

Accuracy Estimation for Sensor Systems

Hongkai Wen, Zhuoling Xiao, Andrew Markham, and Niki Trigoni

Abstract—In most sensing applications, the measurements generated by sensor networks are *noisy* and usually annotated with some measure of uncertainty. The question that we address in this paper is how to estimate the accuracy of these uncertain sensor measurements. Existing studies on estimating the accuracy of uncertain measurements in real sensing applications are limited in three ways. First, they tend to be application-specific. Second, they typically employ *learning* techniques to estimate the parameters of sensor noise models, and ignore alternative state estimation approaches without learning. Third, they do not explore whether exploiting the dynamics of the monitored state can yield significant benefits. We address the above limitations as follows: we define the accuracy estimation problem in a general manner that applies to a broad spectrum of application scenarios. We present a general framework to address this problem, and show that the proposed framework can be implemented in a number of different ways. We evaluate and compare the different implementations in the context of two real sensing scenarios, and discuss how they trade accuracy for computation cost, and how this trade-off largely depends on the user's knowledge of the application scenario.

Index Terms—Accuracy estimation, sensor systems

1 INTRODUCTION

WITH sensor technology gaining maturity and becoming ubiquitous, we are experiencing an unprecedented wealth of sensor data. In most sensing applications, users receive sensor measurements, which are prone to error. As a result, they often contain some measure of uncertainty, such as the confidence intervals or distribution variances, which will be hereafter referred to as *probabilistic measurements*. The presence of noise in sensor data has motivated a lot of research in areas of sensor networks, mobile robotics and machine learning over the last two decades. This work can be broadly categorized into two classes: 1) *state estimation*: the first class has assumed known models of measurement noise, and has explored the inference algorithms to estimate the state of the monitored phenomenon; 2) *parameter estimation*: the second class has employed learning techniques to estimate the parameters of the measurement noise that best explains the generated sensor measurements. In this paper we investigate a related problem that lies in-between the two canonical problems of state estimation and parameter estimation, and which we will hereafter refer to as *accuracy estimation*. We start with a probabilistic measurement (e.g. temperature value with a 95 percent confidence interval), and our objective is to estimate the accuracy of this measurement, i.e. how far it lies from the ground truth given certain distance metric.

This is an important and timely problem in a number of different sensing applications. First, knowing how accurate the measurements of a sensor system are is paramount to deciding whether to use or pay for the service it offers. For example, if a positioning system consistently places a user

at locations far away from the ground truth, the users should have a way of detecting the poor accuracy of this sensing service. Second, a user may be faced with the choice of selecting among multiple co-located sensor systems that offer a similar service (e.g. a WiFi-based versus an FM-based indoor tracking [1] system in the same building). In this case, they should be in a position to compare or rank the accuracy of different systems. Third, when a sensor system is first deployed, the administrator typically assumes a default noise model for the networked sensors. To detect when a sensor starts malfunctioning, it is critical to be able to assess when the accuracy of the measurements drops significantly below a certain threshold. Finally, the emergence of social sensing applications has raised the challenge of estimating the trustworthiness of human participants. When people report some observations (say, estimated air pollution levels), it is key to be able to assess the accuracy of the reported data.

The common denominator in the above examples is that multiple data sources generate probabilistic sensor observations (e.g. mean and variance pairs) about certain physical signals, and the goal is to estimate the accuracy of these observations. This problem is challenging in many ways. First in most sensing applications, the ground truth of the measured signal is *unknown*, and thus the accuracy of the sensor measurements cannot be evaluated empirically. Second, the accuracy of a sensor system is typically *context-dependent*, and we cannot rely on the accuracy of the system in one context to infer its accuracy in another. Third, due to security and privacy concerns, many sensor systems tend to present themselves as *black boxes*, by hiding their implementation and deployment details, such as sensor locations or sensor types. Therefore we have little or no knowledge about how they work or where their sensors have been deployed, or how optimistic/pessimistic they are in estimating uncertainty.

Finally, the reported accuracy information encoded in a probabilistic sensor measurement, e.g. the confidence interval or the error ellipse, is not always a reliable indicator of

• The authors are with the Department of Computer Science, University of Oxford, Oxford, OX1 3QD, U.K. E-mail: {hongkai.wen, zhuoling.xiao, andrew.markham, Niki.Trigoni}@cs.ox.ac.uk.

Manuscript received 31 Mar. 2014; accepted 5 Aug. 2014. Date of publication 14 Sept. 2014; date of current version 1 June 2015.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2014.2352262

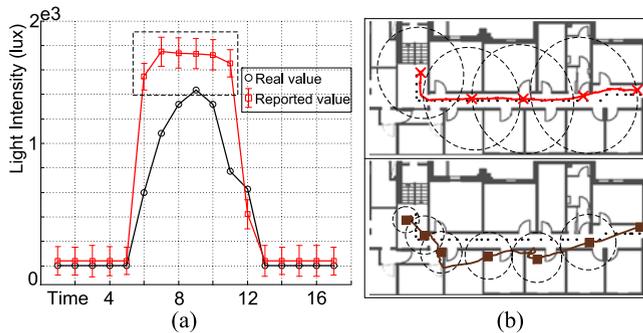


Fig. 1. (a) At the highlighted part, the real values of the light intensity consistently fall out of the reported confidence intervals. (b) The reported accuracy (error ellipses of the position measurements) is misleading as to which is the most accurate sensor system.

its real accuracy. We observe two general cases where this happens in practice, and provide illustrative examples in real sensing applications. *Case 1*: the ground truth consistently falls out of the reported accuracy bounds. For instance consider the light intensity data reported by a sensor system, as shown in Fig. 1a. The system consistently underestimates the error in its measurements during the highlighted time period. Another example could be the safety-critical environments, where the sensor systems may be compromised to consistently report wrong estimates with very high reported accuracy. *Case 2*: the size of the accuracy bounds reported by different sensor systems can lead to wrong conclusion about which one is more accurate. For instance, consider the indoor positioning scenario where two real systems are reporting the positions of the user, as shown in Fig. 1b. The top trajectory in Fig. 1b is generated by an inertial tracking system, while the bottom one is from the commercial WiFiSLAM system [2]. Given the ground truth (the black dotted line in both cases), one can immediately see that the estimated trajectory on top is far better than the bottom one, but the reported accuracy (indicated by the dashed ellipses) is significantly lower. Therefore the aim of this paper is to address these challenges, and provide a general framework to solve the accuracy estimation problem for sensor systems.

To our knowledge, there have been few efforts to tackle this problem. For example, the work in [3], [4] and [5] estimates the correctness of measurements reported by human participants in social sensing applications by solving the expectation maximization (EM) problem. Our previous work in [6] shows how to assess the accuracy of co-located positioning systems by extending the Baum-Welch algorithm—a special case of EM algorithm for dynamic systems. These papers use algorithms tailored to the specific application scenarios, and employ *learning* (EM) techniques. There is currently no systematic study that provides a general solution for accuracy estimation and compares the algorithms proposed for different applications in a common experimental setup. In this paper we show that the problem of accuracy estimation for sensor systems can be addressed with a general framework, whose components can be implemented by a number of different techniques. We advocate that *learning* is not the only option for accuracy estimation, but *inference* techniques are equally viable alternatives, which

should not be confined to their traditional use in state estimation problems. We show that, in certain cases, inference techniques can offer more attractive trade-offs between computational cost and estimation accuracy than their learning counterparts. To summarize, the contributions of this paper are as follows:

- 1) We formulate the problem of accuracy estimation for sensor systems in a general manner, which covers a broad spectrum of sensing applications. We have motivated the problem in the context of pricing sensing services, ranking them if they are competing for the same users, detecting faults, and establishing trustworthiness of different individuals in social sensing.
- 2) We propose a general framework to address the formulated accuracy estimation problem, which contains four layers: pre-processing, state estimation, accuracy estimation and accuracy indexing. We show that by passing sensor observations through those layers, the proposed framework is able to assess the accuracy of the reported measurements, and further reason about the accuracy of the sensor systems.
- 3) We demonstrate that the state estimation layer can be implemented in various ways, and create a taxonomy of approaches ranging from simple voting schemes, to inference-based and learning-based techniques. We show how inference and learning techniques can be further sub-divided into those that exploit knowledge of the dynamics of the monitored process, and those that do not. We also show that any prior information on the monitored phenomenon can be easily incorporated into inference and learning techniques in both the static and dynamic cases.
- 4) We propose two accuracy metrics for the accuracy estimation layer, one based on *proximity* and the other on *similarity* between the probabilistic measurements and the monitored state. Those metrics evaluate the accuracy of sensor measurements from different perspectives, and thus are suitable for different classes of applications. For accuracy indexing layer, we propose a scheme that can build accuracy indices of sensor systems by aggregating and interpolating the accuracy of reported sensor measurements over given attributes.
- 5) We perform a systematic experimental evaluation of the proposed accuracy estimation framework with two real sensor datasets, one containing position sensor measurements in an indoor environment, and the other containing light intensity sensor measurements. We show how different implementations of the state estimation layer can influence the performance of the proposed framework, in terms of accuracy estimation quality, computational cost, sensitivity to the prior state distributions and coexisting sensor systems, etc. We also show that those techniques have their own merits in different contexts, and by carefully selecting and tuning them, we can achieve the desired trade-offs in different sensing scenarios.

The remainder of this paper is organized as follows: Section 2 formulates the problem of accuracy estimation in

the context of one or more coexisting sensor systems, and presents the high level idea of the accuracy estimation framework. Sections 3, 4, 5 and 6 describe the layers of pre-processing, state estimation, accuracy estimation and accuracy indexing of the proposed framework in more detail. Section 7 presents the experimental evaluation of the proposed accuracy estimation framework in the context of two real sensor datasets, and provides a comprehensive discussion of the experiment results. Section 8 overviews related work, while Section 9 concludes the paper and discusses future work.

2 PROBLEM FORMULATION

2.1 Model and Assumptions

Monitored states. Let x_t be the real value of the monitored signal (e.g. temperature of a room or position of user) at a given discrete time t , where the timestamps $t = 1 : T$ are a totally ordered set. In the following text, we refer to x_t as the *state*, and denote the value domain of x_t with a set Ω . Without loss of generality, this paper focuses on dynamic processes, where the monitored states evolve over time. Stationary processes can be viewed as special cases with only one timestamp, i.e. $T = 1$, where the measurements collected during the entire period are all collapsed to $t = 1$.

Sensor systems. We consider the general case where M coexisting sensor systems sn_1, \dots, sn_M are monitoring the underlying states x_t and providing sensor measurements. Depending on different sensor systems, a sensor measurement could be either a single estimated value, or a value paired with certain error bounds. This paper assumes that the measurements are *probabilistic*: at a given time t sensor system sn_m reports a probability distribution Z_t^m defined on Ω , representing its estimate of the monitored state x_t . Deterministic measurements can be viewed as special cases where Z_t^m is reduced to a point distribution. For instance, if Ω is a finite discrete set containing N elements (e.g. all rooms in a building), Z_t^m is then a vector of probabilities $[Z_t^m(1), \dots, Z_t^m(N)]$, where $Z_t^m(j)$ is the belief of the system sn_m that the state x_t is the j th element of Ω .

Priors. We also assume that in certain timestamps, there may be certain prior knowledge on the state x_t . For instance, consider an indoor positioning scenario where the states are the locations of the user. Planned events, e.g. calendar entries, or social interactions like store check-ins, may directly reveal the actual location of the user at a given timestamp [6]. We refer to such information as the *priors*, and use a probability distribution Ψ_t defined on Ω to represent the prior knowledge on the state x_t .

Estimated state. In practice, the real state x_t can not be known or measured exactly, but needs to be estimated. In this paper we use a probability distribution \hat{X}_t defined on Ω to approximate the real state x_t . The distribution \hat{X}_t is evaluated with the observed probabilistic sensor measurements and available priors, which represents our best knowledge of the state x_t . Techniques that can be used to compute \hat{X}_t will be explained in Section 4.

Sensing applications. A sensing application is the agent that requires information on the monitored state x_t , and selects the sensor systems to task based on its accuracy requirements. We assume that a sensor application describes its accuracy requirements by providing a pair

$\langle f_\epsilon, A \rangle$, where f_ϵ is an *accuracy metric*, and A is a set of attributes that specifies the *accuracy index*.

Accuracy metric. An accuracy metric f_ϵ is a function which computes the accuracy of a probabilistic sensor measurement Z_t^m . We define the real accuracy of Z_t^m as a function with respect to the monitored state x_t , denoted as $f_\epsilon(Z_t^m; x_t)$. However, since x_t is typically unknown, the real accuracy $f_\epsilon(Z_t^m; x_t)$ can not be evaluated directly. Therefore in this paper, for a given Z_t^m , we consider its estimated accuracy $f_\epsilon(Z_t^m; \hat{X}_t)$, which is evaluated by feeding the estimated state \hat{X}_t into the accuracy metric f_ϵ .

Accuracy index. An accuracy index is an array containing the aggregated accuracy of a sensor system, defined by a set of attributes A , e.g. time or location. Assignments of attributes A specify the *context* in which the measurements are made, and can be used to aggregate them into groups, e.g. the measurements reported during a given time period or at a particular location. Thus elements of an accuracy index are the average accuracy of measurements grouped by the assignments of attributes A , which describes the expected accuracy of a sensor system in different context.

2.2 The Accuracy Estimation Problem

As discussed above, the sensing application needs to sense the physical phenomenon x_t , while M coexisting sensor systems are offering sensing services by providing probabilistic sensor measurements Z_t^m , $1 \leq m \leq M$, $1 \leq t \leq T$. The *accuracy estimation problem* studied in this paper is to assess the accuracy of the M sensor systems according to the accuracy requirements $\langle f_\epsilon, A \rangle$ from the sensing application, given all the observed sequences of sensor measurements $Z_{1:T}^1, \dots, Z_{1:T}^M$ and the prior knowledge on the monitored states $\Psi_{1:T}$. This will empower the sensing application to proactively select and task the suitable systems that satisfy its accuracy requirements. Of course in practice the sensing application may have other quality requirements, e.g. cost, energy consumption, or privacy. In this paper we assume that such information is already available from other sources (e.g. energy monitoring tools, web sources, etc.), and focus on estimating the accuracy. We also assume that the sensing application only tasks one sensor system at a time, which achieves the best trade-off between accuracy and the other quality requirements.

2.3 The Accuracy Estimation Framework

To address the accuracy estimation problem, we propose an *accuracy estimation framework*, which lies between the sensing application and the underlying sensor systems, and estimates the accuracy of the sensor systems given the requirements of the application. The proposed framework provides a general solution for accuracy estimation, which can be implemented with different techniques. With the available domain knowledge, it is also applicable to a broad spectrum of sensing scenarios. Intuitively, the accuracy of a sensor system should depend on the accuracy of its measurements. Although the actual state x_t is unknown, as discussed above, it is still possible to estimate the accuracy of a given measurement Z_t^m with the estimated state distribution \hat{X}_t , under the accuracy metric f_ϵ provided by the sensing application. Following this intuition, the proposed accuracy

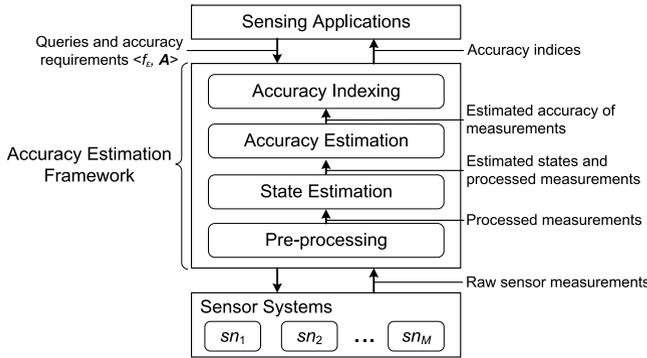


Fig. 2. Measurements reported by sensor systems are passed through the proposed accuracy estimation framework with four layers: pre-processing, state estimation, accuracy estimation, and accuracy indexing.

estimation framework has four layers: pre-processing, state estimation, accuracy estimation, and accuracy indexing.

Fig. 2 illustrates the layers of the proposed framework. The sensing application on top requires information on certain physical signals, e.g. the temperature of a building or the locations of a user, and would like to select one sensor system to task. It poses corresponding queries to the proposed framework, and specifies an accuracy metric f_e and a set of attributes A , which will be used to evaluate and index the accuracy of the reported measurements. The framework first activates all the coexisting sensor systems that can provide measurements on the queried states for a short period of time, and the raw sensor measurements pulled from the systems are first processed through the *pre-processing* layer. Then the processed measurements are forwarded to the *state estimation* layer, where the states are estimated based on sensor observations and available prior knowledge. The next *accuracy estimation* layer takes the estimated states together with the pre-processed sensor measurements as input, and evaluates the accuracy of measurements with the accuracy metric f_e . The estimated measurement accuracy is then forwarded to the *accuracy indexing* layer, where an accuracy index is built for each sensor system, according to the attributes A . Finally, the accuracy indices of the sensor systems are reported as the output of the accuracy estimation framework, which empowers the sensing application to choose the desired system to task depending on the context. The framework will initiate this process periodically to keep the accuracy indices updated. Now we are in a position to explain the layers of the proposed framework in more detail.

3 PRE-PROCESSING

The pre-processing layer collects and processes the raw measurements from coexisting sensor systems. It is optional, but usually necessary in practice. Raw measurements from different sensor systems are typically heterogeneous, for instance they may not conform to a global clock, or they may be generated at different time or space granularities. Moreover, some sensor systems may only be able to provide deterministic measurements without any error bounds. To address this, the pre-processing layer processes the raw sensor measurements through two steps: a) the *synchronization* step, and b) the *resample* step.

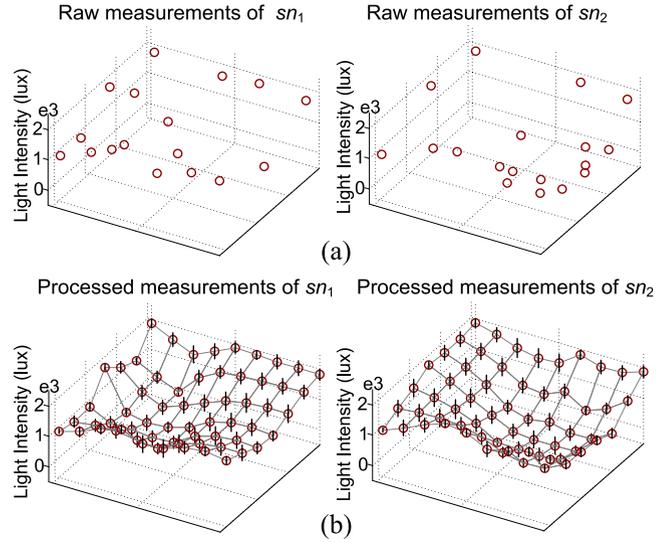


Fig. 3. (a) A snapshot of the raw sensor measurements on light intensity generated from two sensor systems sn_1 and sn_2 . (b) A snapshot of the pre-processed light intensity measurements by Gaussian process non-linear regression.

In the synchronization step, the measurements from different sensor systems are firstly re-timestamped according to the global clock provided by the proposed framework. If the measurements are generated in different metric systems, e.g. some temperature readings may be reported in Celsius scale while others could be in Fahrenheit scale, they are converted into the same metric system. For measurements without error bounds reported, this step also generates the corresponding error estimations and thus parses the deterministic measurements into probabilistic ones. This can be achieved by various existing techniques, such as model-driven approaches [7], [8], or approaches that leverage prior knowledge on sensor noise characteristics [9].

After the synchronization step, the resample step further subsamples or interpolates the measurements so that measurements from different sensor systems have the same time and space granularity. There is also a solid body of techniques that can be used in this context, and in our experiments, we use Gaussian process non-linear regression [10] to interpolate the sensor data over time and space. Fig. 3a shows the raw sensor measurements on light intensity, generated by two coexisting sensor systems used in our experiments. Note that each system only generates deterministic measurements at the locations where they have sensors. Fig. 3b shows the pre-processed sensor measurements, which have the same space and time granularity, and are annotated with confidence intervals.

4 STATE ESTIMATION

Given the pre-processed sensor measurements and the available priors as input, the state estimation layer computes the estimated state distribution \hat{X}_t . We assume that after pre-processing, at any given timestamp t , each sensor system sn_m , $1 \leq m \leq M$, provides a probabilistic measurement Z_t^m of the state x_t , where Z_t^m is a probability distribution defined on the sample space Ω . We also assume that certain priors Ψ_t may be available at some of the

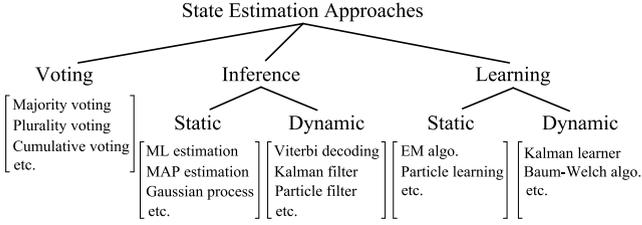


Fig. 4. The taxonomy of state estimation approaches, where for each class of approaches, several representative techniques are listed.

timestamps, as defined in Section 2.1. This layer can be implemented in various ways, and we design a taxonomy consisting of three main classes of approaches: voting, state inference and learning. The inference- and learning-based approaches can be further divided into two subclasses: static and dynamic, depending on whether the dynamics of the monitored process are taken into consideration. Fig. 4 illustrates the taxonomy of those approaches, along with specific examples of techniques under each class. In the following Sections 4.1, 4.2 and 4.3, we provide the high level overview of the approaches, and discuss the algorithm implementations of those approaches in the context of an indoor positioning scenario, where the positions of a user are monitored by multiple coexisting positioning systems.

Concretely, we assume that the indoor environment can be represented as a finite set L of N discrete locations l_1, \dots, l_N , e.g. different rooms or corridor segments (note that the described approaches can also be applied in the continuous case, but the detailed derivations are omitted due to space limitations). Therefore a probabilistic measurement Z_t^m can be represented as a vector of probabilities $[Z_t^m(1), \dots, Z_t^m(N)]$, where $Z_t^m(j)$ is the belief of the sensor system that the user is in location l_j . We use the same representation for the estimated state distribution \hat{X}_t and the priors Ψ_t . We also assume that Ψ_t exists at each timestamp: if prior information on state x_t is not provided, Ψ_t is reduced to uniform distribution. Finally, we always assume that the initial location of the user (i.e. state x_1) can be known exactly, e.g. from the card swipe at the main entrance.

4.1 Voting-Based Approach

Voting is a widely used approach, which aggregates the preferences from multiple information sources to achieve a collective decision. In the context of state estimation, the

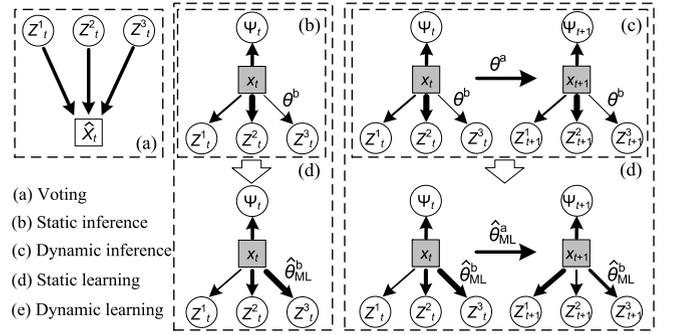


Fig. 5. Comparison of different state estimation approaches. (a) Voting merges sensor measurements with equal weight to evaluate \hat{X}_t . (b) Static inference infers state x_t with the measurements at t and known model parameters θ^b , which determine the likelihood of different measurements given the state (weighted arrows). (c) Dynamic inference estimates x_t with the full sequence of measurements and known model parameters, which include both θ^b and in addition θ^a which control the transitions between states. (d) Static learning re-estimates the model parameters with the measurements at t , and uses the learned $\hat{\theta}_{ML}^b$ (weighted arrows at the bottom, different from those in (b)) to infer x_t . (e) Dynamic learning learns the model parameters with the full sequence of measurements, and infers x_t with the learned $\hat{\theta}_{ML}^b$ and $\hat{\theta}_{ML}^a$.

voting approach works on one timestamp at a time. It treats the sensor systems as individual *voters*, and a probabilistic measurement Z_t^m is the *voting plan* of the system sn_m , which distributes a fixed amount of scores (the probability mass) across the sample space Ω according to the measured distribution. The estimated state distribution \hat{X}_t is then evaluated by combining the measured probability distributions Z_t^m from the M coexisting sensor systems (using equal weights) according to certain voting rules. Fig. 5a shows the basic idea of the voting-based approach.

Algorithm implementation. In the indoor positioning scenario, at a given timestamp t the reported measurements from the sensor systems are M probability vectors, $[Z_t^1(1), \dots, Z_t^1(N)], \dots, [Z_t^M(1), \dots, Z_t^M(N)]$. In this paper, we consider a simple yet effective voting-based algorithm (referred to as *VA* hereafter) that implements cumulative voting [11], where

$$\hat{X}_t(j) = c \sum_{m=1}^M Z_t^m(j), \quad (1)$$

for every location l_j (c is a normalizing constant). Fig. 6g shows the result of this voting-based algorithm (*VA*), which

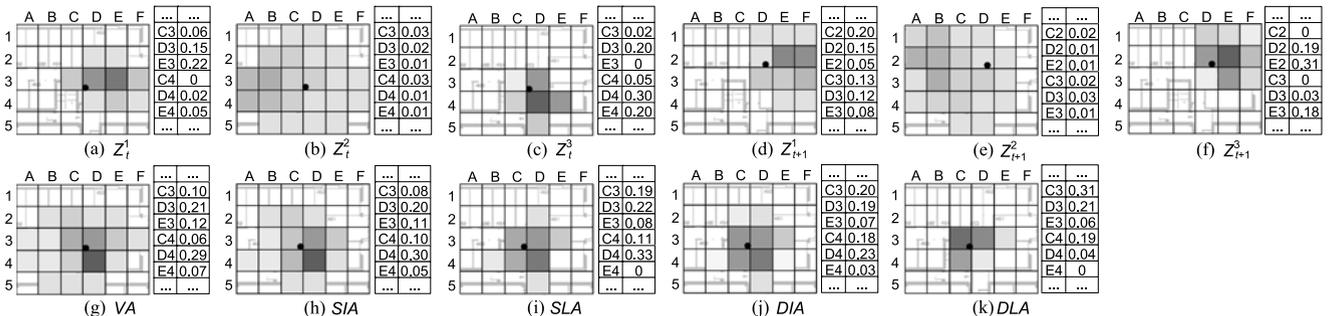


Fig. 6. Estimated states produced by different algorithms in the indoor positioning scenario, where the black dots in each figure show the ground truth. The blocks with different densities indicate the likelihood that the user is believed to be in those blocks (darker blocks mean higher probabilities). (a)–(f) The probabilistic measurements observed at two timestamps t and $t + 1$. (g)–(k) The estimated state \hat{X}_t computed by the voting (*VA*), static inference / learning (*SIA* / *SLA*) and dynamic inference / learning (*DIA* / *DLA*) algorithms respectively.

fuses the three probabilistic sensor measurements observed at time t , as shown in Figs. 6a, 6b and 6c.

4.2 Inference-Based Approach

The inference-based approach models the monitored states, the measurements from one or more sensor systems, and the available prior knowledge in a probabilistic model with a set of parameters θ known a priori. Generally θ depends on the type and structure of the model, and in our context we assume θ contains two parts: $\theta = \{\theta^a, \theta^b\}$, where θ^a determines the probability of being in current state x_t given the previous state x_{t-1} ; and θ^b controls the probabilities of observing probabilistic sensor measurements $Z_t^{1:M}$ given the current state x_t . The inference-based approach estimates the latent states based on observed measurements and priors, given the known θ . Depending on whether we exploit the temporal correlations between states, the approach can be further divided into *static* and *dynamic* inference.

4.2.1 Static Inference

Static inference ignores any temporal correlations between the latent states, and only accesses the measurements and priors at single timestamps. Therefore static inference does not consider the parameters θ^a , and evaluates \hat{X}_t as the posterior state distributing given the observed $Z_t^{1:M}$ and Ψ_t with the known θ^b only. If we assume the measurements from different sensor systems are conditionally independent and ignore any external sources of noise, the estimated state distribution \hat{X}_t can be computed as:

$$\hat{X}_t \propto p(x_t | \Psi_t) \prod_{m=1}^M p(Z_t^m | x_t, \theta^b), \quad (2)$$

where $p(Z_t^m | x_t, \theta^b)$ is the conditional probability distribution of observing the measurement Z_t^m given state x_t and known parameters θ^b . $p(x_t | \Psi_t)$ represents the state distribution given that the prior Ψ_t is observed. Note that Ψ_t is treated as a special observation here, which is not controlled by any model parameters and $p(x_t | \Psi_t)$ is assumed to be the prior distribution Ψ_t (we assume no additional knowledge on x_t is available). Fig. 5b shows an example of static inference, where state x_t is influenced by prior Ψ_t , and the likelihood of observed measurements are determined by parameters θ^b .

Algorithm implementation. We now explain the algorithm implementation of static inference in the indoor positioning scenario. We assume the identical setting as explained in voting-based algorithm. In this context, the model parameter θ^b can be defined concretely. We define $\theta^b = \{b_j^m(k)\}$, which represent the probabilities of having a location l_k in the measurement given the fact that the user is in location l_j for system sn_m , and are referred to as the *emission probability* hereafter. As shown in Eq. (2), $p(x_t | \Psi_t)$ is determined solely by the prior distribution Ψ_t , and thus the main task of this static inference-based algorithm (SIA) is to evaluate the likelihood of observing the measurements $Z_t^{1:M}$ given the state x_t . If the measurements are deterministic (e.g. single locations as in a standard HMM) and assuming conditional independence of measurements given the state, the likelihood $p(Z_t^{1:M} | x_t)$ is directly given by the products of the M

known emission probabilities $b_j^m(k)$ by definition. In the presence of probabilistic measurements, this likelihood can be evaluated by summing over all l_k :

$$\begin{aligned} \hat{X}_t(j) &\propto p(x_t = l_j | \Psi_t) p(Z_t^{1:M} | x_t = l_j) = \Psi_t(j) r_t(j); \\ r_t(j) &= \prod_{m=1}^M \sum_{k=1}^N Z_t^m(k) b_j^m(k), \end{aligned} \quad (3)$$

where variable $r_t(j)$ represents the likelihood of observing the probabilistic measurements $Z_t^{1:M}$ given that the state is l_j . In our experiments, the parameters $b_j^m(k)$ are initialized as follows: for each measurement Z_t^m , we find the location \tilde{l} that has the largest reported probability. Then $b_j^m(k)$ is computed as the average of $Z_t^m(k)$ for all measurements from sn_m , whose \tilde{l} is l_j . In other words, we assume that the states are \tilde{l} and use them to compute the parameters $b_j^m(k)$ for sn_m empirically. If the observed measurements are not sufficient to estimate all the emission probabilities, those uninitialized $b_j^m(k)$ are set to default values. Fig. 6h shows the estimated state distribution computed by this algorithm (SIA), which is similar to that of voting (VA).

4.2.2 Dynamic Inference

Dynamic inference assumes the monitored states are temporally correlated, and estimates the states with all the observed measurements and priors. Let the states be a time-varying sequence: $x_{1:T}$, with M measurement sequences $Z_{1:T}^1, \dots, Z_{1:T}^M$ from the sensor systems. In practice, the dynamics are usually assumed to be *Markovian* for simplicity, i.e. $p(x_t | x_{1:t-1}) = p(x_t | x_{t-1})$, which is determined by the known model parameters θ^a . In addition, measurements generated at different timestamps are assumed to be independent conditioned on the states. Under those assumptions, the estimated state distribution \hat{X}_t , i.e. the posterior state distribution given all the observed sensor measurements and priors, can be represented as:

$$\hat{X}_t \propto \underbrace{p(Z_{1:t}^{1:M}, \Psi_{1:t}, x_t | \theta)}_{A_t} \underbrace{p(Z_{t+1:T}^{1:M} | \Psi_{t+1:T}, x_t, \theta)}_{B_t}, \quad (4)$$

where A_t is the joint distribution of the observed measurements $Z_{1:t}^{1:M}$, priors $\Psi_{1:t}$ and the state x_t until time t , given the known parameters θ ; while B_t is conditional distribution of the measurements made from time $t+1$, given the current state x_t , future priors $\Psi_{t+1:T}$ and parameters θ . Both the term A_t and B_t can be evaluated iteratively:

$$\begin{aligned} A_t &\propto p(Z_t^{1:M} | x_t, \theta^b) \int_{x_{t-1}} p(x_t | x_{t-1}, \Psi_t, \theta^a) A_{t-1} dx_{t-1}; \\ B_t &\propto \int_{x_{t+1}} p(Z_{t+1}^{1:M} | x_{t+1}, \theta^b) p(x_{t+1} | x_t, \Psi_{t+1}, \theta^a) B_{t+1} dx_{t+1}, \end{aligned} \quad (5)$$

where $p(Z_t^{1:M} | x_t, \theta^b)$ and $p(Z_{t+1}^{1:M} | x_{t+1}, \theta^b)$ can be factored as in the static case. In the dynamic case, the priors are incorporated in the probability distribution $p(x_t | x_{t-1}, \Psi_t, \theta^a)$ that links two consecutive states, so that the state transition is governed by both the model parameters θ^a and prior Ψ_t . Fig. 5c shows a simple case with two timestamps, where the state x_{t+1} is influenced by x_t (through θ^a) and Ψ_{t+1} , while measurements are emitted under the control of θ^b .

Algorithm implementation. In the indoor positioning scenario, under the assumption of discrete state space, we define the model parameter $\theta^a = \{a_{ij}\}$, where $a_{ij} = p(x_{t+1} = l_j | x_t = l_i)$, i.e. the probability that the user moves from one location l_i to another l_j at two consecutive timestamps (under the Markovian assumption). We refer to a_{ij} as the *transition probabilities*, which are assumed to be known, and in our experiments a_{ij} is determined by physical constraints: e.g. a user has equal probabilities of moving to any adjacent locations. The parameter $\theta^b = \{b_j^m(k)\}$ is defined and initialized in the same way as the static algorithm *SIA*. Following Eqs. (4) and (5), this dynamic inference-based algorithm (*DIA*) evaluates the estimated state distribution \hat{X}_t by extending the forward-backward algorithm in hidden Markov models (HMMs):

$$\hat{X}_t(j) = p(x_t = l_j | Z_{1:T}^{1:M}, \Psi_{1:T}) \propto \alpha_t(j)\beta_t(j); \quad (6)$$

where $\alpha_t(j)$ and $\beta_t(j)$ are the extended *forward* and *backward* variables:

$$\begin{aligned} \alpha_t(j) &\propto \left[\sum_{i=1}^N \alpha_{t-1}(i)a_{ij} \right] \Psi_t(j)r_t(j); \\ \beta_t(j) &\propto \sum_{i=1}^N [a_{ji}\Psi_{t+1}(i)r_{t+1}(i)\beta_{t+1}(i)], \end{aligned} \quad (7)$$

where $r_t(j)$ is the same as defined in Eqn. (3). Note that the derivation of those variables are very different from the standard HMMs, since here we have multiple sequences of probabilistic measurements. Also the priors Ψ_t are incorporated at each timestamp to bias the estimated state. Fig. 6j shows the distribution \hat{X}_t estimated by this dynamic inference-based algorithm (*DIA*), where comparing with that in static algorithm (*SIA*), more probability mass is concentrated in locations near the ground truth.

4.3 Learning-Based Approach

Unlike the inference-based approach, the learning-based approach does not assume any prior knowledge on the model parameters θ . It starts with an estimate of the parameters, and iteratively refines this estimate to be more consistent with the sensor measurements and priors. The estimated state distribution \hat{X}_t is then inferred with the learned model parameters and the observed data.

4.3.1 Static Learning

Static learning first tries to find the model parameters $\theta = \{\theta^b\}$ (θ^a are not included in this case since the correlation between states is ignored) that are the most consistent with the data observed within each timestamp, given by the maximum likelihood (ML) estimate $\hat{\theta}_{ML}$, as shown in Fig. 5d. As in general latent variable models, $\hat{\theta}_{ML}$ can be computed by the EM approach [12], which works iteratively with the following two steps until convergence:

- a) The *E*-step, which computes the expected log likelihood function $Q(\theta', \theta)$ of the new parameters θ' . The expectation is taken with respect to the conditional distribution of the states given the observed data, under the current parameters θ :

$$\begin{aligned} Q(\theta', \theta) &= E_{x_t | Z_t^{1:M}, \Psi_t, \theta} [\log p(Z_t^{1:M}, \Psi_t, x_t | \theta')] \\ &= \int_{x_t} p(x_t | Z_t^{1:M}, \Psi_t, \theta) \log p(Z_t^{1:M}, \Psi_t, x_t | \theta') dx_t, \end{aligned} \quad (8)$$

- b) The *M*-step, which finds the new parameters θ' that maximize the Q function: $\theta' = \arg \max_{\theta'} Q(\theta', \theta)$.

After convergence, this approach evaluates the distribution of the estimated state \hat{X}_t with the learned model parameters $\hat{\theta}_{ML}$ as in static inference.

Algorithm implementation. For the indoor positioning scenario, in this static case the model parameters that need to be learned are $\theta^b = \{b_j^m(k)\}$, i.e. the emission probabilities. Following the EM scheme discussed above, this static learning-based algorithm (*SLA*) finds the new parameters $b_j^m(k)'$ that maximize the Q function, which are given by:

$$\begin{aligned} b_j^m(k)' &= \hat{X}_t(j)^{-1} s_t^m(j, k); \\ s_t^m(j, k) &= \Psi_t(j) Z_t^m(k) b_j^m(k) \prod_{\substack{\tilde{m}=1 \\ \tilde{m} \neq m}}^M \sum_{i=1}^N Z_t^{\tilde{m}}(k) b_j^{\tilde{m}}(i), \end{aligned} \quad (9)$$

where $\hat{X}_t(j)$ is the estimated probability of being in location l_j , and $s_t^m(j, k)$ represents the probability that system sm_m has location l_k in its measurement Z_t^m and observing all measurements $Z^{\tilde{m}t}$ ($1 \leq \tilde{m} \leq M$, $\tilde{m} \neq m$) from other systems when the actual state is l_j . Both of $\hat{X}_t(j)$ and $s_t^m(j, k)$ are evaluated with the current parameters. Note that inference is in fact a subroutine of learning, and therefore the estimated state distribution can be computed during the last learning iteration. Fig. 6i shows the \hat{X}_t estimated by this static learning-based algorithm (*SLA*).

4.3.2 Dynamic Learning

Similar to dynamic inference, dynamic learning also assumes the hidden state varies over time, and the state transitions are governed by parameters θ^a . Instead of relying on model parameters known in advance, it firstly use all measurements and priors to learn the parameters $\theta = \{\theta^a, \theta^b\}$, as shown in Fig. 5e. It also uses the EM scheme, but the derivation of the Q function is different from the static case. We make identical assumptions as in dynamic inference, and the likelihood function $Q(\theta', \theta)$ becomes:

$$\begin{aligned} Q(\theta', \theta) &= E_{x_{1:T} | Z_{1:T}^{1:M}, \Psi_{1:T}, \theta} [\log p(Z_{1:T}^{1:M}, \Psi_{1:T}, x_{1:T} | \theta')] \\ &= \int_{x_{1:T}} p(x_{1:T} | Z_{1:T}^{1:M}, \Psi_{1:T}, \theta) \log p(Z_{1:T}^{1:M}, \Psi_{1:T}, x_{1:T} | \theta') dx_{1:T}, \end{aligned} \quad (10)$$

which integrates over state sequences $x_{1:T}$. The maximization step is the same as in static learning. In both static and dynamic cases, the priors $\Psi_{1:T}$ are incorporated in the Q function (as in Eqs. (8) and (10)) at each learning iteration, and thus bias the learned parameters and estimated states.

Algorithm implementation. We now explain the implementation of dynamic learning in the context of the indoor positioning scenario. In this case, both the emission probabilities $b_j^m(k)$ and the transition probabilities a_{ij} need to be re-estimated. In addition, the derivation of the Q function in *E*-steps different from the static case, which takes multiple

sequences of probabilistic measurements into account. In the M -step, this dynamic learning-based algorithm (DLA) finds the new parameters a'_{ij} and $b'_j(k)'$ that maximize the derived Q function:

$$\begin{aligned} a'_{ij} &= \left[\sum_{t=2}^T \alpha_{t-1}(i) a_{ij} \Psi_t(j) r_t(j) \beta_t(j) \right] u_{1:T-1}(i); \\ b'_j(k)' &= [s_1^m(j, k) + v_{2:T}^m(j, k)] u_{1:T}(j); \\ u_{1:T}(j) &= \left[\sum_{t=1}^T \hat{X}_t(j) \right]^{-1}; \\ v_{2:T}^m(j, k) &= \sum_{t=2}^T \left[\sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right] s_t^m(j, k) \beta_t(j), \end{aligned} \quad (11)$$

where $\alpha_t(j)$ and $\beta_t(j)$ are the extended forward and backward variables evaluated with the current model parameters, and can be computed as shown in Eq. (7). Fig. 6k shows the estimated state distribution \hat{X}_t computed by dynamic learning (DLA), where the majority of the probability mass is correctly concentrated on the actual position of the user.

5 ACCURACY ESTIMATION

The state estimation layer discussed above approximates the latent states with all observed sensor observations and available priors. The proposed framework then forwards the estimated state distributions together with the pre-processed sensor measurements to the accuracy estimation layer, which estimates the accuracy of the measurements with the metric specified by the sensing application. Given a probabilistic measurement Z_t^m and the computed state distribution \hat{X}_t , the estimated accuracy of Z_t^m is given by $f_\epsilon(Z_t^m; \hat{X}_t)$, which is assumed to be a scalar value. We consider two accuracy metrics, a *proximity-based* and a *similarity-based*, which can be applied in sensing applications with different types of accuracy requirements.

The proximity-based accuracy metric. The proximity-based accuracy metric defines the accuracy of a probabilistic measurement Z_t^m based on its *distance* to the estimated state \hat{X}_t . Recall that both Z_t^m and \hat{X}_t are in fact probability distributions defined on Ω . Let $p_{Z_t^m}(\zeta)$ and $p_{\hat{X}_t}(\chi)$ be the probability density functions (probability mass function in discrete cases) of Z_t^m and \hat{X}_t respectively, $\zeta, \chi \in \Omega$. The proximity-based accuracy of Z_t^m is then defined as:

$$f_\epsilon^p(Z_t^m; \hat{X}_t) = \int_{\zeta, \chi} p_{Z_t^m}(\zeta) p_{\hat{X}_t}(\chi) C(\zeta - \chi) d\zeta d\chi, \quad (12)$$

where $C(\cdot)$ is a distance function that can take various forms, and in this paper we consider the Euclidean distance $C(\zeta - \chi) = \|\zeta - \chi\|$. Therefore $f_\epsilon^p(Z_t^m; \hat{X}_t)$ can be viewed as the “expected” Euclidean distance between the measurement Z_t^m and estimated state \hat{X}_t . It tends to favour the measurements whose probability mass is geometrically close to the estimated state distribution, and thus is typically used in sensing scenarios that prefer such proximity.

The similarity-based accuracy metric. On the other hand, the similarity-based metric defines accuracy as the *divergence*

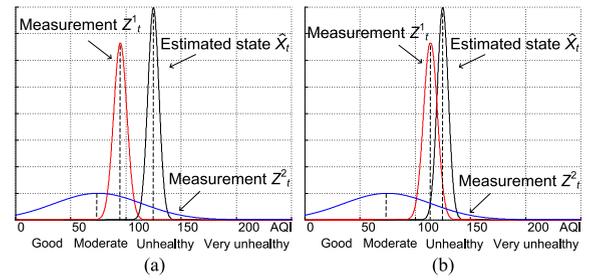


Fig. 7. (a) Under proximity-based accuracy metrics f_ϵ^p the measurement Z_t^1 is more accurate, while under similarity-based accuracy metrics f_ϵ^s the measurement Z_t^2 is more accurate. (b) Under both metrics the measurement Z_t^1 is more accurate.

between the probabilistic measurements and the estimated state distributions. There are many statistical divergence metrics that can be used, and in this paper we consider the widely adopted Kullback-Leibler (KL) divergence. For a measurement Z_t^m , its estimated similarity-based accuracy is defined as the KL divergence between the distributions of the measurement and the estimated state:

$$f_\epsilon^s(Z_t^m; \hat{X}_t) = D_{\text{KL}}(\hat{X}_t \| Z_t^m), \quad (13)$$

which indicates how well the measurement Z_t^m can be used to approximate the estimated state distribution \hat{X}_t . This accuracy metric is particularly useful when the sensing application would like to discourage measurements with unreasonably high confidence.

Of course many other functions can be used as accuracy metrics, and there is no universal accuracy metric that is suitable for all sensing scenarios. In practice, different sensing applications are likely to prefer different accuracy metrics, even if they are monitoring the same physical signal. For example, consider the air pollution monitoring scenario shown in Fig. 7a, where the sensor systems are monitoring the air pollution levels. If a sensing application would like to rank the measurements based on how close they are from the state, e.g. when calibrating the sensors, the proximity-based accuracy metric should be used: in Fig. 7a, Z_t^1 is more accurate than Z_t^2 under the proximity-based accuracy metric. On the other hand, if a sensing application would like to classify the pollution level based on the reported measurements, the proximity-based accuracy metric may lead to wrong conclusions. For instance in Fig. 7a, the proximity-based accuracy metrics would prefer measurement Z_t^1 since it is geometrically closer to the estimated state. However, it is undesirable for this application because Z_t^1 puts the majority of its belief in the “moderate” (AQI 50-100) category, whereas the state is actually “unhealthy” (AQI 100-150). This may cause serious false positives in practice, and the similarity-based accuracy metric can avoid this by favoring the measurement Z_t^2 that reports a relative “flat” distribution without concentrating its belief wrongly.

To generalize, the proximity-based accuracy metric cares more about the *distance* between the measurements and the states, while the similarity-based accuracy metric is more sensitive to the *differences* in the distributions. However, the two metrics are not necessarily always opposite. Fig. 7b shows a similar scenario as in Fig. 7a, but in this case the

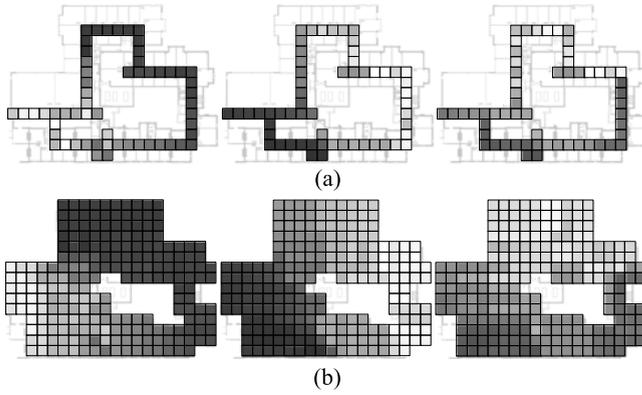


Fig. 8. The accuracy indices over locations built for three sensor systems in the indoor positioning scenario, under the proximity-based accuracy metric f_p^i . The lighter blocks indicate smaller averaged f_p^i values, i.e. the systems are more accurate at those locations. (a) After the accuracy aggregation step, the accuracy indices of the sensor systems only contain the averaged accuracy at the parts that have been visited by the user. (b) The accuracy indices after the accuracy interpolation step.

measurement Z_t^i is considered to be more accurate under both proximity- and similarity-based accuracy metrics.

6 ACCURACY INDEXING

The above accuracy estimation layer evaluates the accuracy of individual sensor measurements, and this accuracy indexing layer further aggregates the estimated accuracy to reason about the accuracy of sensor systems. Intuitively, the accuracy of a sensor system should depend on the accuracy of all the measurements it generates. In practice, however, it is usually more meaningful to investigate the accuracy of a sensor system in a given *context*, e.g. accuracy during a certain period of time or at a particular location. In this paper, we assume the context information is specified by value assignments of a finite set of *attributes* A (e.g. time or location), which are provided by the sensing application. Without loss of generality, we also assume the value domain dom_A of the attributes A is *discrete* and *finite*.

Let $a \in dom_A$ be a value tuple defined in dom_A . For a given sensor system sn_m , we refer to the subset of measurements whose attributes A are assigned to a as the measurements *grouped by* a . For example, it could be all the temperature measurements observed for a particular *location*, or the position measurements of a given *user*. Then for a sensor system sn_m , its accuracy index is a multidimensional array, where each element is the average accuracy of the measurements grouped by a , $a \in dom_A$.

The proposed accuracy indexing layer builds the accuracy indices in two steps: a) the *accuracy aggregation* step, and b) the *accuracy interpolation* step. The first aggregation step is straightforward, where the estimated accuracy of the sensor measurements is averaged according to the attributes A : measurements are grouped by tuple a assigned to attributes A , and for each value assignment with non-empty group of measurements, the corresponding element in the accuracy index is assigned to the average accuracy of all measurements in this group. Fig. 8a shows an example of the indoor positioning scenario, where the sensing application would like to build the accuracy indices of three

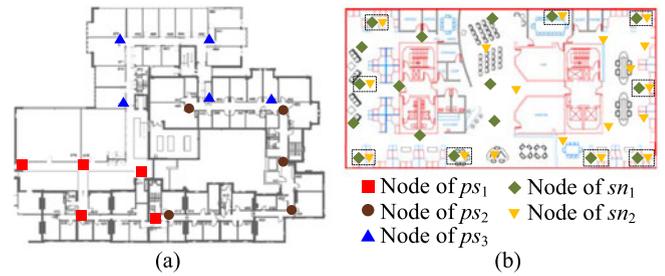


Fig. 9. (a) Experiment setup of the indoor positioning scenario (left) and the environmental monitoring scenario (right).

positioning systems over the discrete space. Each block in Fig. 8a represents the average accuracy of the measurements observed at that particular location, where lighter blocks indicate higher accuracy.

However in some cases, the average accuracy computed by the aggregation step may not be sufficient to build the *complete* accuracy indices. For instance in the example shown in Fig. 8a, since the accuracy estimation process only runs for a limited period of time, the computed accuracy information may not be able to cover the entire environment, but only the parts that have been visited by the user. The accuracy interpolation step addresses this by filling up the unknown accuracy based on computed accuracy information. This step can be implemented by various techniques, such as linear/spline interpolation [13], wavelets [14], or Gaussian processes [10]. Fig. 8b shows the complete accuracy indices after the interpolation step, where each location is annotated with estimated accuracy.

7 EVALUATION

7.1 Experiment Setup

We evaluate the proposed accuracy estimation framework on datasets collected from two real sensing scenarios.

Indoor positioning scenario. The data is collected from an indoor localization scenario on the 4th floor of a CS department building. Three different indoor positioning systems are deployed and running in parallel, reporting user location, as shown in Fig. 9a. Each of the positioning systems, ps_1 to ps_3 , owns a set of WiFi basestations placed in different positions of the floor. These basestations periodically broadcast WiFi beacons, which are received by the nearby mobile devices carried by the users. Each positioning system also receives data from a set of inertial measurement units (IMUs) attached to the feet of the users, and estimates position by combining the inertial data and the WiFi signal strengths from the basestations it owns. The ground truth is collected by the users: the map of the floor is displayed on their mobile devices and they tap the positions they are in to log their coordinates.

We tracked two research students for approximately 3~4 hours per day (limited by the battery life of the IMUs), and collected data for 20 days in total. We randomly selected five days of the data, retrieved the meaningful trajectories (the timestamps that the user is actually moving) by thresholding the accelerometer readings, and subsampled it at a rate of 0.5 Hz. We assume space is discrete, i.e. it is a finite set $L = \{l_k\}$ with N

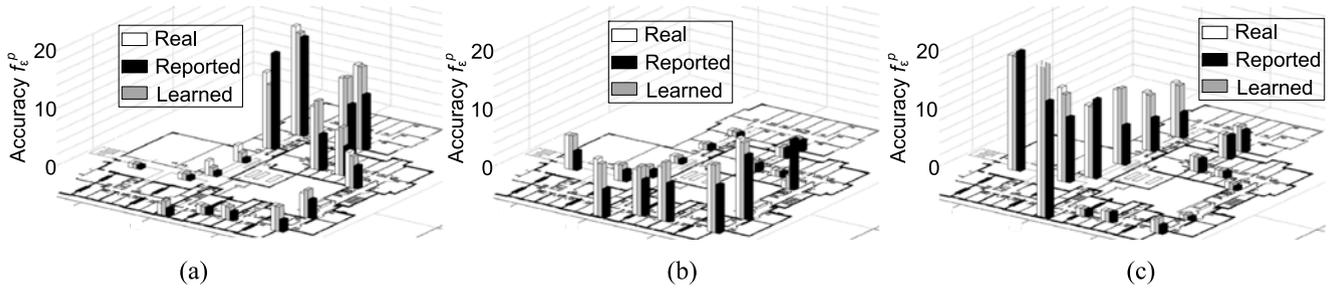


Fig. 10. The real, reported and learnt (computed by *DLA*) accuracy for positioning system ps_1 (left), ps_2 (middle) and ps_3 (right).

discretized locations, e.g. different rooms or corridor segments. In our experiment, the average size of a discrete location is $3 \text{ m} \times 3 \text{ m}$ and $N = 209$. The trajectories were then discretized according to L . For a given timestamp, the measurement from a positioning system is a probability vector of length N , where the i th probability represents the belief of the system that the user is at location l_i .

We also assume that in this scenario the proximity-based accuracy metric f_ϵ^p is preferred, i.e. the estimated accuracy of a probabilistic location measurement Z_t^m is determined by the “expected” Euclidean distance between Z_t^m and the estimated state distribution \hat{X}_t , as defined in Eq. (12).

Environmental monitoring scenario. The data was collected from the Intel Lab dataset [15]. The dataset contains temperature, humidity and light data collected from 54 sensors deployed in a lab environment for more than a month. We randomly selected the light intensity readings of approximately five consecutive days for our experiments. We divided the 51 sensors (three sensors are omitted since they failed midway) into two groups randomly, where data from 26 of them were used to generate light measurements (as explained below), and the rest were used as ground truth.

We created two co-located sensor systems, sn_1 and sn_2 , by selecting different subsets of the 26 training sensors, as shown in Fig. 9b, where sensors that are virtually “shared” by the two systems are grouped by rectangles. We then applied Gaussian process non-linear regression [10] to interpolate the data from each system across the space. Therefore, for a given timestamp t and a given point in space, the measurement from sensor system sn_1 or sn_2 is a Gaussian distribution $\mathcal{N}(\mu, \sigma)$, with an estimated light intensity value μ and variance σ .

Unlike the positioning scenario, here we assume that the similarity-based accuracy metric f_ϵ^s is used in this scenario, which evaluates the accuracy as the KL divergence between the measured distribution Z_t^m and the estimated state distribution \hat{X}_t : $f_\epsilon^s(Z_t^m; \hat{X}_t) = D_{\text{KL}}(\hat{X}_t \| Z_t^m)$.

7.2 Competing Algorithms

We compare different implementations of the proposed accuracy estimation framework in both experiment scenarios introduced above. For simplicity, we assume the pre-processing and accuracy indexing layers are always the same, and the accuracy estimation layer is implemented according to the different accuracy requirements of the scenarios (i.e. the proximity-based metric f_ϵ^p for the positioning scenario while similarity-based f_ϵ^s for the environmental monitoring scenario). We vary the algorithms used for state

estimation layer to create different implementations of the framework. We consider all algorithms in Section 4: the *VA*, the *SIA*, the *DIA*, the *SLA*, and the *DLA*. In addition, the following two algorithms are used as the baselines:

The oracle algorithm (OA) possesses the ground truth x_t , and for a given measurement Z_t^m , the accuracy it computes is the real accuracy, i.e. $f_{\epsilon_0}(Z_t^m) = f_\epsilon(Z_t^m; X_t)$.

The report-based algorithm (RA) uses the mean $\mu_{Z_t^m}$ of the probabilistic measurement Z_t^m as the estimation of the state (\hat{X}_t is reduced to a point distribution) to evaluate the accuracy, i.e. $f_{\epsilon_R}(Z_t^m) = f_\epsilon(Z_t^m; \mu_{Z_t^m})$.

Note that *SIA*, *SLA*, *DIA* and *DLA* can also incorporate prior knowledge on states, as discussed in Section 4. We evaluate the above competing algorithms against the metric of *Accuracy Estimation Error* EE^A . For a given measurement Z_t^m , EE^A is defined as the squared difference between the estimated accuracy $f_\epsilon(Z_t^m; \hat{X}_t)$ and the ground truth accuracy $f_{\epsilon_0}(Z_t^m)$: $EE^A = (f_\epsilon(Z_t^m; \hat{X}_t) - f_{\epsilon_0}(Z_t^m))^2$.

7.3 Experiment Results

The proposed accuracy estimation techniques are implemented in Matlab 8.0, and all experiments were performed on a quad-core machine with Linux 2.6.32.

Accuracy of sensor systems varies over time and space. The first set of experiments shows that the accuracy of a sensor system can vary over time and space, while the reported accuracy may not be a good indicator of the real accuracy. For the positioning scenario, Fig. 10 shows that the real accuracy (averaged over all timestamps) of the co-located positioning systems $ps_1 \sim ps_3$ (white bars) vary over space. The accuracy of a positioning system is higher in areas where it has denser sensing infrastructure. In this experiment we see that ps_1 has good accuracy (shorter white bars) at the left bottom part of the floor, while ps_2 performs well at the right side, and ps_3 dominates the top area. This experiment also shows that the reported accuracy is not always reliable: it (grey bars) consistently over or under estimates the real accuracy (white bars). The accuracy computed by *DLA* (black bars) is much closer to the real accuracy (white bars). This shows that in the absence of ground truth, the real accuracy can be effectively approximated by applying suitable techniques. For the environmental monitoring scenario, Fig. 11 shows that the real accuracy of a sensor system can vary over both location and time. Figs. 11a and 11b show snapshots of the light measurements reported at daytime by systems sn_1 and sn_2 respectively. We can see that: a) the differences between the real light values and reported

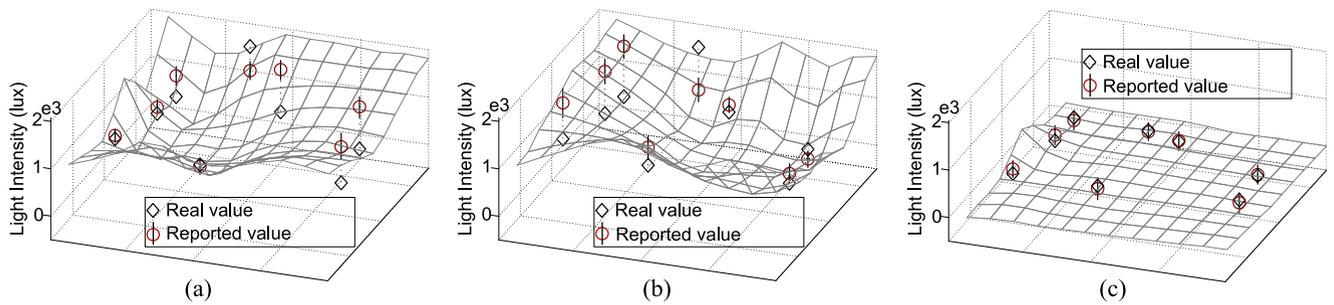


Fig. 11. 3D snapshots showing that the real accuracy varies over space and time. The surfaces show the light intensity measurements (only the means) across space at different timestamps. The first two graphs show real and reported light intensity data generated at daytime by sensor systems sn_1 (left) and sn_2 (middle). The right graph shows real and reported light intensity data generated by sn_1 at night.

ones vary across space, and b) the reported accuracy (variance) is very unreliable and the real light values consistently fall out of the 95 percent intervals of the reported ones. Fig. 11c shows a snapshot of real and reported values (by sn_1) at night: notice the differences are very small, which suggests that sn_1 becomes more accurate at night.

Accuracy estimation performance. The second set of experiments compares the performance of the competing algorithms (discussed in Section 7.2) in terms of the average accuracy estimation error (EE^A) over all measurements. The left graph of Fig. 12a shows the average EE^A of different algorithms in the positioning scenario when we consider the probabilistic measurements of all three co-located positioning systems. We can see that the gap between voting and report-based algorithms (VA and RA) is about 50 percent, which means that measurements from the co-located sensor systems can indeed help to improve the estimation of accuracy, and simple approaches like voting could be quite effective in practice. Techniques that only operate on single timestamps (SIA and SLA) can provide about 10 percent reduction of estimation errors compared to voting, which actually comes from the model parameters that determine the weights of measurements from different systems when combining them. When moving to dynamic approaches, however, we observe significant improvements in estimation error. Dynamic inference (DIA) features about 40 percent improvement compared to voting, because it takes all measurements into account and uses a state transition

model that reflects the underlying state dynamics. Dynamic learning (DLA) offers more than 50 percent benefit compared to voting, since DLA also learns new model parameters that best explain the observed measurements. Finally, the improvement from the naive approach (RA) to the best technique (DLA) is almost eight fold. For the environmental monitoring scenario, as shown in Fig. 12b, there is a similar trend of improvement as we move to more sophisticated techniques, but in this case the improvement from voting to DIA is marginal, because we use naive model parameters (derived directly from reported confidence intervals, which often do not cover the ground truth as shown in Fig. 11). The benefits of DLA , however, are far more pronounced, since the learnt model parameters are more accurate. Fig. 12c shows that the relative performance of different algorithms varies significantly over time.

Sensitivity to the number of information sources. The third set of experiments investigate how the number of co-located sensor systems affects the accuracy estimation. Fig. 12a shows the results of the positioning scenario. We can see that with fewer co-located systems, the performance gaps between the different techniques become smaller. In the case where only one system is available, the best performing algorithm (DLA) has similar EE^A to the naive approach of trusting the reported accuracy (RA).

Running cost versus performance gain. The fourth set of experiments studies the trade-off between accuracy estimation and computation cost. We measure the execution time

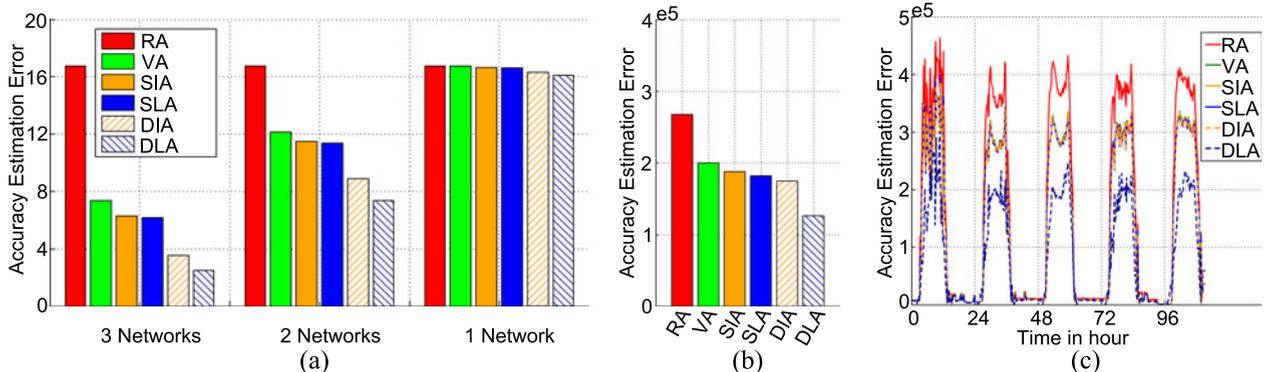


Fig. 12. (a) Average estimation errors of different algorithms in the positioning scenario when the number of co-located systems varies from 3 to 1. (b) Average estimation errors of different algorithms in the environmental monitoring scenario. (c) Estimation errors of different algorithms in the environmental monitoring scenario vary over time.

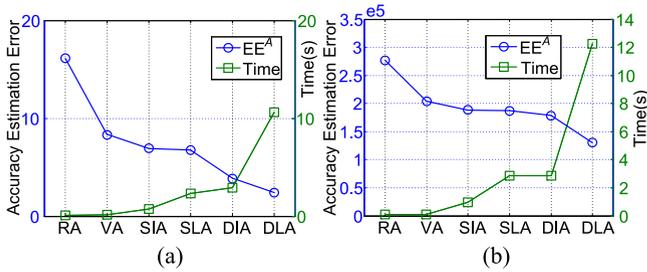


Fig. 13. Running time versus performance for different algorithms in the positioning scenario (left) and environmental monitoring scenario (right).

of different algorithms and compare it with the performance gain in terms of accuracy estimation error EE^A). Fig. 13 shows the trade-off on the two experiment scenarios respectively. In general learning-based techniques are more expensive than the inference-based, since learning requires iterative evaluation of the likelihood of data, which is essentially multiple runs of inference. In our experiments, on average learning is several times slower than inference but it can only improve the estimation performance by at most 30 percent (from *DIA* to *DLA* in the environmental monitoring scenario). The results are similar in the positioning scenario, where moving from dynamic inference to dynamic learning, the performance gain is about 25 percent at the expense of tripling the running time.

Sensitivity to prior knowledge. The last set of experiments shows how prior knowledge can influence the competing algorithms. For both scenarios, the priors are generated by first selecting a random subset of the timestamps. At these timestamps, the distribution of prior ρ_t is set to be the ground truth value plus a small quantity of noise. We vary the percentage of timestamps that have priors, and study the effect on performance of different techniques (Fig. 14). In the positioning scenario, as the amount of priors increases, the static algorithms improve linearly. For dynamic algorithms, the estimation error has a quick drop before the percentage of priors reaches 20 percent, and then becomes flat. This is because the dynamic techniques exploit temporal correlations in the data, which enables prior knowledge to impact nearby states. There is also a small gap between dynamic inference and learning, since learning can use priors to better assess model parameters. In the environmental monitoring scenario, a similar behavior can be witnessed, but the gap between dynamic inference and learning is larger since the model parameters used by inference is trivial, while learning can recalibrate them from both the observed measurements and priors.

8 RELATED WORK

State inference. A large body of research in sensor networks has involved statistical inference about the sensed environment. Examples are regression and prediction of environmental variables, such as temperature, light intensity, humidity and pollution, taking into account spatial and temporal correlations in the sensor readings, and incorporating measurement noise. A common approach is to use techniques such as Kriging [16] and Gaussian Processes (GPs) [10] to interpolate between sensor readings and infer the values of environmental variables in places where there

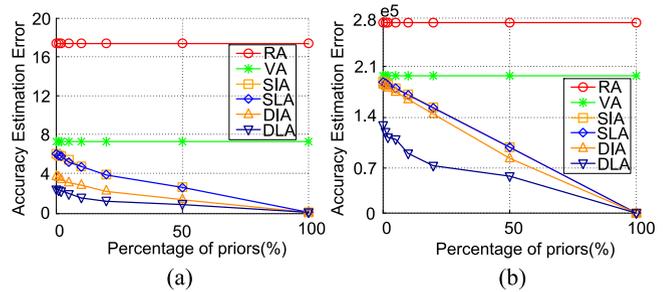


Fig. 14. Estimation error of different algorithms when the percentage of priors varies in the positioning scenario (left) and environmental monitoring scenario (right).

are no sensors, or when sensors have failed or simply did not generate readings at certain timestamps. For example, Osborne et al. have proposed a computationally efficient implementation of GPs for sensor network applications in the context of environmental sensing [17].

A lot of work has also investigated the dynamic version of the estimation problem. A well-studied example is that of node tracking, where the physical locations of moving objects are tracked by fixed and/or mobile sensors. HMMs are commonly used in the context of map matching, i.e. the problem of finding the most likely trajectory that accounts for measurement noise and known map constraints [18], [19], [20], [21], [22]. More specifically, VTrack uses mobile phones mounted in cars to estimate road travel times using a sequence of inaccurate position observations [20]. EasyTracker uses HMMs in the context of transit tracking, and uses the inferred tracks to detect transit stops and predict arrival times [22]. An HMM-based approach is also used in CTrack, where the goal is to associate a sequence of cellular fingerprints to a sequence of road segments on a known map [23]. 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 [24]. A recent comparative evaluation of different filters for person localization using RSS (Received Signal Strength) measurements is presented in [25].

Parameter estimation/learning. While much of the initial work was restricted to distributed inference of the monitored state, more recently there has been considerable interest in parameter estimation. For example, [3] estimates both the correctness of measurements and the reliability of participants in social sensing applications by solving an expectation maximization problem. The work in [4] considers a stream setting, where the quality of observations are recursively updated as new data arrives. Our work in [6] considers the indoor tracking problem, and assumes the coexistence of multiple positioning systems that compete for the same users. It proposes an expectation maximization algorithm to learn the emission probabilities of each positioning system in various parts of the indoor environment. Note that those approaches [3], [4], [6] first learn the parameters of the sensor models, and then use the learned parameters to estimate the accuracy of sensor measurements or reliability of human participants. In this paper, we show that parameter learning (or parameter estimation) is not the only way to estimate the accuracy of sensor measurements. A simpler alternative approach, which has been largely

neglected, is to use inference algorithms that estimate the state of the monitored phenomenon, and then measure the distance of stochastic measurements from the inferred state. In this paper we show that both inference and learning algorithms can be used to tackle the accuracy estimation problem, and their relative performance largely depends on the application scenario, and our prior knowledge of it.

Quality estimation. Our work is also related to quality estimation approaches, e.g. fact finding techniques in information networks [26], [27], [28]. 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 [26], 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. A Bayesian interpretation of fact finding is offered in [29] that allows *quantifying the actual probability that a source is truthful or that an assertion is true*. Whereas we share the same goal of assessing the credibility of different data sources, we cannot directly apply fact finding techniques. The key reason is that fact finding techniques tend to work well when a large number of sources are used to report on the same state (e.g. social sensing), and is therefore not suitable for traditional sensor networks, where only very few sensors typically detect and report the same event. The work proposed in [30] uses a tree of regression models to minimize the estimation error (maximizing the quality of information) within certain cost budget. This work is different in that we do not possess knowledge of the real states, and thus cannot use it to train regression models.

In summary, to date, accuracy estimation for sensor networks has been done primarily through learning techniques such as the EM algorithms. The parameters of sensor measurement models are first determined and they are then used to estimate the accuracy of sensor observations. Inference has been limited to state estimation problems, and has seen little use in the context of accuracy estimation. The learning algorithms that have been used for accuracy estimation are carefully designed to fit the application under consideration (for example, [3] has assumed non-dynamic state, whereas [6] and [4] consider dynamic time-varying state sequences), and have not been compared with each other. To our knowledge, the impact of considering system dynamics or not on the ability to estimate sensor accuracy has not been studied in real-world scenarios. Research efforts have clearly focussed on using learning (parameter estimation) algorithms to estimate the accuracy of sensor measurements, and have shown little attention to inference algorithms. To our knowledge there are currently no empirical studies that compare both inference and learning algorithms in the context of accuracy estimation using real datasets from different sensing applications.

9 CONCLUSION AND FUTURE WORK

In this paper we studied the problem of estimating the accuracy of one or more coexisting sensor systems based on the probabilistic measurements they generate. We

proposed a general accuracy estimation framework, which breaks the problem down to layers and addresses it step by step. We show that the framework can be used in various cases, and implemented in different ways: for the state estimation layer, we created a taxonomy of techniques, including simple voting, inference-based and learning-based approaches, and explained their differences; for the accuracy estimation layer, we introduced two accuracy metrics, a proximity-based and a similarity-based, and shown that they can be used by applications with different types of accuracy requirements. Finally, we evaluated the performance of different implementations of the proposed accuracy estimation framework in two real-world sensing scenarios, which generate probabilistic location and light intensity data.

Our key findings are as follows: 1) the accuracy of sensor systems can vary significantly in different contexts, and typically no system is superior overall; 2) the accuracy estimation performance and running time of different accuracy estimation techniques can be very different; 3) in our case where only a few systems are available, static inference and learning are only marginally better than voting, but much more expensive; 4) dynamic inference and learning are significantly better since the correlations between monitored states are considered; 5) learning is preferred to inference only if the initial knowledge on model parameters is poor, at a much higher computation cost; 6) the more the coexisting sensor systems, the greater the relative benefits of voting (compared to trusting reported accuracy), and of dynamic inference and learning (compared to voting); 7) prior knowledge on states can significantly impact the performance of different algorithms: whereas voting and static algorithms improve linearly, dynamic techniques improve faster with fewer priors since they exploit the state transitions to propagate priors to nearby states.

A limitation of this work is that we have used directed graphical models and recursive Bayesian techniques to capture the correlations between the states and measurements. For future research, we plan to consider undirected graphical models (e.g. CRFs) for inference and learning in the context of accuracy estimation. Another limitation is that the prior knowledge considered is restricted to single timestamps, while in the future, we will extend this work to incorporate other forms of prior knowledge on states, which possibly span over multiple timestamps.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support of the EPSRC through grants EP/G069557/1 and EP/J012017/1, and thank the anonymous reviewers for their feedback.

REFERENCES

- [1] V. Moghtadaiee, A. Dempster, and S. Lim, “Indoor localization using fm radio signals: A fingerprinting approach,” in *Proc. Indoor Positioning Indoor Navigation*, 2011, pp. 1–7.
- [2] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, “Efficient, generalized indoor wifi graphslam,” in *Proc. Int. Conf. Robot. Automat.*, 2011, pp. 1038–1043.
- [3] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, “On truth discovery in social sensing: A maximum likelihood estimation approach,” in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, 2012, pp. 233–244.

- [4] D. Wang, T. Abdelzaher, L. Kaplan, and C. Aggarwal, "Recursive fact-finding: A streaming approach to truth estimation in crowdsourcing applications," in *Proc. ICDCS*, 2013, pp. 530–539.
- [5] D. Wang, L. Kaplan, and T. F. Abdelzaher, "Maximum likelihood analysis of conflicting observations in social sensing," *ACM Trans. Sen. Netw.*, vol. 10, no. 2, pp. 30:1–30:27, 2014.
- [6] H. Wen, Z. Xiao, N. Trigoni, and P. Blunsom, "On assessing the accuracy of positioning systems in indoor environments," in *Proc. Wireless Sensor Netw.*, 2013, pp. 1–17.
- [7] D. Hasenfratz, O. Saukh, and L. Thiele, "Model-driven accuracy bounds for noisy sensor readings," in *Proc. DCoSS*, 2013, pp. 165–174.
- [8] L. Reznik and V. Kreinovich, "Fuzzy and probabilistic models of association information in sensor networks," in *Proc. FUZZ-IEEE*, 2004, pp. 185–189.
- [9] E. Elnahrawy and B. Nath, "Cleaning and querying noisy sensors," in *Proc. Wireless Sensor Netw. Appl.*, 2003, pp. 78–87.
- [10] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2006.
- [11] E. Pacuit, "Voting methods," in *The Stanford Encyclopedia of Philosophy*, Metaphysics Research Lab, Stanford University, Stanford, CA, USA, Winter 2012 ed., 2012.
- [12] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Statist. Soc.. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.
- [13] E. Süli and D. F. Mayers, *An Introduction to Numerical Analysis*, Cambridge, U.K.: Cambridge Unive. Press, 2003.
- [14] A. P. Pentland, "Interpolation using wavelet bases," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 4, pp. 410–414, Apr. 1994.
- [15] S. Madden. (2004). Intel lab data. [Online]. Available: <http://db.csail.mit.edu/labdata/labdata.html>
- [16] N. Cressie. (1990). The origins of kriging. *Mathematical Geology*, vol. 22, pp. 239–252. [Online]. Available: <http://dx.doi.org/10.1007/BF00889887>
- [17] M. A. Osborne, S. J. Roberts, A. Rogers, and N. R. Jennings, "Real-time information processing of environmental sensor network data using bayesian gaussian processes," *ACM Trans. Sen. Netw.*, vol. 9, no. 1, pp. 1:1–1:32, 2012.
- [18] B. Hummel, "Map matching for vehicle guidance," in *Dynamic and Mobile GIS: Investigating Space and Time*. Boca Raton, FL, USA: CRC Press, 2006.
- [19] J. Krumm, J. Letchner, and E. Horvitz, "Map matching with travel time constraints," in *Proc. SAE World Congr.*, 2007, pp. 1–7.
- [20] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proc. SenSys*, 2009, pp. 85–98.
- [21] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *Proc. ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2009, pp. 336–343.
- [22] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson, "Easytracker: Automatic transit tracking, mapping, and arrival time prediction using smartphones," in *Proc. SenSys*, 2011, pp. 68–81.
- [23] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, low-energy trajectory mapping for mobile devices," in *Proc. 8th USENIX Conf. Netw. Syst. Design Implementation*, 2011, pp. 267–280.
- [24] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," in *Proc. IEEE Pervasive Comput.*, 2003, pp. 24–33.
- [25] J. Schmid, F. Beutler, B. Noack, U. Hanebeck, and K. D. Müller-Glaser, "An experimental evaluation of position estimation methods for person localization in wireless sensor networks," in *Proc. Wireless Sensor Netw.*, vol. 6567, pp. 147–162, 2011.
- [26] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, 1999.
- [27] X. Yin, J. Han, and P. Yu, "Truth discovery with multiple conflicting information providers on the web," in *Proc. IEEE Trans. Knowl. Data Eng.*, 2007, pp. 796–808.
- [28] J. Pasternack and D. Roth, "Knowing what to believe (when you already know something)," in *Proc. 23rd Int. Conf. Comput. Linguistics*, 2010, pp. 877–885.
- [29] D. Wang, T. Abdelzaher, H. Ahmadi, J. Pasternack, D. Roth, M. Gupta, J. Han, O. Fatemeh, H. Le, and C. Aggarwal, "On bayesian interpretation of fact-finding in information networks," in *Proc. FUSION*, 2011, pp. 1–8.

- [30] D. Wang, H. Ahmadi, T. Abdelzaher, H. Chenji, R. Stoleru, and C. Aggarwal, "Optimizing quality-of-information in cost-sensitive sensor data fusion," in *Proc. DCoSS*, 2011, pp. 1–8.



Hongkai Wen received the PhD degree in computer science from the University of Oxford, Oxford, United Kingdom. He is currently a post-doctoral researcher in the Department of Computer Science, University of Oxford. His current research interests include sensor networks, localization and navigation, and probabilistic machine learning.



Zhuoling Xiao is currently working toward the PhD degree in the Department of Computer Science, University of Oxford, Oxford, United Kingdom. His current research interests include sensor networks, including localization, communication and coordination protocols for networked sensor nodes, and machine learning techniques for sensor networks and localization.



Andrew Markham received the BScEng (Hons) and PhD degrees, in 2004 and 2008, respectively, from the University of Cape Town, Cape Town, South Africa. He is currently an associate professor of software engineering in the Department of Computer Science, University of Oxford, Oxford, United Kingdom. His current research interests include low power sensing and communication.



Niki Trigoni received the PhD degree at the University of Cambridge, Cambridge, United Kingdom, in 2001. She is currently an associate professor in the Department of Computer Science, University of Oxford, Oxford, United Kingdom. She became a postdoctoral researcher at Cornell University from 2002 to 2004, and a Lecturer at Birkbeck College from 2004 to 2007. Since she moved to Oxford in 2007, she established the Sensor Networks Group, and has conducted research in communication, localization and in-network processing algorithms for sensor networks. Her recent and ongoing projects span a wide variety of sensor networks applications, including indoor/underground localization, wildlife sensing, road traffic monitoring, autonomous (aerial and ground) vehicles, and sensor networks for industrial processes.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.