

Domain-Constrained Semi-Supervised Mining of Tracking Models in Sensor Networks

Rong Pan¹, Junhui Zhao², Vincent Wenchen Zheng¹, Jeffrey Junfeng Pan¹,
Dou Shen¹, Sinno Jialin Pan¹ and Qiang Yang¹

¹Hong Kong University of Science & Technology
{panrong,vincentz,panjf,dshen,sinnopan,
qyang}@cse.ust.hk

²NEC Labs, China, Zhongguancun
East Road, Beijing 100084, China
zhaojunhui@research.nec.com.cn

ABSTRACT

Accurate localization of mobile objects is a major research problem in sensor networks and an important data mining application. Specifically, the localization problem is to determine the location of a client device accurately given the radio signal strength values received at the client device from multiple beacon sensors or access points. Conventional data mining and machine learning methods can be applied to solve this problem. However, all of them require large amounts of labeled training data, which can be quite expensive. In this paper, we propose a probabilistic semi-supervised learning approach to reduce the calibration effort and increase the tracking accuracy. Our method is based on semi-supervised conditional random fields which can enhance the learned model from a small set of training data with abundant unlabeled data effectively. To make our method more efficient, we exploit a Generalized EM algorithm coupled with *domain constraints*. We validate our method through extensive experiments in a real sensor network using Crossbow MICA2 sensors. The results demonstrate the advantages of methods compared to other state-of-the-art object-tracking algorithms.

Categories and Subject Descriptors

I.2.6 [Artificial Intelligence]: Learning; H.2.8 [Database Management]: Database Applications—*Data mining*

General Terms

Algorithms

Keywords

Localization, Calibration, Tracking, Sensor Networks, EM, CRF

1. INTRODUCTION

Recently, *wireless sensor networks* have attracted great interests in several related research fields and industries. Many tasks such as context-aware computing [4] and environmental monitoring can be realized with the help of wireless sensor networks, which offer

unique advantages of being lightweight, distributed, environment-aware and network-based. Object tracking, event-detection and activity recognition can now be realized in sensor networks using probabilistic algorithms [7, 11]. It is a fundamental task for many of these applications to locate mobile client devices using collected wireless signals (in terms of radio-signal-strength, RSS, values) from different sensor nodes that act as beacons.

In the past, some conventional data mining technologies have been applied for solving the localization problem [7, 9]. Generally, some statistical models are obtained offline which can map signals to locations. These models are then used online to predict the client locations based on the real-time signal values. Among the past works, many researchers have developed ranging-based algorithms for localizing mobile nodes in a wireless sensor network. One approach is multilateration (e.g., [9]), consisting of two main steps. It first transforms the sensor readings into a distance measure. It then attempts to recover the coordinate locations in terms of relative distance to the *beacon* nodes. This approach relies on an ideal signal propagation model and extensive hardware support. It suffers from low accuracy problem since RSSs do not follow ideal propagation patterns, especially in complex environments.

In this paper, we address this problem using a semi-supervised statistical relational learning approach based on conditional random fields (Semi-CRF). We assume that a mobile sensor node moves in a sensor network environment. The RSS of the mobile node can be received by several sensors in the network, which are then forwarded to a processor for tracking. It can also happen in the way that all sensors in the network are sending signals to the mobile sensor node, which performs the localization itself. In either case, we have a sequence of multi-dimensional vectors that corresponds to a trace. Each vector along the trace can be labeled with a physical location coordinate, or unlabelled.

This paper makes the following contributions. First, we identify and solve a major bottleneck in the application of data mining technologies in sensor networks. Second, we present a novel semi-supervised learning method for mobile-node tracking and localization by utilizing both labeled and unlabelled RSS trace data. Third, we introduce domain-driven heuristics for reducing the complexity of the learning procedure, which greatly improve the scalability of the statistical models. Finally, we validate the proposed methods through the experiments over a real sensor network.

2. THE Semi-CRF ALGORITHM FOR TRACKING AND LOCALIZATION

2.1 Problem Statement

Consider a two-dimensional tracking problem. Our objective is

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD'07, August 12–15, 2007, San Jose, California, USA.
Copyright 2007 ACM 978-1-59593-609-7/07/0008 ...\$5.00.

to determine the location of a mobile object $y_t = (u_t, v_t) \in \mathcal{C}$ as it moves in a sequence, given the observed signal vectors \mathbf{x}_t . Figure 3 shows an example of the floor in one of our experimental test beds, which consists of $N = 8$ beacon nodes and one mobile *unknown* node. The localization problem can be converted to a supervised classification problem if we had sufficient labeled data for each location. However, when the labeled data are insufficient at each location, we wish to make the best use of some partially labeled or totally unlabeled RSS sequences as well in our prediction.

Now let us formally introduce the notation of training data. In our study, the training data consist of a set of fully labelled sequences $D^f = \{(X_1, Y_1), \dots, (X_M, Y_M)\}$, and a set of partially labeled (or totally unlabeled) sequences $D^p = \{(X_{M+1}, Y_{M+1}), \dots, (X_{M+L}, Y_{M+L})\}$, where X_i is a sequence of signal vectors $\mathbf{x}_{i1}, \dots, \mathbf{x}_{im_i}, i = 1, \dots, M + L$, and Y_i is a sequence of corresponding locations $y_{i1}, \dots, y_{im_i}, i = 1, \dots, M + L$. Some values of y_{ij} are unknown for $M + 1 \leq i \leq M + L$. A mobile robot can be employed to collect these unlabelled data by simply wandering around.

2.2 Linear-chain CRF

2.2.1 The CRF Model

In this paper, we propose a statistical relational learning approach using CRF to exploit the relationship between RSS readings at two neighboring time points in terms of their corresponding physical distance. As a mobile object moves around, a sequence of RSS values can be received, with each corresponding to a certain location. This process can be modeled by an 1-D Linear-chain CRF as introduced in the following section, where the states correspond to the location labels and the inputs or observations correspond to the RSS readings.

Linear-chain CRF models have been widely used to model sequential data. These models can be roughly viewed as conditionally-trained finite state machines [5]. A linear-chain CRF, as shown in Figure 1, defines a distribution over state sequence $\mathbf{y} = y_1, y_2, \dots, y_T$ given an input sequence $\mathbf{x} = x_1, x_2, \dots, x_T$ by making a first-order Markov assumption on states, where T is

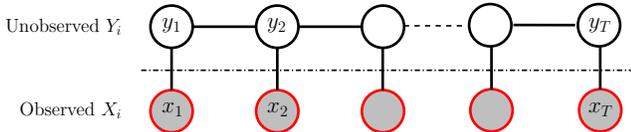


Figure 1: Diagram of Linear-chain CRF

the length of the sequence. These Markov assumptions imply that the distribution over sequences factorizes in terms of potential functions $\Phi_t(y_{t-1}, y_t, \mathbf{x})$ as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{\prod_t \Phi_t(y_{t-1}, y_t, \mathbf{x})}{Z(\mathbf{x})}, \quad (1)$$

where the partition function $Z(\mathbf{x})$ is the normalization constant that makes the probability of all state sequences sum to one. It is defined as follows:

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_t \Phi_t(y_{t-1}, y_t, \mathbf{x}). \quad (2)$$

The potential functions $\Phi_t(y_{t-1}, y_t, \mathbf{x})$ can be interpreted as the cost of making a transition from state y_{t-1} to state y_t at time t , similar to a transition probability in an HMM.

Computing the partition function $Z(\mathbf{x})$ requires summing over the exponentially many possible state sequences \mathbf{y} . By exploiting

the Markov assumption, however, $Z(\mathbf{x})$ (as well as the node marginal $p(y_t|\mathbf{x})$ and the Viterbi labeling) can be calculated efficiently by variants of the standard dynamic programming algorithms used in HMM.

We assume that potentials factorize themselves according to a set of features f_k that are given and fixed, so that

$$\Phi(y_{t-1}, y_t, \mathbf{x}) = \exp\left(\sum_k \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}, t)\right). \quad (3)$$

The model parameters are a set of real weights $\Lambda = \{\lambda_k\}$, one for each feature (to be defined below). The feature functions can describe any aspect of a transition from y_{t-1} to y_t as well as y_t and the global characteristics of \mathbf{x} . For example, f_k may have value 1 when the distance between y_{t-1} and y_t is smaller than 50cm.

2.2.2 Parameter Estimation

The parameters Λ can be estimated through a maximum likelihood procedure using the training data. That is, we can estimate them by maximizing the conditional log-likelihood of the labeled sequences in the training data $\Psi = \{(X_1, Y_1), \dots, (X_M, Y_M)\}$, which is defined as:

$$L(\Lambda) = \sum_{i=1}^M \log(P(Y_i|X_i; \Lambda)), \quad (4)$$

where M is the number of sequences. As discussed in Sutton et al. in [10], $L(\Lambda)$ is concave in light of the convexity of the kind of functions $g(x) = \log \sum_i \exp x_i$.

2.2.3 Inference

Given the conditional probability of the state sequence defined by a CRF in Equation (1) and the parameters Λ , the most probable labeling sequence can be obtained as

$$Y^* = \arg \max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}; \Lambda), \quad (5)$$

which can be efficiently calculated using the Viterbi algorithm [8]. The marginal probability of states at each position in the sequence can be computed by a dynamic programming inference procedure similar to the forward-backward procedure for HMM [3]. We can define the “forward values” $\alpha_t(y|\mathbf{x})$ by setting $\alpha_1(y|\mathbf{x})$ equal to the probability of starting with state y and then iterate as follows:

$$\alpha_t(y|\mathbf{x}) = \sum_{y'} \alpha_{t-1}(y'|\mathbf{x}) \exp(\Lambda_t(y', y, \mathbf{x})), \quad (6)$$

where $\Lambda_t(y', y, \mathbf{x})$ is defined by:

$$\Lambda_t(y', y, \mathbf{x}) = \sum_k \lambda_k f_k(y_{t-1} = y', y_t = y, \mathbf{x}) \quad (7)$$

Then $Z(\mathbf{x})$ equals to $\sum_y \alpha_t(y|\mathbf{x})$. The “backward values” $\beta_t(y|\mathbf{x})$ can be defined similarly.

$$\beta_t(y|\mathbf{x}) = \sum_{y''} \beta_{t+1}(y''|\mathbf{x}) \exp(\Lambda_t(y, y'', \mathbf{x})). \quad (8)$$

After that, we calculate the marginal probability of each location given the observed signal sequence:

$$P(y_t = g|\mathbf{x}) = \frac{\alpha_t(g|\mathbf{x}) * \beta_t(g|\mathbf{x})}{Z(\mathbf{x})}. \quad (9)$$

So far, we have introduced a linear-chain CRF model for unknown mobile-node tracking. We can see that $f_k(y_{t-1}, y_t, \mathbf{x})$ (in

Equation (3)) is an arbitrary feature function over the entire observed sequences and the states at positions t and $t - 1$. In our problem, the locations are two-dimensional continuous values. The number of possible locations are infinite large. Therefore, it is extremely difficult to compute the feature of two arbitrary locations.

Fortunately, the tracking area is known in advance usually. One solution is to discretize a 2-D location space into grids. For instance, in a $5m \times 4m$ area, we can divide it into 10×8 grids with each grid being $50 \times 50cm^2$. This example is shown in Figure 2. In this way, we can convert the known locations into such grids. In the test phase, if a mobile object is located at grid g_i , we can use the coordinates of the center point in g_i to represent the location of the mobile object. After limiting the location space, it is possible to use the linear-chain CRF approach for tracking problem. However, a major issue is how to determine the size of the grid. This problem can be solved in two ways. First, the size often is determined by the nature of the problem itself, which is decided by the precision requirement posed by application users. Another approach is to study the problem empirically, as we will do in the experimental section.

g_1	g_2							g_8	g_9	
			g_{24}							
			g_{35}							
			g_{46}							
										g_{80}

Figure 2: A demo of reduction of locations to grids

2.2.4 Incorporating Domain Constraints

After reducing locations to grids, we can specialize the feature functions for each possible transition among different grids. That is, we can define $f_k(y_{t-1} = g, y_t = h, \mathbf{x})$ by $f_{(g,h)}(t-1, t, \mathbf{x})$. However, the number of the transition feature functions, as well as the corresponding parameters, reaches n^2 (n is the number of grids), which can be quite large. For instance, in the above example in Figure 2, $n = 80$, then in the CRF learning, we need to estimate 6400 parameters for the potential $f_k(y_{t-1}, y_t, \mathbf{x})$. Although we can still estimate the values of the parameters with large n , it will certainly increase the computational cost and run the risk of overfitting. What is worse, learning CRF with more parameters requires more training data, which will increase the labelling costs. In addition, we also need to trade off the complexity of the model and its generalization capability. If we increase the grid size to reduce the computational cost, we will sacrifice the estimation accuracy.

In this paper, we incorporate the domain constraints in the data mining process to reduce the number of parameters that need be learned. In particular, we note that a mobile object in a sensor network typically moves around in the same way, such that the likelihood of transitioning between two neighboring points are roughly same. The likelihood of traveling between two distant points will also be roughly the same, although the value will be much smaller. Such a domain constraint is supported by our experiments.

To incorporate the domain constraints mentioned above, we use a so-called *parameter tying* technique that is designed for speech recognition [6] to combine similar parameters. Our assumption is that the characteristics of two transitions with the same distance are alike. Intuitively, in Figure 2, we observe that the transitions $g_1 \leftrightarrow g_2$ and $g_8 \leftrightarrow g_9$ should happen with similar frequencies as they both transit by one grid in terms of Euclidean distance. Similarly, the transitions $g_{24} \leftrightarrow g_{35}$ and $g_{35} \leftrightarrow g_{46}$ should happen similarly as their Euclidean distances are both $\sqrt{2}$ grids.

From this observation, we can tie the parameters of the transition feature functions so long as they have the same transition distance, which is defined as follows: The value of the transition feature function $f_k(y_{t-1}, y_t, \mathbf{x})$ equals one if and only if $dis(y_{t-1}, y_t) = k$, where the dis defines the distance between the two points. As expected, the number of parameters is greatly reduced by using this constraint.

2.3 The Semi-CRF Algorithm

In this section, we introduce how to incorporate sequences whose labels are fully or partially observed in the parameter estimation of CRF.

An efficient method for parameter estimation with incomplete data can be derived by the extension of EM algorithm [2]. In this paper, we use a Generalized Expectation Maximization (GEM) algorithm to learn the parameters Λ of CRF with both fully and partially observed data [1]. In the GEM algorithm, the probabilistic optimization problem is divided into two-step iterations. The unobserved data are estimated in the E-step with the parameters obtained in the last iteration and the parameters of CRF are optimized in the M-step. We first compute the log likelihood of Equation (4) with expectation over the unobserved data as follows:

$$\begin{aligned}
 & L(\Lambda; \Lambda^t) \\
 &= \sum_{i=1}^M \log P(Y_i | X_i; \Lambda) + \\
 & \quad \sum_{i=M+1}^{M+L} \sum_{Y_i^{(u)}} P(Y_i^{(u)} | X_i, Y_i^{(o)}; \Lambda^t) \log P(Y_i^{(u)}, Y_i^{(o)} | X_i; \Lambda) \\
 &= \sum_{i=1}^M \sum_t \Lambda_t(y', y, X_i) - \sum_{i=1}^{M+L} \log Z(X_i) + \\
 & \quad \sum_{i=M+1}^{M+L} \sum_{Y_i^{(u)}} P(Y_i^{(u)} | X_i, Y_i^{(o)}; \Lambda^t) \sum_t \Lambda_t(y', y, X_i)
 \end{aligned}$$

In this equation, $Y_i^{(u)}$ is the unobserved locations of the i -th sequence, $Y_i^{(o)}$ is the observed counterpart,

$$\Lambda_t(y', y, X_i) = \sum_k \lambda_k f_k(y', y, X_i),$$

and

$$Z(X_i) = \sum_y \exp \left(\sum_t \Lambda_t(y_{t-1}, y_t, X_i) \right).$$

Similar to Equation (4), $L(\Lambda; \Lambda^t)$ is also concave. We can use the same method to optimize it. The only problem left is how to infer for partially observed sequences. We need to change Equations (6) and (8) for some cases. If $y_t = j$ is observed, we directly assign 1 to $\alpha_t(y = j | \mathbf{x})$ and $\beta_t(y = j | \mathbf{x})$ and assign 0 to the other values of α_t and β_t . If y_t is not observed, we follow Equations (6) and (8). The new inference formulae are summarized in Equations (10) to (12).

$$\alpha_t(y = j | \mathbf{x}) = \begin{cases} 0 & y_t \neq j \\ 1 & y_t = j \\ \sum_{y'} \alpha_{t-1}(y' | \mathbf{x}) \exp(\Lambda_t(y', y, \mathbf{x})) & y_t \text{ unseen} \end{cases} \quad (10)$$

$$\beta_t(y = j|\mathbf{x}) = \begin{cases} 0 & y_t \neq j \\ 1 & y_t = j \\ \frac{\alpha_t(y = j|\mathbf{x}) \exp(\Lambda_t(y', y, \mathbf{x}))}{\sum_{y'} \beta_{t+1}(y'|\mathbf{x}) \exp(\Lambda_t(y', y, \mathbf{x}))} & y_t \text{ unseen} \end{cases} \quad (11)$$

$$P(y_t = k|\mathbf{x}) = \frac{\alpha_t(y = k|\mathbf{x}) * \beta_t(y = k|\mathbf{x})}{Z_{\mathbf{x}}}. \quad (12)$$

We now summarize the Semi-CRF learning algorithm in Table 1. There are several ways of parameter initialization. The common one is to randomly assign them values from 0 to 1. To speed up the convergence, we use an alternative that preliminarily estimates parameters with labeled data. As to the number of iterations, we will discuss it in the experimental section.

Table 1: The training algorithm for CRF with both fully and partially observed data.

<p>Algorithm Semi-CRF <i>Input:</i> The fully and partially observed data D^f, D^p <i>Output:</i> The parameters Λ of CRF</p> <hr/> <p>Initialize parameters Λ_0 of CRF. $\Lambda_t = \Lambda_0$. while log-likelihood has not converged or the max number of iterations is not reached, do % ===== E-step ===== Compute the expectations of the all unobserved locations, by Equations (10) to (12). % ===== M-step ===== Optimize Λ using L-BFGS. $\Lambda_t = \Lambda$. endwhile return Λ</p>

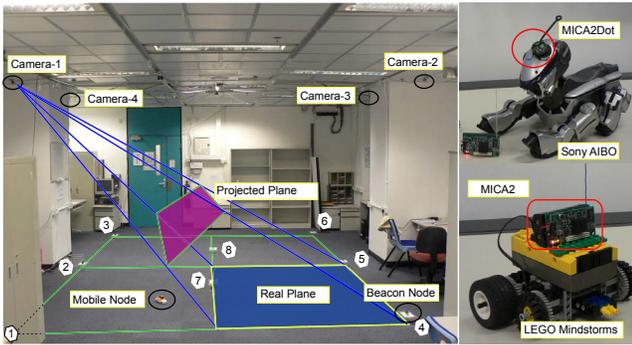


Figure 3: Experimental Test-Bed

3. EXPERIMENTAL EVALUATION

3.1 Experimental Setup

We test the effectiveness and robustness of our location tracking algorithm for mobile sensor nodes in a sensor network based on the RSS signals. Our experiments are performed in the Pervasive Computing Laboratory (Figure 3) in the Department of Computer Science and Engineering at Hong Kong University of Science and Technology. The room is large enough for us to set up

an experimental test-bed of 5.0 meters by 4.0 meters. In Figure 3, $|P_1P_3| = |P_4P_6| = 5.0m$ and $|P_1P_4| = |P_3P_6| = 4.0m$. There are three main components of our setup:

- *Wireless Sensor Networks.* We use CrossBow MICA2 and MICA2Dot to construct a wireless sensor network. We program these sensor nodes to broadcast and detect beacon frames periodically so that they could measure the RSS of each other.
- *Mobile Robots.* We try different kinds of robots that can run freely around the floor at different speeds, such as a Sony AIBO dogs, LEGO Mindstorms and off-the-shelf toy cars. Figure 3 shows that a sensor node is attached on top of a toy car which can be remotely controlled by radio at the speed of 0.4 m/s.
- A *Camera Array* is used to record the ground truth locations of the mobile robots for our training and test data.

We use two performance measurements to evaluate the original CRF and the CRF model using parameter tying (denoted by CRF-PT) localization algorithms. The first metric is the mean error-distance values between estimated and true locations. The second measurement is the accuracy in percentage. Given an error-distance threshold θ , the accuracy rate is the probability that the distance between the estimated and true locations is smaller than θ . Two more baselines in our experiments include (1) Logistic Regression (LR), (2) Support Vector Regression (SVR). We control a mobile robot to run and stop around the test area (Figure 3) for collecting data with sampling interval 0.5s. The data set formed a trace of length about 600m with 3,000 examples. For every experiment below, we randomly select a subset of the data as fully observed training data, a subset of data as partially observed training data by randomly removing the locations associated with them, and evaluate the performance on the rest. To reduce the statistical variability, we repeated the experiments for 30 times and reported the average results.

3.2 Convergence of Semi-CRF

One question about the the Semi-CRF algorithm is the convergence of the EM iterations. In this experiment, we use 10 fully labelled and 50 partially labelled sequential data to train the CRF, where the length of each sequence is 5 and only one node is labelled in the partially labelled data. Figure 4 shows the convergence rate of Semi-CRF. We can see that about 4 iterations are enough. In the experiments of this paper, the maximum number of iterations of the Semi-CRF is set to 10.

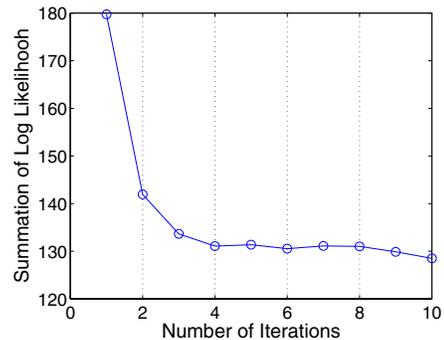


Figure 4: Convergence rate of Semi-CRF

3.3 Semi-CRF vs. Baselines

In the following experiments, we fix the training data size at 550, and tune the ratio of the labelled data from 0.33 to 1. In Figure 5, we show the mean error performance of the four algorithms described above. As can be seen, Semi-CRF consistently outperforms the other algorithms in terms of mean error distance, while CRF beats the remaining two baselines. One important reason is they both effectively leverage the sequential information of the mobile node. Moreover, as Semi-CRF can also learn from the unlabelled data, it gains much better performance when there are a lot of such data with a small portion of labelled ones. We list some more information of these experiments including the accuracy performance in Table 2.

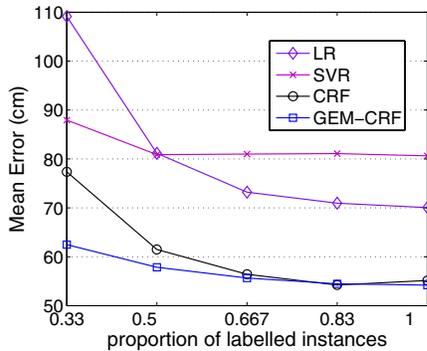


Figure 5: Vary the ratio of training set size

Table 2: Performance of the tested approaches

Approach	Mean(cm)	Accuracy at 100cm
Semi-CRF	56.9267	85.67%
CRF	60.9268	83.67%
SVR	82.7318	67.33%
LR	82.3059	69.85%

3.4 Impact of Grid Sizes

The grid size may affect the performance of the localization algorithms. In this experiment, we fix the ratio of labelled data at 5% and vary the side length of the grids from 10cm to 100cm. Figure 6 shows the experimental results of CRF and Semi-CRF. From the figure we can see that when the grid size ranges from 20cm to 50cm, the performance of both the two methods is less sensitive than that with the grid size of 10cm and 100cm.

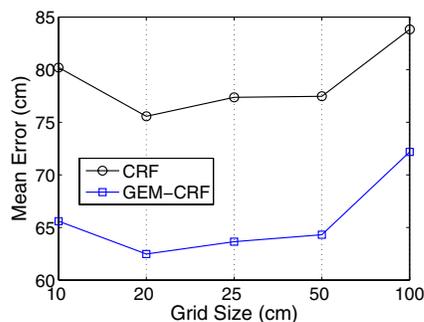


Figure 6: Vary the grid size (ratio of labelled data is 5%).

4. CONCLUSION AND FUTURE WORKS

We have presented a new approach to reducing the calibration effort when tracking *mobile* sensor nodes in a wireless sensor network. Our approach made extensive use of the sequential information of moving sensor's trajectory. These sequences provided unlabelled examples which can be used to train CRF together with the manually labelled RSS values. We introduce a Semi-CRF model to utilize such partially labelled data. By using parameter tying techniques we significantly improve the performance of Semi-CRF algorithm while reducing calibration effort. A sensor network was set up based on Crossbow MICA2 and MICA2Dot nodes which are used as both *beacon* and *mobile* nodes. Experimental results showed that the proposed method could achieve a better performance with relatively fewer number of labelled examples.

In the future, we plan to continue to test the Semi-CRF based framework in a large scale sensor network. We are also interested in introducing different factors, such as changing time and space, to see how the knowledge learned in one setting can be applied to another.

5. REFERENCES

- [1] H. L. Chieu, S. W. Lee, and P. K. Leslie. Activity recognition from physiological data using conditional random fields, January 2006.
- [2] A. P. Dempster, N. M. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 1(39):1–38, 1977.
- [3] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.
- [4] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *IJCAI*, pages 766–772, 2005.
- [5] A. McCallum. Efficiently inducing features of conditional random fields. In C. Meek and U. Kjærulff, editors, *UAI*, pages 403–410. Morgan Kaufmann, 2003.
- [6] C. Neukirchen, D. Willett, and G. Rigoll. Soft State-Tying for HMM-Based Speech Recognition. In *5th International Conference on Spoken Language Processing (ICSLP)*, pages 2999–3002, Sydney, 1998.
- [7] X. Nguyen, M. I. Jordan, and B. Sinopoli. A kernel-based learning approach to ad hoc sensor network localization. *ACM Transactions on Sensor Networks*, 1(1):134–152, 2005.
- [8] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [9] A. Savvides, C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pages 166–179, Rome, Italy, 2001.
- [10] C. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2006.
- [11] J. Yin, X. Chai, and Q. Yang. High-level goal recognition in a wireless LAN. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 578–584, San Jose, CA, USA, July 2004.