 10 \).

4.3. Performance. The proposed algorithm is compared with two other algorithms. First, the standard SVR is conventionally employed for the location awareness problem as in [14, 27]. Second, Laplacian least square (LapLS) is a recently developed semisupervised regression algorithm and its application to the location estimation can be found in [23]. LapLS solves optimization using graph Laplacian and loss function in the least square that does not have any constraint for the optimization. Therefore, an optimal solution can be obtained without a numerical process.

Figure 6 illustrates the true and estimated trajectories of the LapSVR, standard SVR, and LapLS algorithms using the obtained data set in Figure 3, where the LapSVR gives the best tracking performance. In the following, we show three kinds of performance criteria to compare the algorithms.

First, Figure 7 shows the average location estimation errors according to the number of the labeled data, which are calculated after two mobile robots move along their own trajectories. We confirm that the proposed algorithm gives the best performance over the other two algorithms.

Second, location estimation error of the LapSVR according to the number of the unlabeled data is described in Figure 8. From this result, we confirm that the unlabeled data gives a positive effect when the labeled data is insufficient (less than 50 labeled data in Figure 8). The excessive unlabeled data (more than 50 unlabeled data in Figure 8) does not improve the performance.

Third, we examine the sensitivity of the location estimation error with respect to the number of the anchor nodes. We prepare the set 2 consisting of all the deployed nodes. The set 1 is formed with the nodes whose numbers are from 1 to 9 depicted in Figure 2. The set 0 consists of the nodes whose numbers are 1, 3, 5, 7, and 9. In this experiment, we use 40 labeled data and 200 unlabeled data. As the result in Figure 9, the LapSVR algorithm causes the smallest growth of error as the number of the anchor nodes decreases in comparison with the other algorithms.

5. Conclusion

This paper proposes a semisupervised regression algorithm named Laplacian SVR by combining the core concepts of the standard SVR and a manifold regularization framework. The proposed algorithm learns the relation mapping function between the observation space and the physical space using a small number of the labeled data and a large number of the unlabeled data which can often be easily acquired more than the labeled data. The suggested algorithm is evaluated for location estimation of mobile robots in the wireless sensor network. In our multirobot system, the input of the learner is defined as the RSSI observation set and the output is the physical location. The experimental results show that the proposed algorithm gives the most precise performance and
**Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

**Acknowledgments**

This research was financially supported by a grant to MEMS Research Center for National Defense funded by Defense Acquisition Program Administration and the National Research Foundation of Korea (NRF) Grant funded by the Ministry of Science, ICT & Future Planning (MSIP) (no. 2009-0083495).

**References**

[1] J. Araújo, H. Sandberg, and K. H. Johansson, “Experimental validation of a localization system based on a heterogeneous sensor network,” in *Proceedings of the 7th Asian Control Conference (ASCC '09)*, pp. 465–470, August 2009.
[2] L. Briñón-Arranz and L. Schenato, “Consensus-based source-seeking with a circular formation of agents,” in Proceedings of the 12th European Control Conference (ECC ’13), July 2013.

[3] J. H. Yoo, W. Kim, and H. J. Kim, “Event-driven Gaussian process for object localization in wireless sensor networks,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems: Celebrating 50 Years of Robotics (IROS ’11), pp. 2790–2795, September 2011.

[4] S. Oh, “Using an adaptive search tree to predict user location,” Journal of Information Processing Systems, vol. 8, no. 3, pp. 437–444, 2012.

[5] Q. L. Liu and D. H. Oh, “Performance evaluation of multi-hop communication based on a mobile multi-robot system in a subterranean laneway,” Journal of Information Processing Systems, vol. 8, no. 3, pp. 471–482, 2012.

[6] X. Zhou, Y. Ge, X. Chen, Y. Jing, and W. Sun, “A distributed cache based reliable service execution and recovery approach in MANETs,” Journal of Convergence, vol. 3, no. 1, pp. 298–305, 2012.

[7] X. Li, N. Mitton, I. Lille, A. Nayak, and I. Stojmenovic, “Achieving Load Awareness in position-based wireless ad hoc routing,” Journal of Convergence, vol. 3, no. 3, 2012.

[8] V. Viswanathan and I. Krishnamurthi, “Finding relevant semantic association paths through user-specific intermediate entities,” Human-Centric Computing and Information Sciences, vol. 2, no. 9, 2012.

[9] J. Ahn and R. Han, “An indoor augmented-reality evacuation system for the Smartphone using personalized Pedometry,” Human-Centric Computing and Information Sciences, vol. 2, no. 18, 2012.

[10] T. Teraoka, “Organization and exploration of heterogeneous personal data collected in daily life,” Human-centric Computing and Information Sciences, vol. 2, no. 1, 2012.

[11] B. J. Oommen, A. Yazidi, and O. C. Granmo, “An adaptive approach to learning the preferences of users in a social network using weak estimators,” Journal of Information Processing Systems, vol. 8, no. 2, 2012.

[12] J. Kee and Y. Ng, “Ubiquitous healthcare: healthcare systems and applications enabled by mobile and wireless technologies,” Journal of Convergence, vol. 3, no. 2, 2012.

[13] K. Yamano, K. Tanaka, M. Hirayama, E. Kondo, Y. Kimuro, and M. Matsumoto, “Self-localization of mobile robots with RFID system by using support vector machine,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’04), pp. 3756–3761, October 2004.

[14] X. Nguyen, M. Jordan, and B. Sinopoli, “A kernel-based learning approach to ad hoc sensor network localization,” IEEE Transactions on Sensor Networks, vol. 1, no. 1, pp. 134–152, 2005.

[15] N. Patwari and A. O. Hero III, “Manifold learning algorithms for localization in wireless sensor networks,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 857–860, May 2004.

[16] U. Ahmad, A. Gavrilov, U. Nasir, M. Iqbal, S. J. Cho, and S. Lee, “In-building localization using neural networks,” in Proceedings of the IEEE International Conference on Engineering of Intelligent Systems (ICEIS ’06), April 2006.

[17] M. Stella, M. Russo, and D. Begusic, “Location determination in indoor environment based on RSS fingerprinting and artificial neural network,” in Proceedings of the 9th International Conference on Telecommunications (ConTEL ’07), pp. 301–305, June 2007.

[18] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: a geometric framework for learning from labeled and unlabeled examples,” Journal of Machine Learning Research, vol. 7, pp. 2399–2434, 2006.

[19] J. Weston, F. Ratle, and R. Collobert, “Deep learning via semi-supervised embedding,” in Proceedings of the 25th International Conference on Machine Learning, pp. 1168–1175, July 2008.

[20] H. Shin, N. J. Hill, and G. Ratsch, “Graph-based semi-supervised learning with sharper edges,” Proceedings of the 5th European Conference on Machine Learning, 2006.

[21] S. Melacci and M. Belkin, “Laplacian support vector machines trained in the primal,” Journal of Machine Learning Research, vol. 12, pp. 1149–1184, 2011.

[22] O. Chapelle, B. Scholkopf, and A. Zien, Semi-Supervised Learning, MIT Press, Cambridge, UK, 2006.

[23] B. Yang, J. Xu, Y. Yang, and M. Li, “Localization algorithm in wireless sensor networks based on semi-supervised manifold learning and its application,” Cluster Computing, vol. 13, no. 4, pp. 435–446, 2010.

[24] A. J. Smola and B. Schlkopf, “A tutorial on support vector regression,” NeuroCOLT2 Technical Report Series NC2-TR-1998-030, 1998.

[25] B. Schölkopf, R. Herbrich, and A. J. Smola, “A generalized representer theorem,” in Proceedings of the Annual Conference on Computational Learning Theory (COLT ‘01), pp. 416–426, 2001.

[26] The VICON website, 2013, http://www.vicon.com/.

[27] D. A. Tran and T. Nguyen, “Localization in wireless sensor networks based on support vector machines,” IEEE Transactions on Parallel and Distributed Systems, vol. 19, no. 7, pp. 981–994, 2008.