On Assessing the Accuracy of Positioning Systems in Indoor Environments

Hongkai Wen, Zhuoling Xiao, Niki Trigoni, and Phil Blunsom

University of Oxford
{name.lastname}@cs.ox.ac.uk

Abstract. As industrial and academic communities become increasingly interested in Indoor Positioning Systems (IPSs), a plethora of technologies are gaining maturity and competing for adoption in the global smartphone market. In the near future, we expect busy places, such as schools, airports, hospitals and large businesses, to be outfitted with multiple IPS infrastructures, which need to coexist, collaborate and / or compete for users. In this paper, we examine the novel problem of estimating the accuracy of co-located positioning systems, and selecting which one to use where. This is challenging because 1) we do not possess knowledge of the ground truth, which makes it difficult to empirically estimate the accuracy of an indoor positioning system; and 2) the accuracy reported by a positioning system is not always a faithful representation of the real accuracy. In order to address these challenges, we model the process of a user moving in an indoor environment as a Hidden Markov Model (HMM), and augment the model to take into account vector (instead of scalar) observations, and prior knowledge about user mobility drawn from personal electronic calendars. We then propose an extension of the Baum-Welch algorithm to learn the parameters of the augmented HMM. The proposed HMM-based approach to learning the accuracy of indoor positioning systems is validated and tested against competing approaches in several real-world indoor settings.

1 Introduction

We are only on the cusp of the wave of Indoor Positioning Systems (IPS), but there is already an explosion of technologies that compete for adoption in the global smart device market. These range from ultrasound-based positioning systems, to Bluetooth Low Energy, RFID systems, WiFi-based localization (with or without fingerprinting), inertial tracking, ground-based transmitters to extend GPS service indoors, visible light comms, and sensing ambient FM-radio, magnetic, and photo-acoustic signatures.

The above list is not exhaustive; the interest and competition in this area is growing so fast that an alliance of 22 industries, called In-Location, was founded in August 2012 to promote the adoption and deployment of IPS solutions. In the current climate, it is reasonable to expect that many indoor environments, such as schools, airports, hospitals and large businesses, will soon be outfitted with multiple IPS (Indoor Positioning System) infrastructures, which need to coexist, collaborate and / or compete for users. In this emerging ecosystem, it is important for users to be able to *accurately estimate the accuracy of an IPS at different locations of an indoor environment*. We first define

P. Demeester, I. Moerman, and A. Terzis (Eds.): EWSN 2013, LNCS 7772, pp. 1–17, 2013.

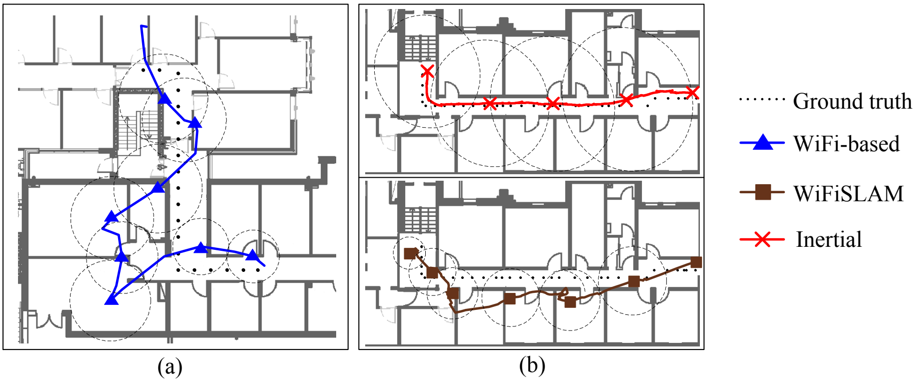


Fig. 1. (a) At the bottom left corner, the true locations fall out of the reported error ellipses. (b) The reported accuracies are misleading as to which is the best IPS.

what we mean by IPS *accuracy*, and then explain why it is both useful and challenging to estimate it.

IPS Accuracy: Consider the simple case where an IPS provides only location estimates without uncertainty information associated with them. Assume that the true location of the user at a given time is x and the measurement generated by the IPS is z. The accuracy ϵ of the IPS at location x depends on the Euclidean distance between the true and measured locations: $\epsilon(x) = f(\|z - x\|_2)$, where f is a monotonic increasing function. Many IPSs tend to also provide accuracy information, for example, they pair a location estimate with an error ellipse around it and a confidence interval. In such cases, the measurement z of an IPS can be viewed as a random variable with a probabilistic distribution function $\mathrm{P}_z : L \to [0, 1]$, where L is the sample space over all possible locations. With respect to the real location x, the accuracy of an uncertain measurement z depends on the *expected* Eculidean distance between the true and measured locations: $\epsilon(x) = f(\sum\limits_{\ell \in L} \mathrm{P}_z(\ell)(\|\ell - x\|_2))$.

Why Is It Useful to Estimate IPS Accuracy: Users will be able to make informed decisions as to which IPS to task when moving in an indoor space, and when to switch from one IPS to another. Of course, accuracy is not the only information required to make such decisions. Cost, power consumption and privacy should also be taken into consideration. It is possible that highly accurate IPSs may impose a small monetary cost on users whereas less accurate ones may be available for free. Also, the power consumption of different IPSs can vary up to magnitudes, and some users may be more concerned about privacy than others. In this paper, we focus on estimating accuracy, and assume that other quality metrics are available from other sources (energy monitoring tools, web sources, etc.). We claim that by providing accuracy information, we empower users to choose the IPS that achieves the desired tradeoff between accuracy and the remaining quality metrics.

Why is it Difficult to Estimate IPS Accuracy: First, the actual trajectory of a user is normally unavailable, and thus the accuracy of an IPS can not be directly computed.

Second, the accuracy of an IPS can vary significantly over space. Hence, we cannot rely on the accuracy of an IPS in one location to infer its accuracy in another. Thirdly, we view an IPS as a black box, and assume little or no knowledge about how it works or where it has been deployed.

Finally, the reported accuracy from an IPS (if provided) is not always a reliable indicator of the real accuracy. We *observe two general cases where this happens, and* provide illustrative examples: *Case 1:* The true location of a user consistently falls out of the reported ellipsoid. This tends to happen when the localization algorithm of an IPS can not cope with changes in the environment. For instance in Fig. 1(a), an IPS that uses WiFi RSSI measurements performs well in the straight corridor, but consistently underestimates the error in its measurements when the node is moving towards the corner. Another example concerns safety-critical environments, where one or more IPSs may be compromised and made to consistently report wrong locations with very high reported accuracy. *Case 2:* The size of ellipsoids provided by different IPSs can lead us to wrong conclusions about which one is more accurate. For instance, consider the two estimated trajectories of a single user shown in Fig. 1(b), where the top one is provided by an IPS using inertial sensors and the other is from WiFiSLAM [5]. One can immediately see that the estimated trajectory on top is far better than the bottom one, but the reported accuracy is significantly lower[1]. The aim of this paper is to address these challenges, and provide a solution to the IPS accuracy estimation problem.

Summary of Contributions: In this paper, we make the following four contributions:

1. To our knowledge, this is the first paper that defines the problem of assessing the accuracy of co-located indoor positioning systems. As explained above, this is a *timely* problem, and an effective solution could benefit a global market of IPS users.
2. We propose a maximum likelihood HMM-based approach to assessing IPS accuracy. We model the stochastic process of a node moving through an area as a HMM (Hidden Markov Model), which we augment: 1) to accommodate non-scalar probabilistic observations, and 2) to incorporate prior information on the user trajectory drawn from personal calendars. We cast the problem of IPS accuracy estimation to that of learning the parameters of the augmented HMM.
3. We propose a novel Expectation Maximization algorithm to solve the HMM learning problem. This algorithm extends the classic Baum-Welch algorithm [18] to account for non-scalar observations and calendar constraints. *We then show how the learnt HMM parameters can be used to evaluate the accuracy of co-located IPSs.*
4. We evaluate the performance of the learning algorithm in a variety of indoor environments. We show that in most cases the learnt accuracy of an IPS is far more accurate than the reported one. Finally, we demonstrate the first *IPS recommendation system*, which takes into account IPS accuracy, and recommends which IPS system to use where.

The rest of the paper is organized as follows: Section 5 overviews related work. Section 2 casts the problem into that of learning the parameters of a HMM. Section 3 explains how to solve the problem by extending the classic Baum-Welch algorithm.

[1] Note that in both cases the real trajectory is covered by the reported ellipses.

Section 4 evaluates the proposed and competing approaches in various indoor environments. Finally, Section 6 concludes the paper, and discusses ideas for future work.

2 Problem Formulation

In this section, we model the movement of a user in an indoor environment as a HMM[13], which we augment to account for probabilistic observations. We show how to incorporate into the model prior information about a user's schedule found in her calendar. We then show how the problem of assessing the accuracy of IPSs can be cast into that of estimating the parameters of the augmented HMM.

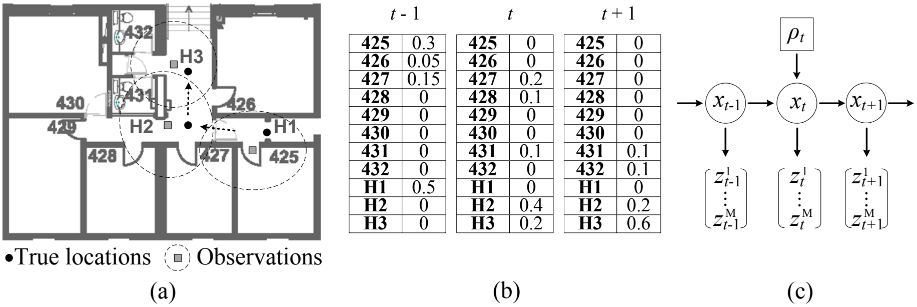
2.1 Definition of HMM Parameters

Suppose that M Indoor Positioning Systems (IPSs) are co-located in an indoor environment and provide positioning services to a user. Given the map of the environment, space is preprocessed into a finite set of N locations $L = \{l_k\}$, $1 \leq k \leq N$, e.g. different rooms or corridor segments in a building. In our experiments, the average size of the discrete locations is 3m×3m. At any discrete timestamp t, $1 \leq t \leq T$, an IPS reports an observation z_t of the user's true location x_t, which is typically unknown.

Hidden States: We model the true locations of the user as the hidden states. We denote the event that the true location of the user at time t is l_j as $x_t = l_j$, where $l_j \in L$. We also implicitly assume the actual trajectory $[x_1, ..., x_T]$ of a user is *Markovian*, i.e. $\mathrm{P}(x_t|x_1, ..., x_{t-1}) = \mathrm{P}(x_t|x_{t-1})$.

State Transition Probabilities and Initial State Distribution: The probability of a user moving from one location to another is governed by the transition probability distribution $A = \{a_{ij}\}$, where $a_{ij} = \mathrm{P}(x_{t+1} = l_j|x_t = l_i), 1 \leq i, j \leq N$. In an indoor environment, the transition probabilities a_{ij} largely depend on the underlying map. We assume that transitions can only be made from a given location to the same or adjacent locations with equal probability. Finally, the initial state distribution defines the probabilities that a user starts at each of the N possible locations: $\pi_i = \mathrm{P}(x_1 = l_i), 1 \leq i \leq N$, which is set to uniform in our experiments.

Observations: Unlike a standard HMM, which emits a single symbol per observation, in our case each observation is a *probability distribution*. For ease of exposition, let's initially assume that we have one IPS (referred to as the *system*). Under the assumption of discrete space, an observation z_t reported by the system at time t is a multinomial distribution $z_t = [p_t(1), ..., p_t(N)]$, where $p_t(k), 1 \leq k \leq N$, is the belief of the system that the user is in location l_k at time t. For example, the probabilistic observations (ellipses) of the system in Fig. 2(a) can be represented as vectors of probabilities shown in Fig. 2(b), where the tuple $\langle \mathrm{H2}, 0.4 \rangle$ at time t indicates that the system places the user at the corridor H2 with 0.4 probability. In the presence of M IPSs, the observation at time t becomes an $M \times N$ matrix $\{z_t^m\}$, where the m-th row is a vector of probabilities $[p_t^m(1), ..., p_t^m(N)]$ reported by the m-th IPS. For simplicity we assume there is no environment dynamics or external interferences that may affect the co-located IPSs,



t - 1	
425	0.3
426	0.05
427	0.15
428	0
429	0
430	0
431	0
432	0
H1	0.5
H2	0
H3	0

t	
425	0
426	0
427	0.2
428	0.1
429	0
430	0
431	0.1
432	0
H1	0
H2	0.4
H3	0.2

t + 1	
425	0
426	0
427	0
428	0
429	0
430	0
431	0.1
432	0.1
H1	0
H2	0.2
H3	0.6

(a) (b) (c)

Fig. 2. (a) Probabilistic observations (ellipses) at time $t-1$, t, and $t+1$. (b) Probabilistic observations can be represented as vectors of probabilities. (c) The augmented HMM.

and thus observations from different IPSs are conditionally independent given the real trajectory of the user, i.e. $\mathrm{P}(z_t^m, z_t^{m'}|x_t) = \mathrm{P}(z_t^m|x_t)\mathrm{P}(z_t^{m'}|x_t)$, for $1 \leq m, m' \leq M$.

Symbol Emission Probabilities: We now explain the symbol emission probabilities of the augmented HMM that admits probabilistic observations. We first assume a single IPS for simplicity. Let $B = \{b_j(k)\}$ be the stochastic matrix of emission probabilities. Each element $b_j(k)$ is defined as the average probability that the system assigns to location l_k, over all possible observations made when the user is actually in location l_j, $1 \leq j, l \leq N$. This probability distribution is different from the emission probabilities $B_{std} = \{b_j(k)_{std}\}$ in standard HMMs, where $b_j(k)_{std}$ is the probability that the user is observed in l_k when the actual location is l_j, $1 \leq j, k \leq N$. In standard HMM, likelihood of an observation at t given the state depends only on B_{std}, because z_t is *scalar* (a single location), and the sample space of z_t is the finite set L. In our model, the probability of observing z_t given the state can not be fully captured by B since z_t becomes a *vector* of symbols associated with different probabilities, which can not be modeled by any finite distributions.

When M IPSs coexist, under the assumption that the observations from different IPSs are independent conditioned on the states, the symbol emission probabilities for our model is a collection of M stochastic matrices $\boldsymbol{B} = \{B_m\}$, where each B_m is the individual symbol emission distribution for the m-th IPS.

In summary, our model $\lambda = (A, \boldsymbol{B}, \pi)$ is a generalization of standard HMMs, which supports probabilistic observations, and considers observations from multiple IPSs.

2.2 Accuracy Assessment as Parameter Estimation

Now we show how the problem of assessing the accuracy of co-located IPSs at different locations can be cast into that of learning the emission probabilities of the HMM λ defined above. Recall from Section 1 that the accuracy ϵ of an IPS at location x is defined as a function of the expected distance between the true and measured locations. Also, recall that the symbol emission probability $b_j(k)$ of an IPS represents the average probability associated with location l_k in the observations when the real state is $x = l_j$. We thus have:

$$\epsilon(l_j) = f(\sum_{l_k \in L} \mathrm{P}(l_k \text{ is reported in } z | x = l_j) \| l_k - l_j \|_2) = f(\sum_{l_k \in L} b_j(k) \| l_k - l_j \|_2) \quad (1)$$

where $\|l_k - l_j\|_2$ is the Euclidean distance between locations l_j and l_k. More generally, in a multi-IPS environment, the accuracy of the m-th IPS at different locations is governed by its symbol emission probabilities B_m. Since the actual user trajectories are unknown, we cannot empirically compute each B_m. The only information we can rely on is the observations Z of the M coexisting IPSs. Therefore, our goal is to find the model parameters λ that are most consistent with the observed data Z. Formally, this is given by the maximum likelihood estimate $\hat{\lambda}_{mle} = \arg\max_\lambda \mathrm{P}(Z|\lambda)$.

2.3 Incorporating User Provided Prior Knowledge

In this section, we explore the idea of using private information found in users' calendars, as clues on the user locations. We refer to such information as *priors*, and show how to incorporate them in our model. Formally, at a given timestamp t, a prior ρ_t is a distribution $\rho_t = [p_t(1), ..., p_t(N)]$ over the N discrete locations L, where each $p_t(k)$, $1 \leq k \leq N$, is the probability that the user is in location l_k at time t. The priors typically come from planned events, such as doctor appointments, flight boarding times and work meetings. Techniques of parsing the knowledge of such calendar information into priors are beyond the scope of this paper, and in our experiments, we adopt a simple approach that assigns a large probability at the location where the event is supposed to take place. For instance, a calendar entry of "meeting in room 100 at 3pm" can be interpreted as a prior $\rho_{3\text{pm}}$, with 0.9 probability associated with room 100, and the rest of the probability mass uniformly distributed elsewhere. We incorporate such knowledge as a prior distribution of the hidden states at time t, as shown in Fig. 2(c). Since we have extra information about the world, the optimization goal becomes finding the best model parameters $\hat{\lambda}_{mle} = \arg\max_\lambda \mathrm{P}(Z, \boldsymbol{\rho}|\lambda)$, which maximizes the joint likelihood of both the observations Z and the set of observed priors $\boldsymbol{\rho}$.

3 Proposed Algorithm

In this section, we first solve the parameter estimation problem formulated in the previous section, and then describe the complete algorithm that estimates the accuracy of IPSs. Recall from the previous section that our goal is to find the optimal model parameters $\lambda = (A, \boldsymbol{B}, \pi)$ that can best explain the observed data, which in our case, are the observations from co-located IPSs and the user-provided priors. The proposed algorithm extends the classic Baum-Welch Expectation Maximization algorithm to account for probabilistic observations and to exploit prior information drawn from users' calendars. We first describe how to derive the likelihood of the observed data with our model, and then show how to find the model parameters that maximize the likelihood.

3.1 Derive the Likelihood of Data

Given the observed data $O = \{Z, \boldsymbol{\rho}\}$, where Z are the observations from the M coexisting positioning systems and $\boldsymbol{\rho}$ the calendar priors provided by the user, the likelihood of data can be computed by summing out all possible hidden state sequences. We assume that given a hidden state sequence X (the true user trajectory), the priors $\boldsymbol{\rho}$ and observations Z from IPSs are conditionally independent. We also assume observation sequences from one IPS Z_m, $1 \leq m \leq M$, are conditionally independent of observations from the other IPSs given X. Then the likelihood of data is:

$$L_\lambda(O) = \sum_{X \in \mathcal{X}} \mathrm{P}(O, X|\lambda) = \sum_{X \in \mathcal{X}} \mathrm{P}(\boldsymbol{\rho}, X|\lambda) \prod_{m=1}^{M} \mathrm{P}(Z_m|X, \lambda) \tag{2}$$

Let us now explain how we derive components $\mathrm{P}(\boldsymbol{\rho}, X|\lambda)$ and $\mathrm{P}(Z_m|X, \lambda)$, starting with the first one that concerns the set of priors $\boldsymbol{\rho}$ provided by the user. Suppose that a prior ρ_t is given at timestamp t. ρ_t is a distribution $[q_t(1), ..., q_t(N)]$ over the set of locations L, where each $q_t(k)$ is the probability that the user is in location l_k, $1 \leq k \leq N$. If we assume the priors are trustworthy, then ρ_t essentially makes the hidden state visible, i.e. the distribution of the state x_t is known. In essence, the set of priors $\boldsymbol{\rho}$ *constrains* the dynamics of the model. For instance consider an extreme example where a prior ρ_t places probability 1 at a single location l_k, and 0 elsewhere. In this case from time $t-1$ to t, there will be no state transition to any locations other than l_k, and the state of our model at t is restricted to l_k only. Assume for simplicity that there is no prior at the beginning of time $t = 1$. Given a fixed state sequence $X = \{x_1, ...x_T\}$, where x_t is the true location of the user at t, the joint probability of the priors $\boldsymbol{\rho}$ and the state sequence X conditioned on the model λ is:

$$\mathrm{P}(\boldsymbol{\rho}, X|\lambda) = \mathrm{P}(x_1|\lambda) \prod_{t=2}^{T} \mathrm{P}(x_t|x_{t-1}) = \pi_{x_1} \prod_{t=2}^{T} \tilde{a}_{x_{t-1}x_t}(t) \tag{3}$$

where $\tilde{a}_{ij}(t)$ is the probability of transiting from state i at time $t-1$ to state j at time t under the constraints specified by the prior:

$$\tilde{a}_{ij}(t) = \begin{cases} a_{ij}, & \text{if no prior is observed at } t \quad \text{(4a)} \\ q_t(j), & \text{if } \rho_t \text{ is observed at } t \quad \text{(4b)} \end{cases}$$

Let us now derive component $\mathrm{P}(Z_m|X, \lambda)$ of Eqn. 2. Let z_t^m be the observation of the m-th IPS at time t. Recall that z_t^m is a vector of probabilities $[p_t^m(1), ..., p_t^m(N)]$, where $p_t^m(k)$, $1 \leq k \leq N$, indicates the belief of the m-th positioning system that the user is in location l_k at time t. The probability of the observation z_t^m given the state x_t and model λ, can be computed by summing over all possible locations l_k as follows:

$$\mathrm{P}(z_t^m|x_t, \lambda) = \sum_{k=1}^{N} p_t^m(k) \mathrm{P}(z_t^m = l_k|x_t, \lambda) = \sum_{k=1}^{N} p_t^m(k) b_{x_t}^m(k) \tag{5}$$

where recall that $b_{x_t}^m(k)$ denotes the emission probability of location l_k at state x_t by the m-th IPS. Then the probability of observation sequence Z_m given a fixed state sequence X and the model λ is the product of the observation likelihood at each timestamp:

$$\mathrm{P}(Z_m|X, \lambda) = \prod_{t=1}^{T} \sum_{k=1}^{N} p_t^m(k) b_{x_t}^m(k)) \tag{6}$$

3.2 Maximize the Likelihood of Data

Our goal is to find model parameter $\lambda = (A, \boldsymbol{B}, \pi)$ that maximizes the likelihood of observed data $L_\lambda(O)$. This can be achieved by the general EM algorithm, and in our case, the idea is to iteratively perform the following two steps of computation: (1) for the current model parameter λ, derive the expected log likelihood function $Q(\lambda', \lambda)$ over a new parameter λ'; and (2) find the λ' that maximizes the Q function, and in the next iteration substitute λ with this newly estimated λ'.

Now we show how to derive the new parameter $\lambda' = (A', \boldsymbol{B}', \pi')$ with the above procedure. Using Eqn. (2), the Q function can be represented as follows:

$$Q(\lambda', \lambda) = \sum_{X \in \mathcal{X}} \mathrm{P}(O, X|\lambda)\Big[\log(\mathrm{P}(\boldsymbol{\rho}, X|\lambda')) + \sum_{m=1}^{M} \log(\mathrm{P}(Z_m|X, \lambda'))\Big] \tag{7}$$

where the terms $\log(\mathrm{P}(\boldsymbol{\rho}, X|\lambda')$ and $\log(\mathrm{P}(Z_m|X, \lambda')$ can be further expanded using Eqns. (3) and (6). To get the parameter $\lambda' = (A', \boldsymbol{B}', \pi')$ that maximizes Q, we use the Lagrange multiplier method, and set the partial derivatives of the resulting objective function to zero, which yields:

$$\pi_i' = \frac{\alpha_1(i)\beta_1(i)}{\sum_1^N \alpha_1(i)\beta_1(i)}; \quad a_{ij}' = \frac{\sum_2^T \Big[\alpha_{t-1}(i)\tilde{a}_{ij}(t)\Big(\prod_{m=1}^{M}\sum_{k=1}^{N} p_t^m(k)b_j^m(k)\Big)\beta_t(j)\Big]}{\sum_2^T \alpha_{t-1}(i)\beta_{t-1}(i)};$$

$$b_j^m(k)' = \frac{1}{\sum_1^T \alpha_t(j)\beta_t(j)}\Big\{\pi_j\Big[p_1^m(k)b_j^m(k)\prod_{m'\neq m}^{M}\sum_{k=1}^{N} p_1^{m'}(k)b_j^{m'}(k)\Big] + \tag{8}$$

$$\sum_2^T\Big[\sum_{i=1}^{N}\alpha_{t-1}(i)\tilde{a}_{ij}(t)\Big]\Big[p_1^m(k)b_j^m(k)\prod_{m'\neq m}^{M}\sum_{k=1}^{N} p_1^{m'}(k)b_j^{m'}(k)\Big]\beta_t(j)\Big\}$$

where the variables α and β in the right hand sides of the above are *forward* and *backward* variables, and the definitions of these variables are extended from the original Baum-Welch algorithm to account for probabilistic observations and to exploit prior information about calendar constraints.

Forward Variables: We define the forward variable $\alpha_t(j)$ as the joint probability of having the probabilistic observations and priors in the first t steps, and of landing at state l_j at time t given the model:

$$\alpha_t(j) = \mathrm{P}(\rho_{1:t}, \boldsymbol{z}_{1:t}, x_t = l_j|\lambda) = \Big[\sum_{i=1}^{N}\alpha_{t-1}(i)\tilde{a}_{ij}(t)\Big]\prod_{m=1}^{M}\sum_{k=1}^{N} p_t^m(k)b_j^m(k) \tag{9}$$

where $b_j^m(k)$ is from the symbol emission probability distribution of the m-th IPS, $p_t^m(k)$ is the probability assigned to location l_k by the m-th IPS in its observation at time t, and $\tilde{a}_{ij}(t)$ is the probability of transiting from state l_i (at time $t-1$) to l_j (at time t) under the constraints specified by the prior (see Eqn. 4).

Algorithm 1. HMM Inference

```
Initialize $\alpha_1(i) \leftarrow \pi_i$
for $t = 2 : T$ do
    Compute $\alpha_t(i)$ as in (9)
end for
Compute $L_\lambda(O) \leftarrow \sum_{i=1}^{N} \alpha_T(i)$
Initialize $\beta_T(i) \leftarrow 1$
for $t = T - 1 : 1$ do
    Compute $\beta_t(i)$ as in (10)
end for
```

Algorithm 2. Accuracy Estimation

```
while $L_\lambda(O)$ does not converge do
    Perform inference as in Algo. 1
    Compute $\lambda'$ as in (8) and do: $\lambda \leftarrow \lambda'$
end while
for each IPS $ps_m$ do
    for location $l_j \in L$ do
        $\epsilon^m(l_j) \leftarrow \sum_{k=1}^{N} b_j^m(k) \|l_k - l_j\|_2$
    end for
end for
```

Backward Variables: We define the backward variable $\beta_t(i)$ as the joint probability of observations and priors from time $t+1$ to T given state l_i at time t and the model λ:

$$\begin{aligned} \beta_t(i) &= \mathrm{P}(\rho_{t+1:T}, \boldsymbol{z}_{t+1:T} | x_t = l_i, \lambda) \\ &= \Big[\sum_{j=1}^{N} \tilde{a}_{ij}(t+1) \prod_{m=1}^{M} \sum_{k=1}^{N} p_{t+1}^{m}(k) b_j^m(k)\Big] \beta_{t+1}(j) \end{aligned} \tag{10}$$

The procedure of evaluating the forward and background variables are shown in Algo. 1, which is equivalent with the above inductive definitions. Now we are in the position of presenting the accuracy estimation algorithm.

3.3 The Accuracy Estimation Algorithm

As shown in Algo. 2, in each iteration the accuracy estimation algorithm first calls the inference algorithm (Algo. 1), which scans the observed data (observations from the IPSs and priors from the user) twice to compute the forward and backward variables. Then it computes the new parameters of the model that maximize the likelihood of data, and uses the newly estimated parameters for the next iteration, until the likelihood of data converges. Then for each IPS, the algorithm estimates its accuracy at a location l_j by computing the expected distance between the true and measured locations, from the learnt symbol emission probability distributions.

From Eqn. 9 and Eqn. 10, we can see the complexity of HMM inference (Algo. 1) is on the order of $O(MN^3T)$, where M is the number of co-located IPSs, N is the amount of discrete locations, and T is the length of the observation sequences. The complexity of accuracy estimation algorithm (Algo. 2) is then $O(IMN^3T)$, where I is the number of iterations until converge. In practice M is typically small and the probabilistic observations $z_t^m = [p_t^m(1), ..., p_t^m(N)]$ from an IPS can be very sparse. Therefore with some optimizations our algorithm is not prohibitively expensive comparing to the classic Baum-Welch algorithm, which is of order $O(IN^2T)$.

4 Evaluation

4.1 Experiment Setup

We evaluate our proposed algorithm in two indoor settings. One is the 4th floor of a CS department, which is a typical office environment. The other is the basement of a building, which includes seminar rooms, lecture theaters and other common areas. Each space is outfitted with multiple co-located indoor positioning systems, as shown in Fig. 3. Four different WiFi-based positioning systems, ps_1 to ps_4, operate on the 4th floor, and three systems: ps_5 to ps_7 on the basement. Each IPS owns a set of WiFi basestations at different areas of the space. The basestations periodically broadcast WiFi beacons, which are then picked up by the mobile devices carried by the users. An IPS estimates location by combining RSSI measurements from its basestations with inertial data collected from a foot mounted IMU. We track two users carrying different mobile devices (a Nexus S and an Asus TF201 tablet) during their work time and collect data for 20 days on the 4th floor and 4 days in the basement. The ground truth is provided by the users: the floor plans are presented at their devices and they touch the actual positions they are visiting to log the coordinates on the maps.

Our proposed approach of learning IPS accuracies, referred to as the ***Learning Algorithm (LA)***, is currently implemented in Matlab. We compare our approach with the following three algorithms:

(a) ***Oracle Algorithm (OA)*** has access to the ground truth and can directly compute the actual accuracy $\epsilon_{oracle}(j)$ of an IPS at location l_j.

(b) ***Report-based Algorithm (RA)*** estimates accuracy based on reported observations z_t from a positioning system. The algorithm treats the location l_j with the largest probability according to z_t as the actual position of the user, and estimates IPS accuracy as the expected distance between the observation z_t and location l_j.

(c) ***Fusion-Based Algorithm (FA)*** takes observations from all co-located IPSs into account. For M IPSs, at time t the algorithm finds the centroid location l_j according to the observations $z_t^1, ..., z_t^M$. Each IPS then estimates its accuracy as the expected distance between its own observation z_t and location l_j.

We evaluate the above competing algorithms against the following two metrics:

(1) ***Estimation Error*** EE. For an accuracy estimation algorithm, EE is defined as the mean squared error between the estimated accuracy ϵ and the ground truth accuracy ϵ_{oracle}: $EE = \frac{1}{N} \sum_{j=1}^{N} (\epsilon(j) - \epsilon_{oracle}(j))^2$. We use EE_L, EE_R and EE_F to denote the estimation errors of *LA*, *RA* and *FA* respectively.

(2) ***Localization Error*** LE. Given estimates of IPS accuracy at different locations, we can partition the space into regions, where an IPS is consistently better than the others. The user can then choose the best available IPS in a given area. As different algorithms provide different estimates of IPS accuracy, they partition the space differently, and suggest a different IPS usage. They thus have a different localization error, which is defined as the mean squared error between the ground truth $\{x_i\}$ and estimated

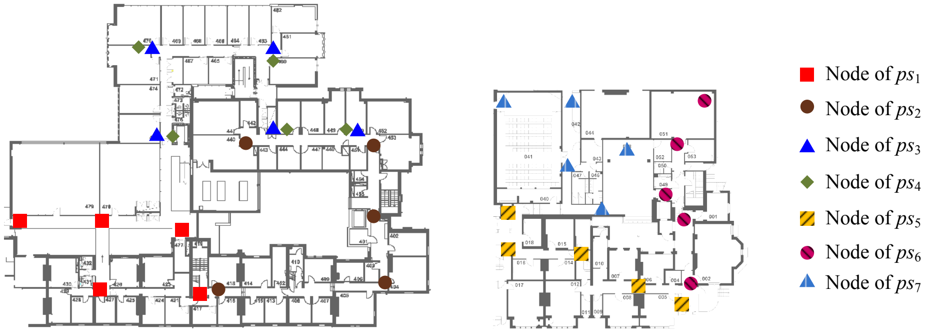
trajectory $\{\hat{x}_i\}$: $LE = \frac{1}{T}\sum_{t=1}^{T}(\hat{x}_t - x_t)^2$. We use LE_L, LE_R and LE_F to denote the localization error when space partitioning is based on *LA*, *RA* and *FA* respectively.

4.2 Experiment Results

IPS Accuracy and Ranking Varies Over Space: The first experiment shows that the accuracy of a positioning system varies significantly over space, and that there may not exist a clear winner, i.e. an IPS that is consistently better than the others at all locations. In this experiment, three different IPSs ps_1, ps_2 and ps_3 are running in parallel on the 4th floor. As shown in Fig. 3(a), ps_1 has most of its nodes at the bottom left corner, ps_2 dominates on the right side, and ps_3 on the top side of the floorplan. Figs. 4(a1)∼Fig. 4(c1) show the estimated trajectories and reported accuracy (ellipses) when each of the three IPSs is tracking a user along the corridor. Figs. 4(a2)∼Fig. 4(c2) show the real accuracies of the three positioning systems at different locations. It is obvious that none of the IPS is globally better than the other, and each one tends to excel in the area where it has the denser infrastructure. Fig. 4(d1) shows the most accurate IPS at each trajectory location. To avoid switching too often, we transform Fig. 4(d1) to Fig. 4(d2), by applying a clustering algorithm that finds broader regions dominated by an IPS. Such a space partitioning would be useful to decide which IPS to use where.

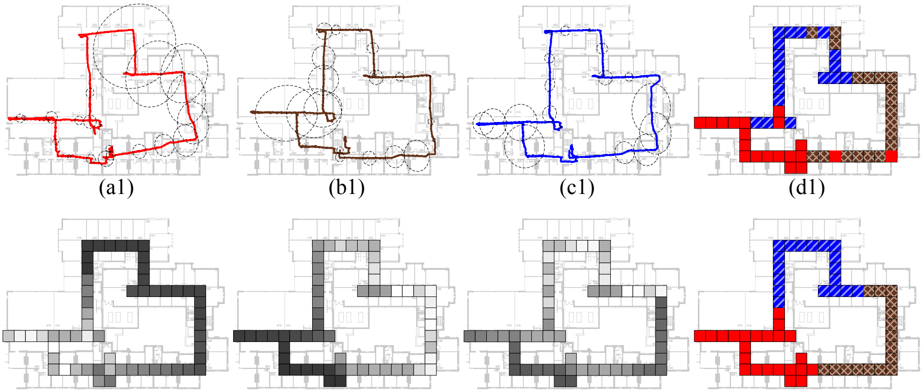
Estimation Performance: The goal of this experiment is to evaluate how the estimation error of our algorithm (*LA*) fares compared to that of competing algorithms (*RA* and *FA*). In this experiment we use the same set up as in the first experiment, i.e. the three co-located IPSs on the fourth floor of a CS building. Fig. 5(a) shows the estimation error EE_L of our proposed algorithm at the end of each learning iteration. Whereas the error initially decreases as expected, it starts to increase again after the third iteration. This is due to the common problem of overfitting. We address this problem by randomly selecting a set of IPS observations (data collected in 5 days) as a training set that we use for learning, and a test set (data collected in another 5 days) that we used to detect when to terminate the learning process. As shown in Fig. 5(b), we terminate learning in the third iteration when the log likelihood of the test data starts to decrease. Fig. 5(a) shows that, when we pause learning in iteration 2, the error of the proposed algorithm (EE_L) is 3-4 times smaller than that the fusion-based error EE_F, and eight times smaller than the report-based error (EE_R). Fig. 5(c) shows that the estimation error of the proposed algorithm (*LA*) can be further improved significantly by incorporating prior information on the location of the user, which can be drawn from user calendars. The x-axis shows the percentage of timestamps for which we have prior information. As expected, the more the priors on user positions, the smaller the estimation error of our algorithm. For example, when priors are available at 10% of the timestamps, the estimation error of *LA* is $1.51m^2$, which is about 30% smaller than the case where no prior is used ($2.14m^2$).

Ranking Performance: In the previous experiment, we showed that the proposed algorithm outperforms competing approaches in terms of estimating IPS accuracy. We now show how this affects the user's ability to rank IPSs, and to choose the best one at each location. We examine four different scenarios depicted in the four rows of Fig. 6 respectively. The first scenario (row 1) includes positioning systems ps_1, ps_2 and ps_3



(a) (b)

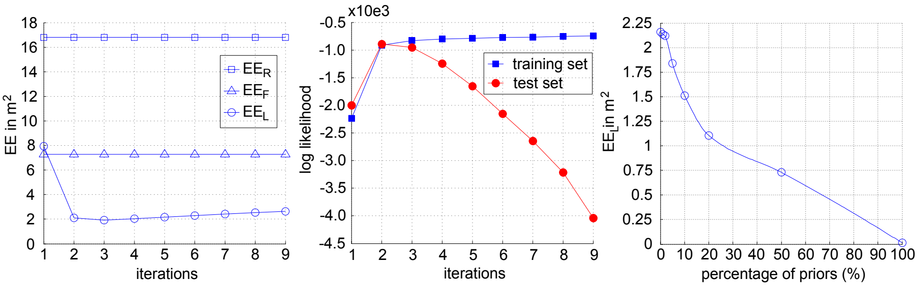
Fig. 3. (a) Experiment setup on the 4th floor. (b) Experiment setup on the basement.



(a1) (b1) (c1) (d1)

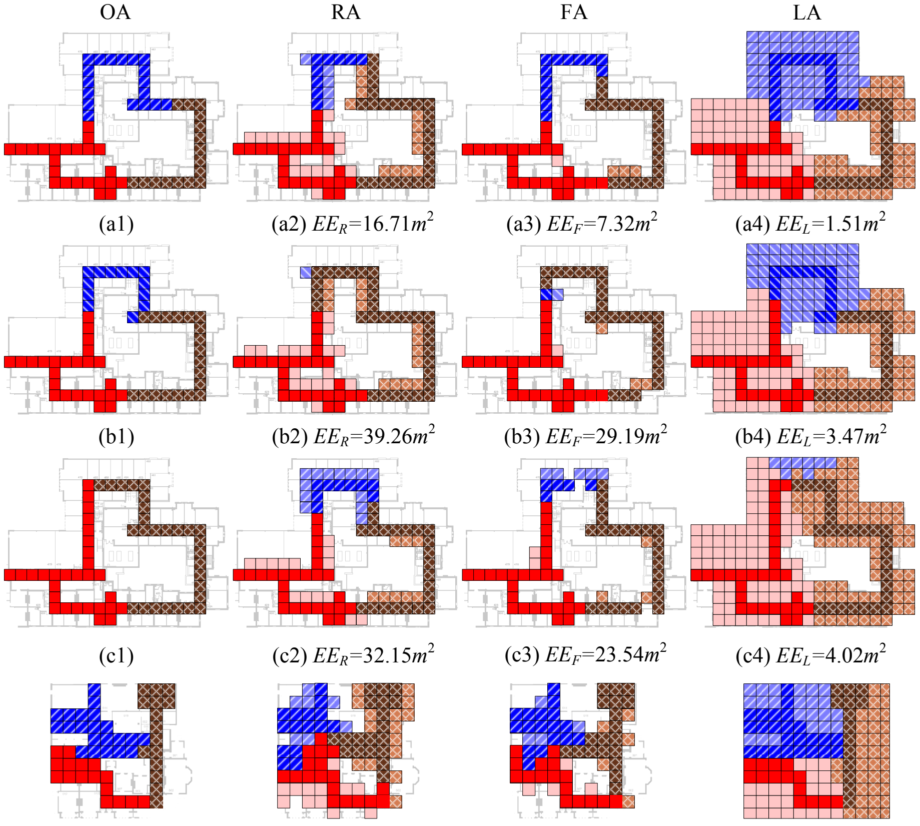
(a2) (b2) (c2) (d2)

Fig. 4. (a1) Estimated trajectory from ps_1. (a2) Real accuracy of ps_1 (darker means less accurate) (b1) Estimated trajectory from ps_2. (b2) Real accuracy of ps_2 (darker means less accurate) (c1) Estimated trajectory from ps_3. (c2) Real accuracy of ps_3 (darker means less accurate) (d1) Each traversed location is labeled with the locally most accurate IPS. (d2) Space is clustered into regions where each of them is dominated by an IPS.



(a) (b) (c)

Fig. 5. (a) Log likelihood of data (both training and test) at different learning iterations. (b) Estimation error of *RA*, *FA* and *LA*. (c) Estimation error of the proposed *LA* as we vary the amount of priors used in the algorithm.



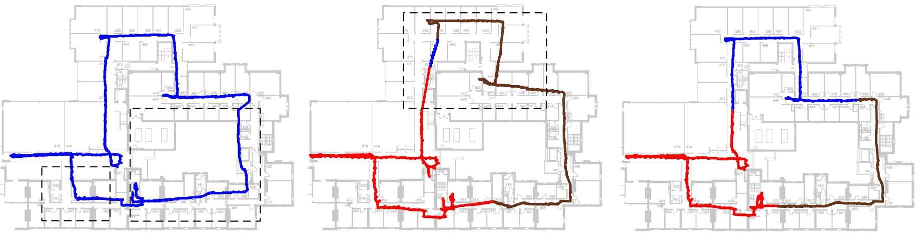
(a1) (a2) EE_R=16.71m^2 (a3) EE_F=7.32m^2 (a4) EE_L=1.51m^2

(b1) (b2) EE_R=39.26m^2 (b3) EE_F=29.19m^2 (b4) EE_L=3.47m^2

(c1) (c2) EE_R=32.15m^2 (c3) EE_F=23.54m^2 (c4) EE_L=4.02m^2

(d1) (d2) EE_R=23.62m^2 (d3) EE_F=10.21m^2 (d4) EE_L=2.97m^2

Fig. 6. Maps showing IPS preferences and estimation errors of different algorithms: *OA*, *RA*, *FA* and *LA*, with the trajectory locations highlighted and empty blocks indicate no information there. Row 1: Loc = 4F, User = Nexus S, IPSs = { ps_1, ps_2, ps_3}; Row 2: Loc = 4F, User = Nexus S, IPSs = { ps_1, ps_2, ps_4}; Row 3: Loc = 4F, User = TF201, IPSs = { ps_1, ps_2, ps_3}; Row 4: Loc = 0F, User = Nexus S, IPSs = { ps_5, ps_6, ps_7}. In all scenarios we assume for 10% of the timestamps we have prior information on locations of the user, which is used by *LA*.



(a) (b) (c)

Fig. 7. (a) Trajectory from the single best IPS. Dashed rectangles indicate areas where localization error is larger than $4m^2$. (b) Switching according to *FA*, and $LE_F = 5.33m^2$. (c) Switching according to *LA*, and $LE_L = 2.07m^2$.

deployed on the 4th floor and a user traversing it holding a Nexus S device. Observe that the proposed algorithm (Fig. 6(a4)) manages to partition the space in almost the same way as the oracle (Fig. 6(a1)) that has perfect knowledge of IPS accuracies. In this scenario, the two competing algorithms (*RA* and *FA*) perform reasonably well (Figs. 6(a2) and 6(a3)), with the reported approach making a few errors in the upper part of the user's trajectory. In the second scenario (row 2), we use a different set of co-located positioning systems: ps_1, ps_2 and ps_4, where ps_4 has the same infrastructure as ps_3 but more conservative noise profile (with twice as high gyroscope and accelerometer variances). Although this does not change the ground truth much, it throws off the two competing algorithms (*RA* and *FA*), which now refrain from using ps_4 on the top part of the trajectory. Notice however that the proposed (*LA*) approach still perceives ps_4 to be the best there and still matches the *real* partitioning. In the third scenario, (row 3) we use the same setup as in the first scenario (row 1), except that the user now carries an Asus TF201 tablet instead of the Nexus S phone. Whereas ps_1 and ps_2 use different radio propagation models for the two devices, ps_3 has been tuned for phone users only. Hence, it performs poorly on the tablet and ceases to perform well on the top part of the user's trajectory (see Fig. 6(c1)). Notice that only the proposed algorithm (*LA*) manages to detect the drop in ps_3's accuracy and partition the space correctly, whereas the other two algorithms (*RA* and *FA*) still prefer ps_3 on the upper part of the trajectory. In the fourth scenario (row 4), we change the venue to the basement, which has more open areas than the the 4th floor. Here, the competing algorithms (*RA* and *FA*) erroneously show a preference on using ps_6 not only on the right hand side of the floor plan, but also further into the center (Figs. 6(d2) and 6(d3)). The proposed algorithm (*LA*) (Fig. 6(d4)) again partitions the space as if it had access to the real IPS accuracies (Fig. 6(d1)).

Localization Performance: The final experiment is set up to investigate how ranking performance has a knock-on effect on localization performance. That is, if a user relies on the space partitioning generated by the proposed algorithm, to switch from one IPS to another, will they be localized more accurately, than if they used that of the competing algorithms (*RA*/*FA*)? Here we compare only with *FA*, since it has proved to be superior to *RA*. We also compare with the strawman approach of using one IPS everywhere, the one that has the best overall performance in terms of localization error LE. Fig. 7 shows that switching IPSs is generally better than using the same IPS everywhere. It also shows that the localization error of the user that relies on the space partitioning of the proposed algorithm ($LE_L = 2.07m^2$) is reduced by more than 60% when compared to that of the best competing algorithm ($LE_F = 5.33m^2$).

5 Related Work

In this section we overview existing work and discuss how it relates to our problem. A large body of research in sensor networks has applied Bayesian State Estimation in localization- and tracking-related applications. For example, HMMs are commonly used to find the most likely trajectory that accounts for measurement noise and known map constraints (e.g. in VTrack, EasyTracker and CTrack systems) [6,10,15,11,1]. In addition, Bayesian filters, such as Kalman and particle filters have also been broadly used for online position estimation both in sensor networks and robotics research [3,14].

In stark contrast to the aforementioned HMM-based and filter-based approaches, we do not assume knowledge of the emission probabilities (a.k.a. observation models) of the systems used for localization. In addition, we have no ground truth data from measurement campaigns, which we could use to empirically learn the emission probabilities. Whereas they address the *inference* problem of estimating the most likely trajectory of a mobile node, we address the *learning* problem of estimating the IPS emission models that best explain their observations.

Another related area of research is that of cost-constrained data acquisition in sensor networks. Most work in this area explores the trade-off between the quality of information and the costs incurred in retrieving the sensor data, and it can be broadly divided into two branches. The first branch assumes that sensor readings concern the same variable of interest and are typically correlated over space. The goal is then to place sensors, or select sensors to be active, at informative locations, e.g. using Gaussian Process (GP) models of the sensed phenomenon [8,2,9,4]. The second branch of cost-bounded data acquisition considers heterogeneous sensors, and applications where the variable of interest may not be directly measured by a sensor, but rather derived from a variety of sensor measurements using a prediction model. An example is the study of selecting suitable regression models to minimize the prediction error given a cost budget [16]. Our work is different from the above cost-constrained data acquisition techniques in that we do not possess knowledge of the true locations where the noisy measurements were made, and thus cannot use it to train Gaussian Process or other regression models.

Our work is also related to fact finding techniques in information networks [7,19,12]. In these networks, sources and assertions are represented as nodes, and each fact "source i made an assertion j" is represented by a link. Nodes are then assigned credibility scores in an iterative manner: for example, in a basic fact finder [7], an assertion's score is set to be proportional to the number of its sources, weighted by the sources' scores; similarly, a source's score is set to be proportional to the number of the assertions it made, weighed by the assertions' scores. Whereas we share the same goal of assessing the credibility of different sources (in our case, IPSs), we cannot directly apply fact finding techniques for three reasons: First, the credibility of an IPS is not the same across different locations, which means that it does not make sense to try to evaluate a single credibility score for an IPS across the entire area of interest. Second, the true user location is not known, so we cannot define a source, as an (IPS,location) pair. Finally, fact finding techniques tend to work well when a large number of sources are used (e.g. social sensing), whereas in our case our goal is to assess and rank a limited number of positioning systems (typically, less than 5).

Our work falls into the class of maximum likelihood estimation (MLE) approaches to ascertaining sensor reliability. In this sense, it bears close resemblance to [17], which estimates both the correctness of measurements and the reliability of participants in social sensing applications by solving an expectation maximization problem. Similar to [17], we also adopt an expectation maximization (EM) approach to get the maximum likelihood estimate of model parameters. However, the underlying probabilistic model, the problem formulation, and the specific derivation of the EM algorithm are very different. Unlike [17], we consider a dynamic model, with probabilistic observations, and cast our problem in that of learning the parameters of a HMM. We thus use a different EM scheme (an extension of Baum-Welch algorithm), which is suitable for HMMs.

6 Conclusion and Future Work

This paper contributes a novel approach of estimating the accuracies of co-located IPSs. The estimation problem is cast into parameter learning of an augmented HMM, and solved by a variation of the EM algorithm. We show that prior information provided by the users can significantly improve the quality of estimation. Experiments in four indoor scenarios shows that our approach outperforms the competing ones both in terms of estimation and localization error. Future work includes supporting continuous observations and on-line learning, utilizing more types of priors, and working with more complex positioning systems, whose performance does not only vary with space.

Acknowledgements. The authors would like to acknowledge the support of the EPSRC through project FRESNEL EP/G069557/1, and thank the anonymous reviewers and our shepherd Philippe Bonet for their helpful feedback.

References

1. Biagioni, J., Gerlich, T., Merrifield, T., Eriksson, J.: Easytracker: automatic transit tracking, mapping, and arrival time prediction using smartphones. In: Proc. SenSys (2011)
2. Das, A., Kempe, D.: Sensor selection for minimising worst-case prediction error. In: Proc. IPSN (2008)
3. Fox, D., Hightower, J., Liao, L., Schulz, D., Borriello, G.: Bayesian filtering for location estimation. In: IEEE Pervasive Computing (2003)
4. Hare, G.M.P.O., Muldoon, C., Trigoni, N.: Combining sensor selection with routing and scheduling in wireless sensor networks. In: Proc. DMSN in Conjunction with VLDB (2011)
5. Huang, J., Millman, D., Quigley, M., Stavens, D., Thrun, S., Aggarwal, A.: Efficient, generalized indoor wifi graphslam. In: Proc. ICRA (2011)
6. Hummel, B.: Map matching for vehicle guidance. In: Dynamic and Mobile GIS: Investigating Space and Time. CRC Press (2006)
7. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM 46(5) (1999)
8. Krause, A., Guestrin, C., Gupta, A., Kleinberg, J.: Near optimal sensor placements: maximising information while minimising communication cost. In: Proc. IPSN (2006)
9. Krause, A., Singh, A., Guestrin, C.: Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. J. Mach. Learn. Res. 9 (2008)
10. Krumm, J., Letchner, J., Horvitz, E.: Map matching with travel time constraints. In: SAE World Congress (2007)
11. Newson, P., Krumm, J.: Hidden markov map matching through noise and sparseness. In: Proc. GIS (2009)
12. Pasternack, J., Roth, D.: Knowing what to believe (when you already know something). In: Proc. COLING (2010)
13. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Pearson Education (2003)

14. Schmid, J., Beutler, F., Noack, B., Hanebeck, U., Müller-Glaser, K.D.: An experimental evaluation of position estimation methods for person localization in wireless sensor networks. In: Proc. EWSN (2011)
15. Thiagarajan, A., Ravindranath, L., LaCurts, K., Madden, S., Balakrishnan, H., Toledo, S., Eriksson, J.: Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In: Proc. SenSys (2009)
16. Wang, D., Ahmadi, H., Abdelzaher, T., Chenji, H., Stoleru, R., Aggarwal, C.: Optimizing quality-of-information in cost-sensitive sensor data fusion. In: Proc. DCOSS (2011)
17. Wang, D., Kaplan, L., Le, H., Abdelzaher, T.: On truth discovery in social sensing: a maximum likelihood estimation approach. In: Proc. IPSN (2012)
18. Welch, L.R.: Hidden Markov Models and the Baum-Welch Algorithm. IEEE Information Theory Society Newsletter 53(4) (2003)
19. Yin, X., Han, J., Yu, P.: Truth discovery with multiple conflicting information providers on the web. In: Proc. KDD (2007)