Comparison of Accuracy Estimation Approaches for Sensor Networks

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

Department of Computer Science, University of Oxford, Oxford, OX1 3QD, UK
{name.lastname}@cs.ox.ac.uk

***Abstract*—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 are often annotated with some measure of uncertainty, such as the distribution variance or a confidence interval, and will be hereafter referred to as *probabilistic measurements*. The question that we address in this paper is how to estimate the accuracy of these probabilistic measurements, that is, how far they lie from the ground truth of the measured attribute. Existing studies on estimating the accuracy of probabilistic 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 approaches that rely on simple state estimation without learning. Third, they do not explore whether exploiting the dynamics of the monitored state can yield significant benefits in terms of accuracy estimation. In this paper, we address the above limitations as follows: We define the problem of accuracy estimation in a general way that applies to a wide spectrum of application scenarios. We then propose a taxonomy of accuracy estimation techniques, which include both state estimation and parameter learning. These techniques are further subdivided into static and dynamic, depending on whether they exploit knowledge of system dynamics. All different approaches in the taxonomy are then applied and compared with each other in the context of two real sensing applications. We discuss how they trade accuracy for computation cost, and how this tradeoff largely depends on the user's knowledge of the application scenario.**

I. Introduction

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% confidence interval), and our objective is to estimate how far it lies from the ground truth (given a distance metric).

We first explain why this problem is important in a number of different settings. First, knowing how much to trust the measurements of a sensor network is paramount to deciding whether to use or pay for the service it offers. For example, if an indoor positioning system reports the user location with a very small uncertainty ellipse, but the user is often found outside that ellipse, then the user should have a way of detecting the poor accuracy of the service. Second, a user may be faced with the choice of selecting among multiple co-located sensor networks that offer a similar service (e.g. a WiFi-based vs. an FM-based indoor tracking system). In this case, they should be in a position to select the most accurate one, which may not necessarily be the one with the smallest *reported* position uncertainty. Third, when a sensor network is first deployed, the network 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 the default assumed accuracy. Finally, the emergence of social sensing applications has raised the challenge of estimating the trustworthiness of human participants. When people report probabilistic sensor measurements (say, estimated mean and variance of 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 data sources generate probabilistic observations (e.g. mean and variance pairs), and the goal is to estimate the accuracy of these observations. If the ground truth of the state was known, then the problem would be trivial. In the absence of ground truth, there have been very few efforts to tackle this problem. For example, the work in [20] estimates the correctness of measurements and the reliability of participants in social sensing applications by solving an expectation maximization (EM) problem. Our previous work shows how to assess the accuracy of co-located positioning systems by extending the Baum-Welch algorithm - an expectation maximization algorithm for dynamic (HMM) systems [21]. These papers use different algorithms that are tailored to the specific application scenario. To our knowledge, there is currently no systematic study that compares the algorithms proposed for different applications in a common experimental setup. More importantly, most existing work has tackled the problem of accuracy estimation by employing *learning* (EM) techniques. In this paper we advocate that *inference* techniques are equally viable alternatives for accuracy estimation, and they should not be confined to their traditional use in state estimation problems. We show that, in certain cases, inference techniques can offer more attractive tradeoffs between computational cost and estimation accuracy than their learning counterparts. To summarize, the contributions of this paper are as follows:

1) We define the problem of accuracy estimation for sensor networks 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

978-0-7695-5041-1/13 $26.00 © 2013 IEEE
DOI 10.1109/DCOSS.2013.56

trustworthiness of different individuals in social sensing.

2) We show that the problem can be addressed in many different ways. We create a taxonomy of accuracy estimation techniques for sensor networks, which covers from simple voting schemes, to state 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. Finally, we show that any additional prior information on the monitored phenomenon can be easily incorporated into both inference and learning techniques in both the static and dynamic cases. This is the first study where all these techniques are juxtaposed and compared in a single taxonomy for solving the accuracy estimation problem for sensor networks.

3) We perform a systematic empirical study to compare the performance of the various accuracy estimation approaches in our taxonomy in the context of two real sensor datasets, one containing position sensor measurements in an indoor environment, and the other containing light intensity sensor measurements. The purpose of our study is to explore the following issues: 1) Learning is inherently more computationally expensive than inference: is the increase in estimation accuracy worth the extra cost? 2) Does exploiting the dynamics of a process lead to significantly higher accuracy and at what computational cost? 3) To what extent is the relative performance of the various estimation algorithms sensitive to sensor network co-location? 4) To what extent is the relative performance of the various accuracy estimation algorithms sensitive to the application scenario?

The remainder of this paper is organized as follows: Section II defines the problem of accuracy estimation in the context of one or more co-located sensor systems. Section III describes the taxonomy of voting, inference and learning approaches that can be used to tackle the accuracy estimation problem, and highlights their differences. Section IV shows the feasibility of all approaches in the proposed taxonomy in the context of two real sensor datasets, and evaluates their performance in terms of accuracy and computation cost. Section V overviews related work, while Section VI concludes the paper and discusses the future work.

II. The Accuracy Estimation Problem

Let x_t be the real value of the signal that a sensor system is measuring (e.g. temperature or the location of a user), and z_t be the sensor measurement generated at discrete time t, where $1 \leq t \leq T$. Without loss of generality, we focus on monitoring dynamic processes, where the signal state changes over time. Stationary processes can be viewed as a special case with $T = 1$, i.e. where measurements taken at different timestamps are all collapsed to one timestamp, $T = 1$.

We assume z_t is a random variable, and at each timestamp t a sensor system generates a probability distribution $p(z_t)$ that approximates the ground truth state x_t. Given x_t, the accuracy of the probabilistic measurement z_t depends on its expected distance from the real state: $\epsilon(z_t; x_t) = E[C(z_t - x_t)]$, where $C(\cdot)$ is a cost function. In this paper we use the quadratic form $C(z_t - x_t) = \|z_t - x_t\|^2$. Then the accuracy of z_t is given by

$$\epsilon(z_t; x_t) = E[\|z_t - x_t\|^2] = \int_{z_t} p(z_t)\|z_t - x_t\|^2 \, \mathrm{d}z_t \qquad (1)$$

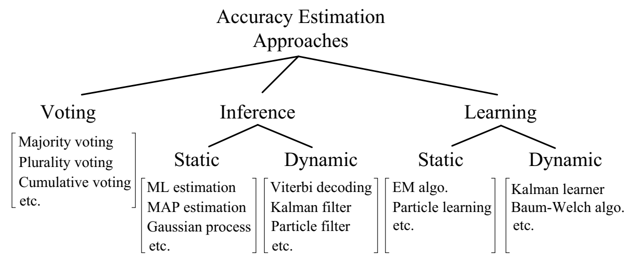


Fig. 1: The taxonomy of accuracy estimation approaches for sensor networks. For each line of approaches, several representative techniques are listed.

where $p(z_t)$ is the probability distribution of the sensor measurement z_t. For generality, we do not confine our study to one sensor system, but consider the case of multiple co-located sensor systems, each reporting its own probabilistic measurement. In this case, our goal is to estimate the measurement accuracies of all co-located sensor systems.

We also assume that in certain timestamps, users may possess prior knowledge of how the state x_t is distributed. For instance, consider an indoor localization scenario where the states are the actual locations of the user. Planned events such as calendar entires, or social interactions like store check-ins, may directly reveal the user location at a given time [21]. We refer to such information as *user-provided priors*, and use $p(x_t)$ to represent the prior distribution on the state at time t.

As shown in Eqn. 1, given the measurement distribution $p(z_t)$, its accuracy is a function of the state x_t, which is typically unknown. Therefore the actual accuracy $\epsilon(z_t; x_t)$ can not be evaluated directly. The *accuracy estimation problem* is to approximate the accuracy of a measurement given the stochastic sensor observations and priors on the states.

III. Taxonomy of Accuracy Estimation Approaches

In this section, we design a taxonomy consisting of the three main approaches that can be used to solve the accuracy estimation problem: voting, state inference and learning. The inference-based 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. All approaches share two key steps: they first produce an estimate of the ground truth $\hat{x}_t$, and they then estimate the accuracy of the probabilistic sensor measurement z_t with respect to the ground truth estimate, i.e. $\epsilon(z_t; \hat{x}_t)$. The three approaches vary in how they estimate the ground truth $\hat{x}_t$. Fig. 1 illustrates the taxonomy of the accuracy estimation approaches discussed above, along with specific examples of techniques under each class. We are now in a position to describe the various approaches in more detail. In Section IV we will evaluate and compare them in the context of two real sensing applications.

A. Voting-based approach

Voting is a widely used approach when multiple co-located sensor systems generate measurements about the same state. In general, voting procedures help aggregate the preferences from individual sensor systems to achieve a collective decision. In the context of accuracy estimation, the voting approach works on one timestamp at a time, say at time t. It combines the measurement distributions $p(z_t^i)$ from M co-located sensor systems ($1 \leq i \leq M$) into a single probability distribution (using equal weights for all sensor systems). It then estimates the latent state at time t according to certain voting rules, such as majority or plurality [11], and uses the estimated state as an approximation of the ground truth to evaluate the accuracy of each stochastic sensor measurement z_t^i.

B. Inference-based approach

The inference-based approach estimates the state, based on the measurements generated by one or more sensor systems, and prior information on the state provided by the user. In this paper, we consider the widely used *maximum a posteriori* (MAP) estimator, which chooses the estimate that maximizes the posterior distribution of the state given the measurements. The key assumption here is that the measurement model (the probability of a measurement given a state) is known a priori.

Static inference The static inference approach ignores any temporal correlations between the latent states, and only accesses the measurements and priors at a single timestamp t. For simplicity, suppose that $M = 2$ co-located sensor systems are monitoring state x_t (our discussion can be easily extended to $M > 2$ systems). The MAP estimate of x_t is given by:

$$\hat{x}_{t_{\text{MAP}}} = \arg\max_{x_t} p(x_t|z_t^1, z_t^2) = \arg\max_{x_t} p(z_t^1, z_t^2|x_t)p(x_t) \tag{2}$$

where $p(z_t^1, z_t^2|x_t)$ is the probability of observing the measurements z_t^1, z_t^2 given x_t, and $p(x_t)$ is the prior belief on the state. $\hat{x}_{t_{\text{MAP}}}$ is the mode of the posterior distribution $p(x_t|z_t^1, z_t^2)$, and can be computed analytically if the distribution is given in closed form, or via numerical or Monte Carlo approaches in more general cases. If no external noise is considered, and the measurements z_t^1, z_t^2 are conditionally independent given the state x_t, Eqn. 2 can be factored as:

$$\hat{x}_{t_{\text{MAP}}} = \arg\max_{x_t} p(z_t^1|x_t)p(z_t^2|x_t)p(x_t) \tag{3}$$

If a user-provided prior is available at time t, then $p(x_t)$ can carry such knowledge and bias the estimated state towards the belief of the user.

Dynamic inference The second variant of the inference approach exploits the dynamics of the monitored state. It assumes both the measurement model and the state transition model are known a priori. Let the states be a time-varying sequence: $x_1, \ldots, x_T$, with measurements $z_1, \ldots, z_T$ from a sensor system. In practice, the dynamics are usually assumed to be *Markovian* for simplicity, i.e. $p(x_t|x_{1:t-1}) = p(x_t|x_{t-1})$, and the measurements are considered to be independent conditioned on the states. The dynamic inference approach estimates the most likely state sequence $x_{1:T}$ with all observed measurements:

$$\begin{aligned} \hat{x}_{1:T_{\text{MAP}}} &= \arg\max_{x_{1:T}} p(x_{1:T}|z_{1:T}) \\ &= \arg\max_{x_{1:T}} p(x_1)p(z_1|x_1) \prod_{t=2}^{T} p(x_t|x_{t-1})p(z_t|x_t) \end{aligned} \tag{4}$$

which shows that for any given time t, the posterior distribution of x_t is influenced by all measurements, linked by the state transition probabilities. Experiments in Sec. IV confirm that this can improve the state estimation, and further has a knock-on effect on the quality of accuracy estimation. Eqn. 4 can be easily extended to accommodate measurements from multiple sensor networks, as in Eqn. 3.

If for certain timestamps t, the user has prior information on the state, then the state transition probabilities $p(x_t|x_{t-1})$ in Eqn. 4 will be scaled by the prior $p(x_t)$.

C. Learning-based approach

Unlike the inference-based approach, the learning-based approach does not assume prior knowledge on the measurement model $p(z|x)$. It starts with an estimate of the model, and iteratively refines this estimate to be consistent with the observed data. It then uses the learnt model to estimate the latent states. Finally it evaluates the accuracy of the measurements with the inferred states.

Static learning The static learning approach operates on the measurements received within one timestamp, and does not consider the dynamics of the monitored state. Let x_t be the state at a given timestamp t. For simplicity, let's assume that at time t there are only two co-located sensor systems, which report measurements z_t^1 and z_t^2. The measurement model $\theta = p(z|x)$ is assumed to be an unknown parameter, and the goal of the learning step is to find the θ that best explains the observed measurements z_t^1 and z_t^2. This is given by the maximum likelihood estimate (ML estimate) $\hat{\theta}_{\text{ML}} = \arg\max_\theta p(z_t^1, z_t^2|\theta)$. As in general latent variable models, $\hat{\theta}_{\text{ML}}$ can be computed by the Expectation Maximization (EM) approach [3]. The EM algorithm works iteratively with the following two steps until it converges to a local maximum:

a) The *E-step*, which computes the expected log likelihood of the data given the current parameter setting $\theta^{(i)}$:

$$\begin{aligned} Q(\theta, \theta^{(i)}) &= E_{z_t^1, z_t^2|x_t, \theta^{(i)}}[\log p(z_t^1, z_t^2, x_t|\theta)] \\ &= \int_{x_t} p(z_t^1, z_t^2, x_t|\theta^{(i)}) \log p(z_t^1, z_t^2, x_t|\theta) \, \mathrm{d}x_t \end{aligned} \tag{5}$$

b) The *M-step*, which finds the new parameter $\theta^{(i+1)}$ that maximizes the function $Q(\theta, \theta^{(i)})$:

$$\theta^{(i+1)} = \arg\max_\theta Q(\theta, \theta^{(i)}) \tag{6}$$

Under the assumption of conditional independence between measurements, it is also possible to factor the term $p(z_t^1, z_t^2|x_t)$ as $p(z_t^1, |x_t)p(z_t^2|x_t)$, which could further simplify the computation. If a user-provided prior is available at timestamp t, the prior distribution $p(x_t)$ can be directly included into the derivation of the Q function when evaluating the joint likelihood $p(z_t^1, z_t^2, x_t|\theta^{(i)}) = p(z_t^1, z_t^2|\theta^{(i)}, x_t)p(x_t)$.

With the learnt model parameter θ_{ML}, the static learning approach estimates the state $\hat{x}_t$ as in static inference, and it uses $\hat{x}_t$ as an approximation of the ground truth to evaluate the accuracy of the measurements z_t^1 and z_t^2 respectively.

Dynamic learning Similar to dynamic inference, the dynamic learning approach also assumes the hidden state varies over time. But instead of relying on the model parameters known in advance, the dynamic learning approach takes all measurements into account and learns the parameters from the data observed. This approach is capable of learning both the state transition model and the measurement model. However, re-estimating the state transition model is optional: if we have solid knowledge on the state transition probabilities, then it would be better to leave them untouched.

Dynamic learning also uses the EM scheme, which iteratively finds the best parameters until it converges. But the derivation of the likelihood function (the Q function in Eqn. 5) is different from the static case. We make identical assumptions as discussed in dynamic inference, i.e. $x_{1:T}$ is the state sequence with Markovian property, and $z_{1:T}$ are the measurements from a sensor network. For simplicity, we still consider only one network, and the technique can be extended to the multiple case with similar formulation as in Eqn. 5. We also assume that the approach does not re-estimate the state transition model, but only learns the measurement model $\theta = p(z|x)$. Then the function $Q(\theta, \theta^{(i)})$ becomes:

$$Q(\theta, \theta^{(i)}) = E_{z_{1:T}|x,\theta^{(i)}}[\log p(z_{1:T}, x_{1:T}|\theta)] \\ = \int_{x_{1:T}} p(z_{1:T}, x_{1:T}|\theta^{i}) \log p(z_{1:T}, x_{1:T}|\theta) \, \mathrm{d}x_{1:T} \tag{7}$$

where the term $p(z_{1:T}, x_{1:T}|\theta)$ is the joint likelihood given θ:

$$p(z_{1:T}, x_{1:T}|\theta) = p(x_1)p(z_1|x_1) \prod_{t=2}^{T} p(x_t|x_{t-1})p(z_t|x_t) \tag{8}$$

The maximization step is the same as in static case. Note that in Eqn. 7, the integral (or summation) is over all possible latent state sequences $x_{1:T}$, which can be computed efficiently with dynamic programming in some special cases, such as forward-backward algorithm in Hidden Markov Models [13]. In the presence of user-provided priors at a given timestamp t, the prior distribution $p(x_t)$ can be incorporated to Eqn. 8 in the same way as in dynamic inference. Prior information can therefore influence the learnt measurement model and the estimated state sequence $\hat{x}_{1:T}$.

IV. Evaluation

A. Experiment setup

We evaluate and compare the accuracy estimation approaches on two datasets from real sensor deployments.

Indoor tracking dataset 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. 2(a). 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 to log their coordinates.

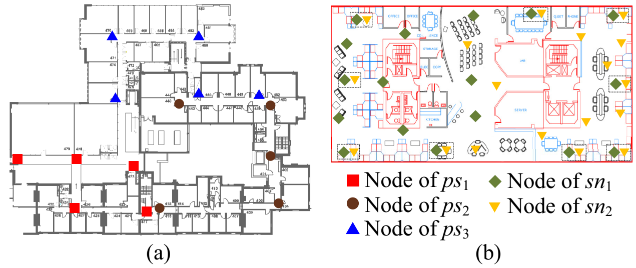


Fig. 2: (a) Experiment setup of the indoor tracking dataset (left) and the light intensity dataset (right).

We tracked a 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 5 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 0.5Hz. We assume space is discrete, i.e. it is a finite set $L = \{l_i\}$ with N discretized locations, e.g. different rooms or corridor segments. In our experiment, the average size of a discrete location is 3m×3m 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.

Light intensity dataset The data was collected from the Intel Lab dataset [8]. The dataset contains temperature, humidity and light data collected from 54 sensors deployed in a lab environment for more than a month. We selected the light intensity readings of approximately 5 consecutive days for our experiments. We divided the 51 sensors (3 sensors are omitted since they failed midway) into two groups randomly, where data from 26 of them were used to generate stochastic light measurements (as explained below), and the rest were used as the ground truth.

We created two co-located sensor networks, sn_1 and sn_2, by selecting different subsets of the 26 training sensors, as shown in Fig. 2(b), where sensors that are virtually "shared" by the two networks are grouped by rectangles. We then applied Gaussian process non-linear regression to interpolate the data from each network across the space. Therefore, for a given timestamp t and a given point in space, the measurement from sensor network sn_1 or sn_2 is a Gaussian distribution $\mathcal{N}(\mu, \sigma)$, with an estimated light intensity value μ and variance σ.

B. Competing algorithms

We implemented the following representative algorithms for accuracy estimation, which cover all of the approaches discussed in Sec. III:

a) ***Oracle Algorithm (OA)*** has access to the latent state x_t (i.e. the ground truth), and for a given measurement z_t, the accuracy it computes is the real accuracy, i.e. $\epsilon_{oracle}(z_t) = \epsilon(z_t; x_t)$. For clarity, we omit the state estimate used to evaluate the accuracy where it is unambiguous.

b) ***Report-based Algorithm (RA)*** uses the mean $\bar{z}_t$ of the measurement as an approximation of the ground truth to evaluate the accuracy, i.e. $\epsilon_R(z_t) = \epsilon(z_t; \bar{z}_t)$.

c) ***Voting-based Algorithm (VA)*** takes the measurements from all co-located networks at a given time t into account. For each timestamp t, the algorithm approximates the latent state x_t by cumulative voting, i.e. it finds the state estimate $\hat{x}_t$ with the largest probability, and uses it to evaluate the accuracy of a measurement $\epsilon_V(z_t)$.

d) ***Static Inference-based Algorithm (SIA)*** considers all sensor measurements generated at timestamp t. It infers the maximum a posteriori (MAP) estimate of the state. The measurement model $p(z_t|x_t)$ for each sensor network is derived from the distributions of all stochastic measurements of that network whose mean is equal to x_t, i.e. the measurement means are used to approximate the ground truth x_t when generating $p(z_t|x_t)$. It uses this state estimate to compute the accuracy $\epsilon_{SI}(z_t)$.

e) ***Static Learning-based Algorithm (SLA)*** first learns the measurement model $\hat{\theta} = \hat{p}(z_t|x_t)$ that is most consistent with the observed data at timestamp t, which is given by the maximum likelihood estimate (MLE) $\hat{\theta}_{\text{ML}}$. Then it computes the MAP state estimate $\hat{x}_{t_{\text{MAP}}}$ with the learnt model in the same way as *SIA* does, and uses this estimate to evaluate accuracy $\epsilon_{SL}(z_t)$.

f) ***Dynamic Inference-based Algorithm (DIA)*** assumes known system dynamics (state transition model) and exploits this knowledge to infer the most probable state sequence $\hat{x}_{1:T_{\text{MAP}}}$ with measurements at all timestamps. The measurement models are directly derived from the stochastic measurements as in *SIA*. The algorithm then evaluates the accuracy of a given measurement z_t with the t-th element in the estimated state sequence as $\epsilon_{DI}(z_t)$.

g) ***Dynamic Learning-based Algorithm (DLA)*** considers measurements at all timestamps and recalibrates the measurement model $\theta = \hat{p}(z|x)$ with its MLE $\hat{\theta}_{\text{ML}}$. It then uses the learnt measurement model and known state transition model to estimate the MAP state sequence $\hat{x}_{1:T_{\text{MAP}}}$. It uses the state estimate at time t to compute accuracy $\epsilon_{DL}(z_t)$.

In the learning algorithms (*SLA* and *DLA*) the measurement models are initialized in the same way as in the inference algorithms (*SIA* and *DIA*). Note that the last four algorithms (*SIA*, *SLA*, *DIA* and *DLA*) can also incorporate user-provided priors, as discussed in Sec. III. We evaluate the above competing algorithms against the metric of ***Estimation Error EE***. For a given measurement z_t, *EE* is defined as the L_1 distance between the estimated accuracy $\epsilon(z_t)$ and the ground truth accuracy $\epsilon_{oracle}(z_t)$: *EE* $= |\epsilon(z_t) - \epsilon_{oracle}(z_t)|$.

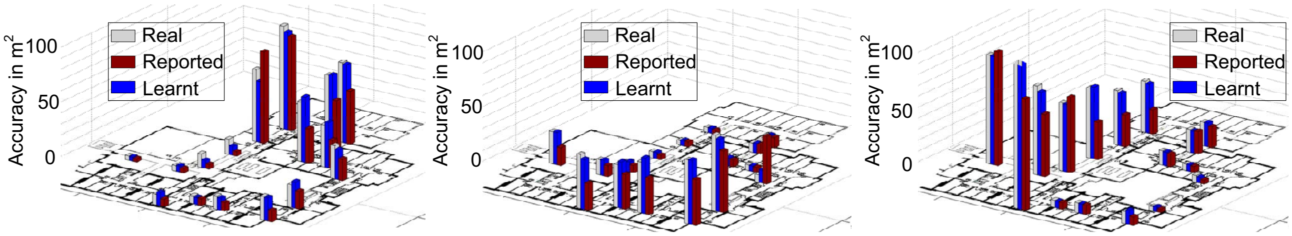
C. Experiment results

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 networks varies over time and space: The first set of experiments show that the accuracy of a sensor network can vary over time and space, while the reported accuracy may not be a good indicator of the real accuracy. For the tracking dataset, Fig. 3 shows that the real accuracy (averaged over all timestamps) of the co-located positioning systems $ps_1 \sim ps_3$ (the grey 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 grey 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; the reported accuracy (red bars) consistently over or under estimates the real accuracy (grey bars). The accuracy computed by *DLA* (blue bars) is much closer to the real accuracy (grey bars). This shows that in the absence of ground truth, the real accuracy can be effectively approximated by applying suitable techniques. For the light intensity dataset, Fig. 4 shows that the real accuracy of a sensor network can vary over both location and time. Figs. 4(a) and 4(b) show snapshots of the light measurements reported at daytime by networks sn_1 and sn_2 respectively. We can see that: a) the differences between the real light values and reported ones vary across space, and b) the reported accuracy (variance) is very unreliable and the real light values consistently fall out of the 95% intervals of the reported ones. Fig. 4(c) shows a snapshot of real and reported values (by sn_1) at night; notice that the differences here between real and reported values are very small, which suggests that sn_1 becomes more accurate at night.

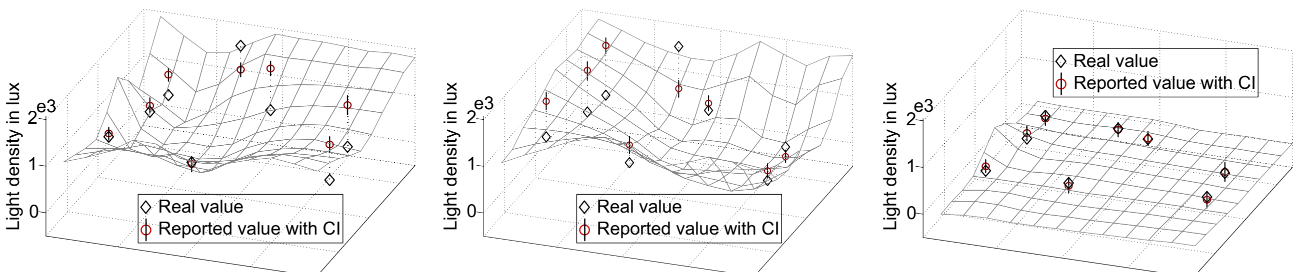
Comparison of accuracy estimation algorithms: The second set of experiments compares the performances of the accuracy estimation algorithms in terms of *Estimation Error* (*EE*), averaged over all measurements. The left graph of Fig. 5(a) shows the estimation error of different algorithms on the tracking dataset when we consider the stochastic 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 40%, which means that measurements from the co-located sensor networks can indeed help to improve the estimation of accuracy, and simple approaches like voting could be quite effective in practice. The differences between techniques that operate on the measurements of a single timestamp (*VA*, *SIA* and *SLA*) are negligible, due to the limited number of information sources. When moving to dynamic approaches, however, we observe significant improvements in estimation error. Dynamic inference (*DIA*) features about 30% 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 approximately 50% benefit compared to voting, since *DLA* also learns a new measurement model that best explains the stochastic measurements. Finally, the improvement from the naive approach (*RA*) to the best technique (*DLA*) is almost 3-fold. For the light intensity data, as shown in Fig. 5(b), 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 negligible, because we use a naive measurement model (that is derived directly from reported confidence intervals, which often do not cover the ground truth as shown in Fig. 4). The benefits of *DLA*, however, are far more pronounced, since the learnt measurement model is more accurate. Fig. 5(c) 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. 5(a) shows the results on the tracking data. We can see that with fewer co-located networks, the performance gaps between the different techniques become smaller. In the case where only one network is available, the best performing algorithm (*DLA*) has similar



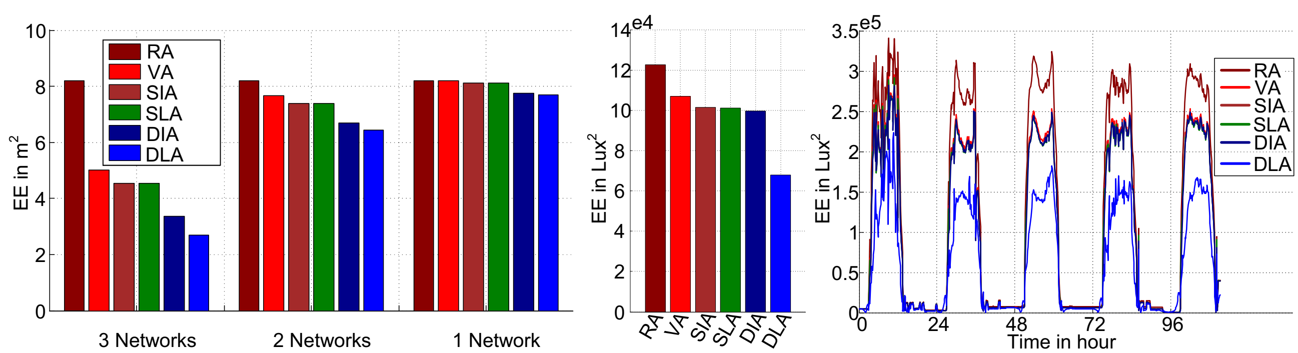
(a) (b) (c)

Fig. 3: The real, reported and learnt (computed by *DLA*) accuracy for positioning system ps_1 (left), ps_2 (middle) and ps_3 (right).



(a) (b) (c)

Fig. 4: 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 networks sn_1 (left) and sn_2 (middle). The right graph shows real and reported light intensity data generated by sn_1 at night.



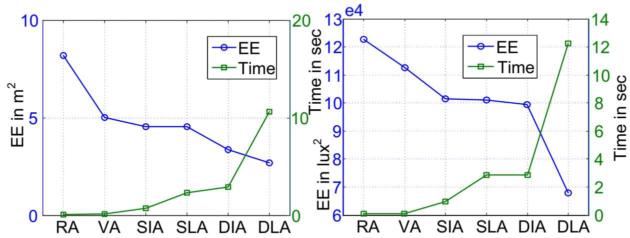
(a) (b) (c)

Fig. 5: (a) Average estimation errors of different approaches on tracking dataset when the number of co-located networks varies from 3 to 1. (b) Average estimation errors of different approaches on the light intensity dataset. (c) Estimation errors of different approaches on the light intensity dataset vary over time.

estimation error to the naive approach of trusting the reported accuracy (*RA*).

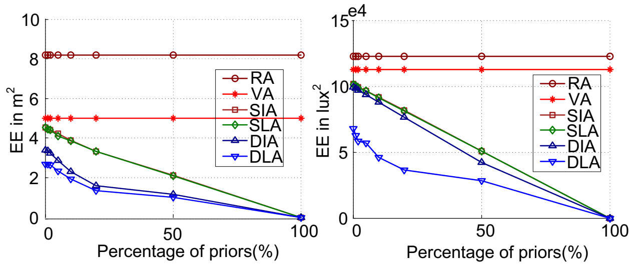
Running cost vs. performance gain: The fourth set of experiments studies the trade-off between accuracy estimation and computation cost. We measure the execution time of different algorithms and compare it with the performance gain in terms of estimation error (*EE*). Figs. 6 shows the trade-off on the tracking and light intensity dataset respectively. In general learning-based techniques are more expensive than 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 2∼3 times slower than inference but it can only improve the estimation performance by at most 30% (from *DIA* to *DLA* on the light intensity dataset). The results are similar on the tracking data, where moving from static inference to dynamic inference, the performance gain is about 25% at the expense of tripling the running time.

Sensitivity to the user-provided priors The last set of experiments shows how the user-provided prior can influence different accuracy estimation approaches. For both datasets, the priors are generated by first selecting a random subset of the timestamps. At these timestamps, the prior distribution $p(x)$ is set to be the ground truth value plus a small quantity



(a) (b)

Fig. 6: Running time vs. performance for different algorithms on the tracking dataset (left) and light intensity dataset (right).



(a) (b)

Fig. 7: Estimation error of different approaches when the percentage of priors varies on the tracking dataset (left) and light intensity dataset (right).

of noise. We vary the percentage of timestamps that have priors, and study the effect on the performance of accuracy estimation techniques (Fig. 7). On the tracking dataset, as the percentage of priors increases, the static approaches improve linearly. For dynamic approaches, the estimation error has a quick drop before the percentage of priors reaches 20%, and then becomes flat. This is because the dynamic approaches exploit temporal correlations in the data, which enables prior knowledge to impact future states. There is also a small gap between dynamic inference and learning, since learning can use priors to better assess measurement models. On the light intensity dataset, a similar behavior can be witnessed, but the gap between dynamic inference and learning is larger since the measurement model used by inference is trivial, while learning can recalibrate it from the data.

V. 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 [2] and Gaussian Processes (GPs) [14] to interpolate between sensor readings and infer the values of environmental variables in places where there 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 [10].

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 [5], [7], [17], [9], [1]. More specifically, VTrack uses mobile phones mounted in cars to estimate road travel times using a sequence of inaccurate position observations [17]. EasyTracker uses HMMs in the context of transit tracking, and uses the inferred tracks to detect transit stops and predict arrival times [1]. 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 [16]. 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 [4]. A recent comparative evaluation of different filters for person localization using RSS (Received Signal Strength) measurements is presented in [15].

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, [20] estimates both the correctness of measurements and the reliability of participants in social sensing applications by solving an expectation maximization problem. The work in [21] considers the indoor tracking problem, and assumes the colocation of multiple positioning systems that compete for the same users. It proposes an expectation maximization algorithm to learn the observation model (measurement probabilities) of each positioning system in various parts of the indoor environment. Note that both approaches [20], [21] first learn the parameters of the sensor measurement models, which can then be used to estimate the accuracy of sensor measurements. 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 [6], [22], [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 [6], 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 [18] 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 [19] uses a tree of regression models to minimize the estimation error (maximizing the quality of information) within certain cost budget. Our 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 EM algorithms. The parameters of measurement models are first determined and they are then used to estimate the accuracy of sensor measurements. 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, [20] has assumed non-dynamic state, whereas [21] considers a dynamic time-varying state), 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.

VI. Conclusion and Future work

In this paper we studied the problem of estimating the accuracy of stochastic measurements generated by one or more co-located sensor systems. We considered an array of techniques from simple voting, to inference-based and learning-based approaches, and discussed their differences. We then compared them in the context of two sensing applications for indoor environments, which generate probabilistic location and light intensity data. Our key findings from the empirical study are as follows: 1) Inference and learning are only marginally better than the voting approach in the static case (when the dynamics of the monitored state are not taken into account); voting is thus preferred among the three due to its simplicity and low cost. 2) Dynamic inference- and learning-based approaches, however, can significantly reduce the accuracy estimation error when compared to voting (by up to 50% in our scenarios). 3) The merits of learning vs. inference are more pronounced in the scenarios where we have a poor prior knowledge of the measurement model; when this is not the case, inference is preferred because its accuracy estimation performance is close to that of learning, at a much lower computation cost. 4) The more the co-located sensor systems, the greater the relative benefits of voting (compared to trusting reported accuracy), and of dynamic inference and learning (compared to voting). 5) Prior knowledge on the state distribution significantly impacts the relative performance of the algorithms: the more the timestamps on which we have user-provided priors, the smaller the estimation error of inference and learning algorithms. Whereas static approaches improve their performance linearly, dynamic approaches improve faster with fewer priors because they exploit the state transition model to propagate prior knowledge to nearby states.

A limitation of our work is that the accuracy of a sensor measurement is evaluated with respect to a single estimate of the ground truth, e.g. the mode of the state's posterior. In the future we will extend our approach to estimate accuracy against the entire posterior distribution of the state. We will also extend our 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 helpful feedback.

References

[1] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson. Easytracker: automatic transit tracking, mapping, and arrival time prediction using smartphones. In *Proc . SenSys*, 2011.

[2] N. Cressie. The origins of kriging. *Mathematical Geology*, 22:239–252, 1990.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, **39**(1), 1977.

[4] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. In *IEEE Pervasive Computing*, 2003.

[5] B. Hummel. Map matching for vehicle guidance. In *Dynamic and Mobile GIS: Investigating Space and Time*. CRC Press, 2006.

[6] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, **46**(5), 1999.

[7] J. Krumm, J. Letchner, and E. Horvitz. Map matching with travel time constraints. In *SAE World Congress*, 2007.

[8] S. Madden. Intel lab data, 2004.

[9] P. Newson and J. Krumm. Hidden markov map matching through noise and sparseness. In *Proc. GIS*, 2009.

[10] 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.*, **9**(1), 2012.

[11] E. Pacuit. Voting methods. In *The Stanford Encyclopedia of Philosophy*. Winter 2012 edition, 2012.

[12] J. Pasternack and D. Roth. Knowing what to believe (when you already know something). In *Proc . COLING*, 2010.

[13] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**(2), 1989.

[14] C. Rasmussen and C. Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, MA, 2006.

[15] 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 . EWSN*, 2011.

[16] A. Thiagarajan, L. Ravindranath, H.Balakrishnan, S. Madden, and L. Girod. Accurate, Low-Energy Trajectory Mapping for Mobile Devices. In *Proc . NSDI*, 2011.

[17] 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.

[18] D. Wang, T. Abdelzaher, H. Ahmadi, J. Pasternack, D. Roth, M. Gupta, J. Han, O. Fatemieh, H. Le, and C. Aggarwal. On bayesian interpretation of fact-finding in information networks. In *Proc . FUSION*, 2011.

[19] 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.

[20] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher. On truth discovery in social sensing: a maximum likelihood estimation approach. In *Proc . IPSN*, 2012.

[21] H. Wen, Z. Xiao, N. Trigoni, and P. Blunsom. On assessing the accuracy of positioning systems in indoor environments. In *Proc. EWSN*, 2013.

[22] X. Yin, J. Han, and P. Yu. Truth discovery with multiple conflicting information providers on the web. In *Proc . KDD*, 2007.