Semi-Supervised Learning for Mobile Robot Localization using Wireless Signal Strengths

Jaehyun Yoo and Karl H. Johansson

***Abstract*—This paper proposes a new semi-supervised machine learning for localization. It improves localization efficiency by reducing efforts needed to calibrate labeled training data by using unlabeled data, where training data come from received signal strengths of a wireless communication link. The main idea is to treat training data as spatio-temporal data. We compare the proposed algorithm with the state-of-art semi-supervised learning methods. The algorithms are evaluated for estimating the unknown location of a smartphone mobile robot. The experimental results show that the developed learning algorithm is the most accurate and robust to the varying amount of training data, without sacrificing the computation speed.**

I. Introduction

Indoor localization is important due to the need for location information where GPS is not available. Fortunately, prevalence of wireless access points located in commercial buildings, homes, and public places helps in developing wifi-based localization, without installation of positioning devices. However, the wifi received signal strength (RSS) as a function of distance between a receiver and a transmitter is non-linear and varying due to interference of other radio signals and obstacles. In order to overcome this problem, the learning-based localization methods have been proposed by training the non-linear RSS data [1]–[4].

In order to implement the learning-based localization, it is typically required to calibrate training points manually. Recently much effort has been concentrated on reducing the calibration cost [5]–[7]. Semi-supervised learning is one of the the efficient-learning localization in that it uses both labeled and unlabeled data for learning. Benefit of semi-supervised learning is that unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy. This method can bring big advancement to learning-based indoor localization because we can reduce manual operation to collect labeled training data, while unlabled data can be easily collected by recording wifi signal strength without position information. By using a large amount of unlabeled data and a small amount of labeled data, the semi-supervised learning algorithm improves efficiency. The challenge is to maintain accuracy despite using a small set of labeled training data.

*This work has been supported in part by the Knut and Alice Wallenberg Foundation, the Swedish Research Council and the Swedish Foundation for Strategic Research within the SSF-NRF Sweden-Korea research program.

J. Yoo and K. H. Johansson are with the ACCESS Linnaeus Center and the School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, Emails: {jaehyun,kallej@kth.se}

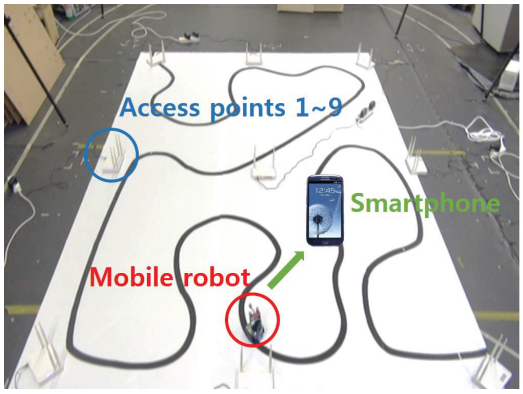
978-1-5090-6299-7/17/$31.00 © 2017 IEEE

Many existing semi-supervised learning algorithms [8]–[11] utilize unlabeled data only for manifold regularization, which captures the intrinsic geometric structure of training data. More sophisticated usage of unlabeled data is pseudolabelling where unlabeled data are artificially labeled and then used for learning the model as if the data are labeled. In [12]–[17], pseudolabels are iteratively updated, and then the model is learned by the final pseudolabels. Most pseudolabelling methods [13]–[17] have focused on classification problems, while only few regression problems such as localization have been reported [12].

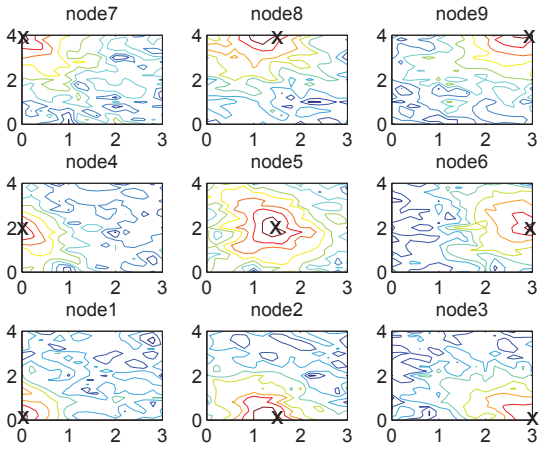
This paper proposes a new semi-supervised learning algorithm. We separate the pseudolabelling process and the learning process. First, in the pseudolabelling process, we employ the Laplacian Embedded Regression Least Square (LapERLS) [12] framework, which propagates the labeled data to the whole data set (both unlabeled and labeled data). Based on this framework, we add time-series regularization [18] where training data become time-stamped by sorting the data set in chronological order. This is reasonable for a smoothly moving robot because training data are collected as a time series. Therefore, pseudolabels from the time-series learning become more sophisticated by considering both spatial and temporal relationships than the conventional pseudolabels that implies the spatial aspect only.

Because pseudolabels are artificial, we cannot trust the pseudolabeled data as much as the labeled data. Therefore, it is desirable to limit the reliance on pseudolabeled data. Altough the existing semi-supervised methods [8], [12] have addressed this issue, many parameters in [8], [12] that are coupled to both labeled and unlabeled (or pseudolabeled) terms cause algorithm complexity. Furthermore, the balancing becomes unfeasible when amount of training data vary because the parameters are optimized with respect to a fixed number of labeled and unlabeled data. In worst case, the unlabeled data may not be helpful at all. We solve this imbalance problem by using the optimization framework Laplacian Least Square (LapLS) that combines manifold regularization and transductive support vector machine (TSVM) [13]. In this learning process, two decoupled balancing parameters are individually weighted to labeled and pseudolabeled terms. Therefore, it is easy to handle the varying amount of labeled and unlabeled data.

We evaluate the proposed algorithm on a problem to estimate the unknown location of a smartphone robot, by comparing with recently-developed semi-supervised algorithms, LapERLS, semi-supervised least square support vector regression (SSL) [19], and semi-supervised colocalization (SSC)



(a) Experimental setup in 3m × 4m



(b) Wifi signal strength distribution of each of 9 access points

Fig. 1: (a) Experimental setup with 9 wifi access points and one smartphone-based mobile robot; (b) Wifi signal strength distributions of 9 APs. The APs are placed near the peeks of each distribution.

[8]. We empirically show that our algorithm estimates the location more accurately with fewer labeled data, by efficiently exploiting unlabeled data. Also, when we test the localization with respect to varying number of unlabeled data, our algorithm gives the best localization performance and the lowest variability about randomly picked training data. For varying number of training data from 10 to 100, computation time of the proposed algorithm is slightly longer than LapERLS and SSL (at most 0.2 sec difference), while it is 2–6 times faster than SSC.

This paper is organized as follows. Section II defines wifi-based localization using a mobile robot. Section III presents existing semi-supervised learning algorithms. Section IV describes the proposed algorithm. Section V reports empirical results. Finally, concluding remarks are given in Section VI.

II. Wifi-based Indoor Localization

Fig. 1(a) shows the localization setup where 9 wifi access points are deployed in the workspace. Although it seems the dense AP deployment in area, the signal strength propagation of all APs is adjusted to cover the room complementarily, as shown in Fig. 1(b). Fig. 1(b) shows wifi RSSs of each of the 9 APs which are located at the mark 'x' in the figures. It is shown that the highest peek is located at AP's true location and RSS decreases according to distance (say rough Gaussian distribution). A mobile robot (Wheelphone produced by GCtronic) is controlled by a smartphone (Galaxy S3 produced by Samsung Electronics) in order to track a designated path. The smartphone receives wifi received signal strength (RSS) from the access points (APs). For accuracy analysis, the true location of the mobile robot is measured by Vicon motion capture system.

Labeled RSS training data are obtained by placing the smartphone at different locations. Let us define the wifi observation set as $x_i = \{z_{i1}, \ldots, z_{in}\} \in \mathcal{R}^n$ from n APs ($n = 9$ in this paper), where z_{ij} ($1 \le j \le n$) is a scalar decibel measurement of the j-th access point corresponding to the robot's location of $(y_{\mathrm{X}i}, y_{\mathrm{Y}i}) \in R^2$. Total of l labeled training data are given by $\{x_i\}_{i=1}^{l}$ with $x_i \in X \subseteq \mathcal{R}^n$, and $\{y_{\mathrm{X}i}\}_{i=1}^{l}, \{y_{\mathrm{Y}i}\}_{i=1}^{l}$. The unlabeled data set $\{x_i\}_{i=l+1}^{l+u}$ consists of only the RSS measurements, without position information. Labeled and unlabeled data are obtained as we let the mobile robot move autonomously over the area.

The training phase builds separate mappings $f_{\mathrm{X}} : X \to \mathcal{R}$ and $f_{\mathrm{Y}} : X \to \mathcal{R}$ which denote relationships between wifi signal strength and location of the smartphone robot, using the labeled training data $\{(x_i, y_{\mathrm{X}i})\}_{i=1}^{l}$ and $\{(x_i, y_{\mathrm{Y}i})\}_{i=1}^{l}$, respectively, and the unlabeled data $\{x_i\}_{i=l+1}^{l+u}$. Because the models f_{X} and f_{Y} are learned independently, we omit the subscripts of f_{X}, f_{Y}, and y_{X}, y_{Y}, for simplification.

III. Semi-Supervised Learning

In this section, we first describe the framework of laplacian semi-supervised learning in III-A. Then, we briefly review the extended semi-supervised algorithms, namely Laplacian least square SVR (LapLS) in III-B and Laplacian embedded regression least square (LapERLS) in III-C. Key ideas from these algorithms will be used for our proposed algorithm in the next Section IV.

A. Basic Semi-Supervised Learning

Given a set of l labeled samples $\{(x_i, y_i)\}_{i=1}^{l}$ and a set of u unlabeled samples $\{x_i\}_{i=l+1}^{l+u}$, Laplacian semi-supervised learning aims to establish a mapping f by the following regularized minimization functional:

$$f^* = \arg\min_{f \in \mathcal{H}_k} C \sum_i V(x_i, y_i, f) + \gamma_A \|f\|_A^2 + \gamma_I \|f\|_I^2, \quad (1)$$

where V is a loss function, $\|f\|_A^2$ is the norm of the function in the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_k$, $\|f\|_I^2$ is the norm of the function in the low dimensional manifold, and C, γ_A, γ_I are the regularization weight parameters.

The solution of (1) is defined as an expansion of kernel function over the labeled and the unlabeled data, given by

$$f(x) = \sum_{i=1}^{l+u} \alpha_i K(x_i, x) + b \tag{2}$$

with the bias term b and the kernel function $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, where $\phi(\cdot)$ is a nonlinear mapping to RKHS.

The regularization term $\|f\|_A^2$ associated to RKHS is defined as

$$\|f\|_A^2 = (\Phi\alpha)^T(\Phi\alpha) = \alpha^T K \alpha, \tag{3}$$

where $\Phi = [\phi(x_1), \ldots, \phi(x_{l+u})]$, $\alpha = [\alpha_1, \ldots, \alpha_{l+u}]^T$, and K is the $(l+u) \times (l+u)$ kernel matrix whose element is K_{ij}. We adopt Gaussian kernel given by

$$K_{ij} = K(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2 / \sigma_k^2\right), \tag{4}$$

where σ_k^2 is the kernel width parameter.

According to the manifold regularization, datapoints are samples obtained from a low-dimensional manifold embedded in a high-dimensional space. This is represented by the graph Laplacian [20]:

$$\begin{aligned} \|f\|_I^2 &= \frac{1}{(l+u)^2} \sum_{i=1}^{l+u} \sum_{j=1}^{l+u} W_{ij} \left(f(x_i) - f(x_j)\right)^2, \\ &= \frac{1}{(l+u)^2} \mathbf{f}^T L \mathbf{f}, \end{aligned} \tag{5}$$

where L is the normalized graph Laplacian given by $L = D^{-1/2}(D - W)D^{-1/2}$, $\mathbf{f} = [f(x_1), \ldots, f(x_{l+u})]^T$, 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 general, the edge weights W_{ij} are defined as Gaussian function of Euclidean distance, given by

$$W_{ij} = \exp\left(-\|x_i - x_j\|^2 / \sigma_w^2\right), \tag{6}$$

where σ_w^2 is the kernel width parameter.

Minimizing $\|f\|_I^2$ is equivalent to penalizing the rapid changes of the regression function evaluated between two data points. Therefore, $\gamma_I \|f\|_I^2$ in (1) controls the smoothness of the data geometric structure.

B. Laplacian Least Square (LapLS)

We produce LapLS by combining manifold regularization (5) and transductive SVM (TSVM) [13]. In LapLS, the loss function V in (1) is defined by

$$V(x_i, y_i, f) = e_i = y_i - \left(\sum_{i=1}^{l+u} \alpha_i K(x_i, x) + b\right). \tag{7}$$

LapLS finds optimal parameters α, b, and the labels $y_1^*, \ldots, y_u^*$ of the unlabeled data when regularization parameters C and C^* are given:

$$\begin{aligned} \min_{\alpha, e, e^*, b, y_1^*, \ldots, y_u^*} & \frac{C}{2} \sum_{i=1}^{l} e_i^2 + \frac{C^*}{2} \sum_{j=1}^{u} (e_j^*)^2 \\ & + \gamma_A \alpha^T K \alpha + \gamma_I \alpha^T K L K \alpha \\ \text{subject to : } & y_i - \sum_{k=1}^{l+u} \alpha_k K_{ik} - b - e_i = 0, \quad i = 1, \ldots, l. \\ & y_j^* - \sum_{k=1}^{l+u} \alpha_k K_{jk} - b - e_j^* = 0, \quad j = 1, \ldots, u. \end{aligned} \tag{8}$$

Optimizing the problem of (8) with respect to all $y_1^*, \ldots, y_u^*$ is combinatorial problem [13]. In order to find the solution, we have to search over all possible 2^u labels of the unlabeled data. Therefore, this method is not useful when a large amount of the unlabeled data is applied.

C. Laplacian Embedded Regularized Least Square (LapERLS) [12]

LapERLS introduces an intermediate decision variable $g \in \mathcal{R}^{(l+u)}$ and additional regularization parameter γ_C into the laplacian semi-supervised framework (1), as follows:

$$\begin{aligned} \min_{f \in \mathcal{H}_k, g \in \mathcal{R}^{(l+u)}} & C \sum_{i=1}^{l+u} V(x_i, g_i, f) + \gamma_C \sum_{i=1}^{l} (g_i - y_i)^2 \\ & + \gamma_A \|f\|_A^2 + \gamma_I \|g\|_I^2. \end{aligned} \tag{9}$$

The optimization problem of (9) enforces the intermediate decision variable g to be close to the labeled data and also to be smooth with respect to the graph manifold.

Loss function is given by:

$$V(x_i, g_i, f) = \xi_i = g_i - \left(\sum_{j=1}^{l+u} \alpha_j K(x_i, x_j)\right). \tag{10}$$

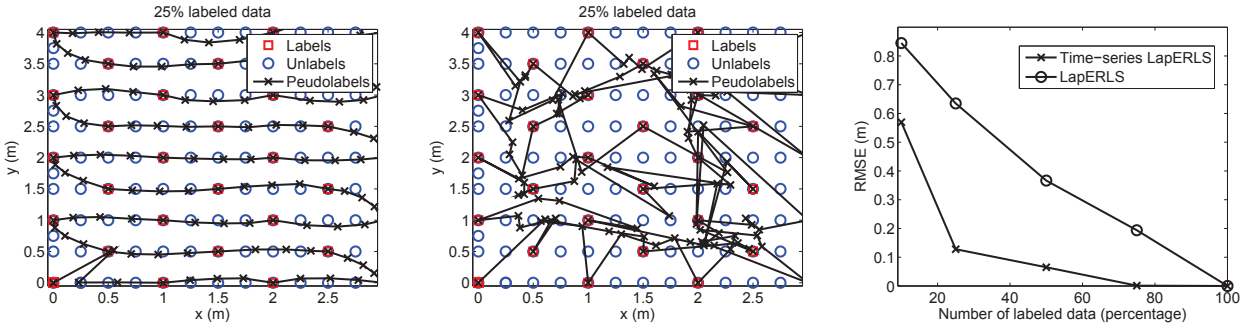
After reorganizing the terms in (9) with respect to manifold regularization and decision function and corresponding parameter, the primal optimization problem is as follows:

$$\begin{aligned} \min_{\alpha, g, \xi \in \mathcal{R}^{(l+u)}} & \frac{C}{2} \sum_{i=1}^{l+u} \xi_i^2 + \alpha^T K \alpha \\ & + \frac{1}{2}(g - y)^T \Lambda (g - y) + \frac{1}{2} \mu g^T L g \\ \text{subject to : } & \xi_i = g_i - \sum_{k=1}^{l+u} \alpha_k K_{ik}, \quad i = 1, \ldots, l+u, \end{aligned} \tag{11}$$

where Λ is a diagonal matrix of trade-off parameters with $\Lambda_{ii} = \lambda$ if x_i is a labeled data point, and $\Lambda_{ii} = 0$ if x_i is unlabeled, $y = [y_1, \ldots, y_l, 0, \ldots, 0]^T \in \mathcal{R}^{(l+u)}$, and C, λ, μ are tuning parameters.

Also, dual formulation of (11) is given by:

$$\min_{\beta \in \mathcal{R}^{(l+u)}} \frac{1}{2} \beta^T \tilde{Q} \beta + \beta^T \tilde{y}, \tag{12}$$



(a) Pseudolabels of time-series LapERLS (b) Pseudolabels of standard LapERLS (c) RMSE of pseudolabels between time-series LapERLS and standard LapERLS

Fig. 2: Accuracy of pseudolabels obtained by (a) time-series LapERLS (19); (b) standard LapERLS (13); (c) RMSE of each pseudolabel according to the amount of labeled data.

where

$$\begin{aligned} \tilde{Q} &= K + (\Lambda + \mu L)^{-1}, \\ \tilde{y} &= (\Lambda + \mu L)^{-1}\Lambda y, \\ \beta &= -\alpha. \end{aligned} \tag{13}$$

The main characteristics of this method lies in using $\tilde{y}$ as the input to learning, unlike standard semi-supervised learning that uses $y = [y_1, \ldots, y_l, 0, \ldots, 0]^T$. In other words, zero values in y are modified to some values denoting pseudolabels of unlabeled data.

Accuracy of LapERLS is often low because the original labeled set y is replaced with the intermediate decision variable g. Moreover, when available number of labeled data is small, accuracy of the pseudolabels, which is difference between true labels of unlabeled datapoints and pseudolabels of the unlabeled datapoints, is significantly low.

IV. Proposed Semi-Supervised Learning

This section describes a new algorithm by extracting key ideas from LapLS and LapERLS reviewed in the previous section. In section IV-A, we add a time-series representation to unlabeled data in order to obtain accurate pseudolabels. In section IV-B, pseudolabels are used in LapLS structure, which gives optimal solution by balanced pseudolabels and labeled data. Notations are equivalent to the previous section.

A. Time-series LapERLS

We first review the time-series learning algorithm [18]. In [18], optimization problem is built by applying Hodric-Prescott (H-P) filter [21] that obtains smoothed-curve representation of a time-series from training data, given by

$$\min_f \sum_{i=1}^{t} \left(f(x_i) - y_i\right)^2 + \gamma_T \sum_{i=3}^{t} \left(f(x_i) + f(x_{i-2}) - 2f(x_{i-1})\right)^2, \tag{14}$$

where $\{(x_i, y_i)\}_{i=1}^t$ is time-series labeled training data. The second term is to make the sequential points $f(x_i), f(x_{i-1}), f(x_{i-2})$ on a line. The solution of (14) in the matrix form is,

$$f = \left(I + \gamma_T D D^T\right)^{-1} y,$$

where

$$D = \begin{bmatrix} 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & -2 & 1 \end{bmatrix}_{t \times t}. \tag{15}$$

Our idea is to assign a temporal meaning to unlabeled data, while in standard LapERLS unlabeled data are represented for only spatial meaning by graph Laplacian. Representation of spatio-temporal unlabeled data is reasonable because training datapoints for a smoothly-moving object are collected in a chronological order.

Now, we add H-P filter term into LapERLS formulation (9):

$$\begin{aligned} \min_{f \in \mathcal{H}_k, g \in \mathcal{R}^{(l+u)}} & C \sum_{i=1}^{l+u} V(x_i, g_i, f) + \gamma_C \sum_{i=1}^{l} (g_i - y_i)^2 \\ & + \gamma_A \|f\|_A^2 + \gamma_I \|g\|_I^2 \\ & + \gamma_T \sum_{i=3}^{l+u} \left(g(x_i) + g(x_{i-2}) - 2g(x_{i-1})\right)^2. \end{aligned} \tag{16}$$

After rearranging (16) using the process similar to (9) through (12), we can obtain the optimization form of time-series LapERLS as the following:

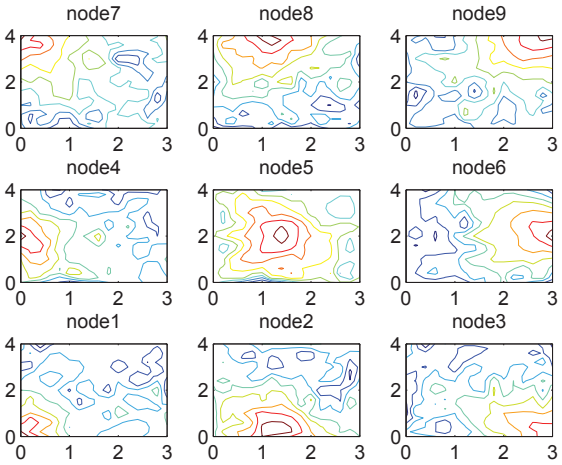
$$\min_{\beta \in \mathcal{R}^{(l+u)}} \frac{1}{2} \beta^T \tilde{Q} \beta + \beta^T \tilde{y}, \tag{17}$$

where

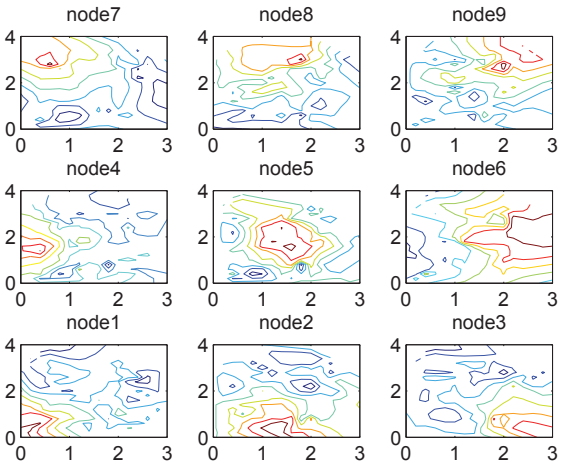
$$\tilde{Q} = K + (\Lambda + \mu_1 L + \mu_2 D D^T)^{-1}, \tag{18}$$

$$\tilde{y} = (\Lambda + \mu_1 L + \mu_2 D D^T)^{-1} \Lambda y, \tag{19}$$

$$\beta = -\alpha.$$



(a) Pseudolabel distribution using time-series LapERLS



(b) Pseudolabel distribution using standard LapERLS

Fig. 3: Wifi signal strength distribution of 9 access points using pseudolabels obtained from 25% of the labeled data among 221 training datapoints.

In comparison with (13) of LapERLS, $\mu_2 DD^T$ is added. We perform an example for showing difference of pseudolabels using LapERLS and time-series LapERLS.

Example 1: We collect time-series training data as the robot moves over time and space smoothly as shown in Fig. 2(a). In this example, we use 25% labeled training data and 75% unlabeled data among 221 training data. Fig. 2 illustrates estimations of pseudolabels between time-series LapERLS and standard LapERLS. As shown in Figs. 2(a) and 2(b), pseudolabels produced by the time-series LapERLS are accurate while the standard LapERLS cannot generate meaningful pseudolabels. Furthermore, pseudolabels of the standard LapERLS are incorrect even if we use 80% of the labeled data as shown in Fig. 2(c), while pseudolabels of the time-series LapERLS are accurate even when using only

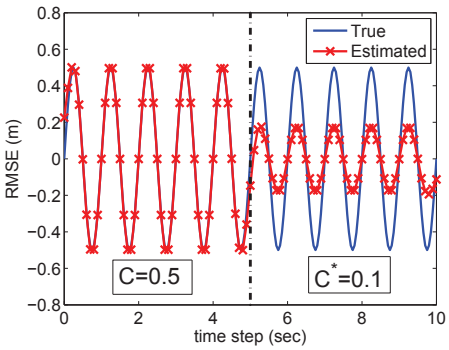


Fig. 4: Sine function estimation using LapLS in III-B with different values of balancing parameters C and C^*.

20% of the labeled data. We note that if a large number of unlabeled data are incorrectly pseudolabeled such as Fig. 2(b), that model results in bad localization performance. Another physical interpretation about pseudolabels can be seen from Fig. 3. We obtain pseudolabels from 25% labeled data, and plot pseudolabel distribution of each access point. In case of the standard LapERLS, data distributions of nodes 6, 7, 8, and 9 are severely distorted due to incorrect pseudolabels. On the other hand, the time-series LapERLS gives data distribution very similar to the original distribution using 100% labeled data in Fig. 1.

Unfortunately, it is difficult to obtain good $\tilde{Q}$ and $\tilde{y}$ simultaneously because tuning parameters Λ, μ_1, and μ_2 are used for both transformed kernel matrix $\tilde{Q}$ (18) and pseudolabel set $\tilde{y}$ (19). Our strategy is to pick up only $\tilde{y}$ (19) and use as a new input of other semi-supervised learning structure so that it becomes easy to balance the labeled data and the pseudolabeled data, as to be discussed in the following section.

B. Balancing Labeled and Pseudolabeled data

It is difficult to regard pseudolabels as labeled data, because true labels of unlabeled data are not known. A desirable way is to properly balance labeled and pseudolabeled data. This is feasible by applying the LapLS structure in section III-B, which can control the balance of training data with decoupled parameters C and C^* in (8).

Example 2: Fig. 4 illustrates estimation of sine function using LapLS in section III-B where we divide labeled training set in half and use different values of parameters, i.e. $C = 0.5$ and $C^* = 0.1$. In the latter part with $C^* = 0.1$, estimation is not accurate. In (8), as the parameter $C^{(*)}$ becomes smaller, the related term $C^{(*)} \sum_i (e_i^{(*)})^2$ becomes also smaller. In other words, optimization less focuses on the training datapoints with the smaller parameter value of $C^{(*)}$.

Our idea is to use pseudolabels $\tilde{y}$ (19) as the labels of unlabeled data for LapLS (8), which forms the following

Algorithm 1 Proposed semi-supervised learning for localization

Step 1 : Collect the training data set in time-series, i.e. l labeled samples $\{(x_i, y_i)\}_{i=1}^{l}$ and u unlabeled samples $\{x_j\}_{j=1}^{u}$.
Step 2 : Build the kernel matrix K (4), normalized Laplacian matrix L in (5), and Λ in (11).
Step 3 : Choose values of μ_1 and μ_2 in (19), and then obtain pseudolabels using (19).
Step 4 : Choose values of C and $\tilde{C}^*$, and then solve linear equation (21).

optimization:

$$\begin{aligned}\min_{\alpha\in\mathcal{R}^{(l+u)},e\in\mathcal{R}^{l},e^*\in\mathcal{R}^{u},b\in\mathcal{R}} & \frac{C}{2}\sum_{i=1}^{l}e_i^2+\frac{\tilde{C}^*}{2}\sum_{j=1}^{u}(\tilde{e}_j^*)^2 \\ & +\gamma_A\alpha^T K\alpha+\gamma_I\alpha^T KLK\alpha \\ \text{subject to : } y_i-\sum_{k=1}^{l+u}\alpha_k K_{ik}-b-e_i=0, & \quad i=1,\ldots,l. \\ \tilde{y}_j^*-\sum_{k=1}^{l+u}\alpha_k K_{jk}-b-\tilde{e}_j^*=0, & \quad j=1,\ldots,u,\end{aligned} \tag{20}$$

where $\tilde{y}_j^*$ are pseudolabels of unlabeled data from $\tilde{y}$ (19). Therefore, the non-convex problem of LapLS (8) is modified to a convex problem due to insertion of pseudolabels. After KKT conditions, we obtain the following linear system:

$$A\mathbf{X}=\mathbf{Y}, \tag{21}$$

where

$$\begin{aligned}A &= \begin{bmatrix} K+\Gamma & \mathbf{1}_{(l+u)\times 1} \\ \mathbf{1}_{1\times(l+u)} & 0 \end{bmatrix}, \\ \mathbf{X} &= \begin{bmatrix} \alpha \\ b \end{bmatrix}, \mathbf{Y}=\begin{bmatrix} \tilde{Y} \\ 0 \end{bmatrix},\end{aligned}$$

where K is the kernel matrix in (4), $\alpha=[\alpha_1,\ldots,\alpha_{l+u}]\in\mathcal{R}^{(l+u)}$, $b\in\mathcal{R}$ in (2), $\mathbf{1}_{(l+u)\times 1}=[1,\ldots,1]^T\in\mathcal{R}^{(l+u)}$, and Γ is diagonal matrix with $\Gamma_{ii}=1/C$ for $i=1,\ldots,l$ and $\Gamma_{ii}=1/\tilde{C}^*$ for $i=l+1,\ldots,l+u$. Pseudolabel vector $\tilde{Y}\in\mathcal{R}^{(l+u)}$ is a time-series set of labeled datapoints and pseudolabels of unlabeled data.

The proposed algorithm of (21) improves LapLS and LapERLS. First, pseudolabels are accurately estimated by assigning temporal-spatio representation into unlabeled data. Second, it is easy to balance pseudolabels and labeled data. Moreover, by incorporating pseudolabels into the LapLS structure, the non-convex problem is transformed to a convex one, which corresponds to a linear system that can be computed quite fast. The proposed algorithm is summarized in **Algorithm 1**.

V. Experiments

This section describes parameter setting in V-A and localization result of the proposed learning algorithm using wifi signal strength data in V-B. Section V-C shows localization results according to the different number of unlabeled data and values of tuning parameters, and Section V-D describes the computation time of the compared algorithms.

A. Parameter Setting

Usually, parameters in machine learning have been selected by cross validation that obtains optimal parameters to minimize the total training error of split training data, e.g. 10-fold cross validation [22]. However, most semi-supervised learning applications use a small number of labeled data, so it is not suitable to employ the cross validation. Instead of empirical selection of parameters using training data, this section guidelines selections of parameters by describing the physical meaning of each parameter in our algorithm.

First, λ in Λ in Step 2 of **Algorithm 1** can be interpreted as importance of labeled data relative to unlabeled data by (13) or (19). If λ is relatively small, resultant pseudolabels of the labeled data are different from the true labels. Therefore, we select the value of λ larger than the value that makes difference between pseudolabels of labeled data and true labels. Second, μ_1 and μ_2 in Step 3 of **Algorithm 1** have trade-off relationship between spatial and temporal correlation. If it is desirable to weight temporal meaning more than spatial meaning, μ_2 is selected larger than μ_1. Finally, C and $\tilde{C}^*$ in Step 4 of **Algorithm 1** have trade-off relationship about importance between labeled data and pseudolabels of unlabeled data, as described in IV-B. Thus in general, the value of $\tilde{C}^*$ is selected to be smaller than the value of C.

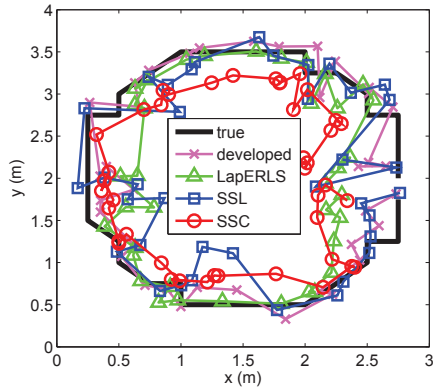
B. Localization using Wifi Signal Strength

This section shows the localization test using only wifi measurement in comparison with LapERLR, SSL, and SSC. The mobile robot moves along the circular trajectory in Fig 5(a). Fig 5(b) shows RMSE (root mean squared error) according to percentage of the used labeled data among 221 labeled data. As a result, the proposed algorithm yields the best localization among the compared algorithms over various numbers of labeled training data. In particular, although only 20 labeled datapoints are used, our algorithm yields accurate localization in comparison with the other algorithms, which confirms that our algorithm efficiently exploits the unlabeled data.

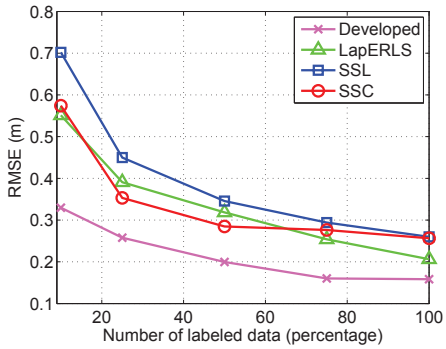
C. Robustness to Variations of Unlabeled Data and Tuning Parameters

Although utilization of unlabeled data is a key advantage of semi-supervised learning, if unlabeled data are incorrectly pseudolabeled, they may deteriorate the accuracy as mentioned in *Example 1* of the section IV-A. Also, unlabeled data may not be helpful when tuning parameters are not well selected. In this section, we examine effects of amount of unlabeled data for the fixed labeled datapoints and the change of values of tuning parameters. We test the learning performance for localization whose setup is the same as section V-B.

First, in the test for variation of unlabeled data, we fix values of tuning parameters for all the compared algorithms. They are



(a) Localization results of the compared algorithms in circular path



(b) RMSE according to the percentage of labeled data

Fig. 5: Localization results

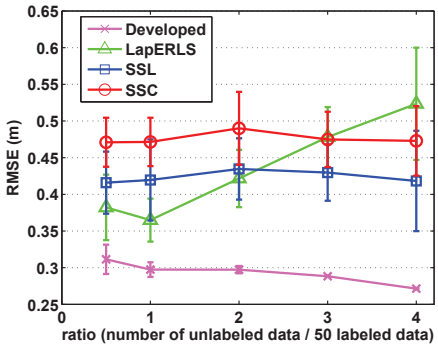


Fig. 6: Localization performance according to ratio of the unlabeled data per 50 fixed labeled data.

set to the values that provide the lowest error for the case when the ratio between the number of unlabeled data and 50 labeled data is 1. Moreover, the variability about randomly selected unlabeled data is analyzed through 10 repeated simulations, whose mean and deviation are shown in Fig. 6. The proposed algorithm shows gradually decreasing error according to the increasing number of the unlabeled data, and shows the lowest deviation of error. LapERLS shows the increasing error with respect to the increasing amount of the unlabeled data, which

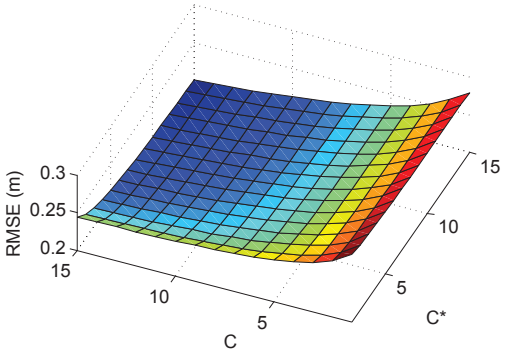


Fig. 7: Localization performance according to variation of tuning parameters of the proposed algorithm.

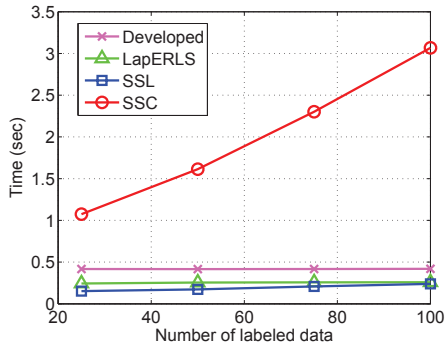


Fig. 8: Computation time of the compared algorithms

demonstrates the disadvantage of the incorrectly obtained pseudolabels. The unlabeled data are not helpful for SSC and SSL because they cannot use unlabeled data efficiently when only a few labeled data are available.

Second, we test our algorithm by varying the parameters C and $\tilde{C}^*$ in (20) using the same set of labeled data. In Fig. 7, except very small value of C, the RMSE does not show the large difference. Also, when C is fixed, the variation of the value of C^* does not much affect the RMSE. This is advantageous in that our algorithm is robust to variation of values of the tuning parameters.

D. Speed Test

The combined computation time of the SSL algorithm and pseudolabelling (19) is considered as the computational cost of the proposed algorithm. Therefore, it needs slightly more time than each SSL and LapERLS. Fig. 8 shows computational time of the compared algorithms using 50 unlabeled datapoints and the varying amount of labeled data, where the same localization setup as described in section V-C is used. The computation time of SSC increases significantly according to the increasing number of labeled data while the others remain bounded regardless of the amount of labeled data. The proposed algorithm needs little more time than LapERLS and SSL, but this difference is negligible in applications such as indoor localization.

VI. Conclusion

This paper proposes a new semi-supervised learning algorithm by combining core concepts of pseudolabelling of LapERLS, time-series learning, and LapLS with balanced training data. From the experiment, our algorithm achieves good accuracy using only a small number of the labeled training data. In comparison with state-of-art semi-supervised algorithms, the proposed algorithm yields the most precise performance, robustness to the varying amount of training data, and fast computation.

References

[1] J. Yoo, W. Kim, and H. J. Kim, "Distributed estimation using online semi-supervised particle filter for mobile sensor networks," *IET Control Theory & Applications*, vol. 9, no. 3, pp. 418–427, 2015.

[2] S. Choi, J. Yoo, and H. J. Kim, "Machine learning for indoor localization: Deep learning and semi-supervised learning," in *International Conference on Indoor Positioning and Indoor Navigation*, 2015.

[3] J. Yoo, H. J. Kim, and K. H. Johansson, "Mapless indoor localization by trajectory learning from a crowd," in *IEEE International Conference on Indoor Positioning and Indoor Navigation*, 2016, pp. 1–7.

[4] B. Balaguer, G. Erinc, and S. Carpin, "Combining classification and regression for wifi localization of heterogeneous robot teams in unknown environments," in *IEEE International Conference on Intelligent Robots and Systems*, 2012, pp. 3496–3503.

[5] C. Wu, Z. Yang, Y. Liu, and W. Xi, "Will: Wireless indoor localization without site survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 839–848, 2013.

[6] M. D. Redžić, C. Brennan, and N. E. O'Connor, "Seamloc: Seamless indoor localization based on reduced number of calibration points," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1326–1337, 2014.

[7] S.-H. Fang and T. Lin, "Principal component localization in indoor wlan environments," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 100–110, 2012.

[8] J. J. Pan, S. J. Pan, J. Yin, L. M. Ni, and Q. Yang, "Tracking mobile users in wireless networks via semi-supervised colocalization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 587–600, 2012.

[9] J. Yoo and H. J. Kim, "Target localization in wireless sensor networks using online semi-supervised support vector regression," *Sensors*, vol. 15, no. 6, pp. 12 539–12 559, 2015.

[10] L. Gómez-Chova, G. Camps-Valls, J. Munoz-Mari, and J. Calpe, "Semisupervised image classification with laplacian support vector machines," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 3, pp. 336–340, 2008.

[11] J. Chen, C. Wang, Y. Sun, and X. Shen, "Semi-supervised laplacian regularized least squares algorithm for localization in wireless sensor networks," *Computer Networks*, vol. 55, no. 10, pp. 2481–2491, 2011.

[12] L. Chen, I. W. Tsang, and D. Xu, "Laplacian embedded regression for scalable manifold regularization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 6, pp. 902–915, 2012.

[13] O. Chapelle, V. Vapnik, and J. Weston, "Transductive inference for estimating values of functions," in *Conference on Neural Information Processing Systems*, vol. 12, 1999, pp. 42l–427.

[14] X. Zhu, Z. Ghahramani, J. Lafferty *et al.*, "Semi-supervised learning using gaussian fields and harmonic functions," in *International Conference on Machine Learning*, vol. 3, 2003, pp. 912–919.

[15] M. M. Adankon, M. Cheriet, and A. Biem, "Semisupervised least squares support vector machine," *IEEE Transactions on Neural Networks*, vol. 20, no. 12, pp. 1858–1870, 2009.

[16] F. Nie, D. Xu, X. Li, and S. Xiang, "Semisupervised dimensionality reduction and classification through virtual label regression," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 3, pp. 675–685, 2011.

[17] P. Kumar Mallapragada, R. Jin, A. K. Jain, and Y. Liu, "Semiboost: Boosting for semi-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2000–2014, 2009.

[18] D. A. Tran and T. Zhang, "Fingerprint-based location tracking with Hodrick-Prescott filtering," in *IEEE Conference on Wireless and Mobile Networking*, 2014, pp. 1–8.

[19] J. Yoo and H. J. Kim, "Online estimation using semi-supervised least square svr," in *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2014, pp. 1624–1629.

[20] M. Belkin and P. Niyogi, "Semi-supervised learning on riemannian manifolds," *Machine learning*, vol. 56, no. 1-3, pp. 209–239, 2004.

[21] M. O. Ravn and H. Uhlig, "On adjusting the Hodrick-Prescott filter for the frequency of observations," *Review of economics and statistics*, vol. 84, no. 2, pp. 371–376, 2002.

[22] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International Joint Conference on Artificial Intelligence*, vol. 14, no. 2, 1995, pp. 1137–1145.