The Tradeoff between Energy Efficiency and User State Estimation Accuracy in Mobile Sensing

Yi Wang[1], Bhaskar Krishnamachari[1], Qing Zhao[2], and Murali Annavaram[1]

[1] Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, USA
{wangyi,bkrishna,annavara}@usc.edu
[2] Department of Electrical and Computer Engineering, University of California, Davis, USA
qzhao@ece.ucdavis.edu

Abstract. People-centric sensing and user state recognition can provide rich contextual information for various mobile applications and services. However, continuously capturing this contextual information on mobile devices drains device battery very quickly. In this paper, we study the tradeoff between device energy consumption and user state recognition accuracy from a novel perspective. We assume the user state evolves as a hidden discrete time Markov chain (DTMC) and an embedded sensor on mobile device discovers user state by performing a sensing observation. We investigate a stationary deterministic sensor sampling policy which assigns different sensor duty cycles based on different user states, and propose two state estimation mechanisms providing the best "guess" of user state sequence when observations are missing. We analyze the effect of varying sensor duty cycles on (a) device energy consumption and (b) user state estimation error, and visualize the tradeoff between the two numerically for a two-state setting.

Key words: mobile sensing, energy efficiency, user state estimation accuracy, tradeoff

1 Introduction

Mobile smart phones these days contain a wide range of features including sensing abilities using GPS, audio, WiFi, accelerometer, and so on. By utilizing all the sensing units and extracting more meaningful characteristics of users and surroundings in real time, applications can be more adaptive to the changing environment and user preferences. For instance, it would be convenient if the phones can automatically adjust the ring tone profile to appropriate volume and mode according to the surroundings and the events in which the user is participating. Or imagine an "automated twittering" application (Twitter: [10]), where the user's activity can be automatically recognized and updated online in real-time by mobile device instead of manual input from the user. As these kinds of applications require that the user context be automatically and correctly recognized, in this paper we use “user state” as an important way to represent human context information, similar to the work in [12]. A set of user states can be detected by embedded sensor on mobile device with high accuracy. For example, a single accelerometer is able to discriminate a set of human motions such as walking, running, riding a vehicle and so on with high confidence.

Although we believe that user’s contextual information brings application personalization to new levels of sophistication, a big problem in context detection is that the limited battery capacity of mobile devices has restricted the continuous functioning of sensors without recharging the battery, because embedded sensors on mobile devices are major sources of energy consumption. For example, our experiments show that a fully charged battery on Nokia N95 device would be completely drained within six hours if its GPS is operated continuously. In this paper, we address this problem and study how to assign different duty cycles to a sensor at different states such that the energy consumption can be significantly reduced while maintaining high user state recognition accuracy.

In general, some user states once entered may stay unchanged for a long while; therefore, if the sensor is sampled too frequently within that state duration, an excessive amount of energy will be consumed. For example, if the user is identified to be sitting in a class, sensors such as accelerometer and microphone do not need to be monitored every minute if we know that on average the user is going to spend more than an hour in the class. On the other hand, if the user is in a mode of frequently transiting among different states, e.g., frequently traveling in an area with multiple tasks, we prefer to sample the corresponding sensor more frequently to capture his/her context, as otherwise it becomes harder to figure out “what happened” between two sensor observations.

In this work, we assume that time is discretized into slots. Ideally, if the sensor can be sampled in each time slot, the true user state sequence will be obtained perfectly. However, due to energy constraints the sensor have to adopt duty cycles in order to extend mobile device lifetime. We use a discrete time Markov chain (DTMC) to model the user states, with different state transition probabilities indicating different amounts of average duration a user spent in each state. We consider a stationary deterministic sensor sampling policy which assigns different duty cycles to the sensor based on different user states. In other words, at each user state the sensor will adopt a fixed sampling duty cycle. As a consequence, given the fact that the sensor does not make observation in every time slot, there exist periods of time when user state information is not observed and has to be estimated using only available data and knowledge of the transition probabilities of the user state Markov chain.

We propose two methods based on the Forward-Backward Algorithm and the Viterbi Algorithm [11, 8] in order to perform state estimation for those time slots with missing observation. In the first method, for each time slot we pick the most likely user state at that time slot and compute the overall expected per-slot estimation error. In the second method, we choose the most likely state sequence with the highest probability, and compute the probability that the estimated sequence is incorrect.

One can easily tell from intuition that if the sensor makes frequent observations, the user state estimation error should be small. In this paper, we derive the expression for both the expected energy consumption and the expected state estimation error in terms of the user state transition probability matrix and sensor duty cycle parameters, and visualize the tradeoff between them.

The rest of this paper is organized as follows. In section 2, we present relevant prior work. In section 3, we introduce our mathematical formulation of sensor duty cycle selection, propose two methods that estimate user state sequence with missing observation data, and for each method derive its expected energy consumption and the corresponding expected state estimation error. In section 4, we enable the visualization of the tradeoff between energy consumption and state estimation error numerically for a two-state DTMC and discuss our findings. Optimal sensor duty cycles are found for different state transition probability matrices and energy budgets. Finally, we conclude and present directions for future work in section 5.

2 Related Work

Embedded sensors on mobile phone such as GPS, Bluetooth, WiFi detector, accelerometer, and so on have been well studied and explored in order to conduct user activity recognition [7, 5, 1, 12]. For example, Annavaram *et al.* [1] show that by using data from multiple sensors and applying multi-model signal processing, seemingly similar states such as sitting and lying down can be accurately discriminated.

Due to the fact that low battery capacity on mobile device limits the application lifetime, the problem of energy management on mobile devices has been extensively studied in the literature such as [9, 6, 12]. Shih *et al.* [9] study event-driven power saving method and focused on reducing the *idle power*, the power a device consumes in a "standby" mode, such that a device turns off the wireless network adaptor to avoid energy waste while not actively used. In terms of sensor management, Wang *et al.* [12] propose a framework for energy efficient mobile sensing system (EEMSS), and show that by only operating necessary sensors and manage sensors hierarchically based on user state the device lifetime can be extended by more than 75% than existing similar systems while maintaining high user state recognition accuracy. Krause *et al.* [6] investigate the topic of trading off prediction accuracy and power consumption in mobile computing, and the authors showed that even very low sampling rate of accelerometer can lead to competitive classification results while device energy consumption can be significantly reduced.

In this paper, however, we study the energy management problem from a different perspective: we assume that each sensor sampling takes a unit of time to complete with a corresponding energy cost. Whenever the sensor makes an observation the user state will be discovered; on the other hand, when the sensor stays idle the user state is unknown and needs to be estimated using existing information. Instead of focusing on how a single sensor sample can be optimized like the work in [6], we study how the sensor can select its duty cycles in order to achieve the best tradeoff between the expected energy consumption and expected user state estimation error along the observation process.

Several previous works have investigated the problem of using Markov models to model user states and their transitions in the context of mobile networks and applications. For example, Bhattacharya and Das [4] use Markov chain to model user movement profile as he or she transits between cell towers, and the model could be updated as users move between cells or stay in a cell for a long period of time. Ashbrook *et al.* [2] incorporate user location traces into a Markov model that can be consulted for use with a variety of applications.

As introduced in the previous section, we model the background user state as a hidden discrete time Markov chain and as the chain evolves a sensor makes observations to detect user state in real-time. The Hidden Markov Model (HMM) technique is one of the most well-known techniques in the field of estimation and recognition [3, 8]. Both the Forward-Backward Algorithm and the Viterbi Algorithm are well known algorithms that derive the most likely underlying state sequence of HMM given an state observation sequence [11, 8]. Yu and Kobayashi [13] propose a new Forward-Backward algorithm with linear running time for Hidden Semi-Markov Model (HSMM) which can be applied to missing observations and multiple observation sequences. In this paper we simplify this process by assuming that sensor makes perfect observations, i.e., the observed state always equals the true user state. Recall that in order to achieve energy efficiency, the sensor would not be sampled in each time slot, and therefore the unknown state sequence only contains all the time slots without an associated sensor observation.

3 Model and Analysis

In this section, we present our mathematical model of sampling a DTMC based user state transition process. A stationary deterministic sensor sampling policy is proposed and two different mechanisms based on the Forward-Backward Algorithm and the Viterbi Algorithm are investigated which estimate user state sequence for all periods with missing observations, and finally derive the expression for expected sensor energy cost as well as expected state user estimation error, in terms of sensor duty cycle parameters and the DTMC transition probabilities.

3.1 Preliminaries

We assume that there are N user states, and the state transition follows a N-state discrete time Markov chain with transition probability $p_{ij}(i, j = 1, ...N)$ from state i to state j. We denote the discrete time line as $T = \{t : t = 0, 1, 2, ...\}$.

As the Markov chain evolves, a sensor is operated based on the following deterministic stationary sensing policy: If the sensor detects user state i at some time t, it will stay idle for I_i time slots (I_i belongs to the set of positive integers), and is re-sampled at time slot $t + I_i$. In other words, I_i is the length of the sensor idle period when state i is detected[1]. If the sensor makes an observation, there is a corresponding unit energy cost incurred in that time slot, whereas if the sensor stays idle the energy cost in that time slot is 0. We define O as the set of time slots when the sensor makes observation and O is thus a subset of T, as shown by figure 1. Consider the case where $I_1 = I_2 = ... = I_N = 1$, which indicates that the sensor performs state detection in every time slot. In this scenario $O = T$, and the user state can be correctly obtained in each time slot. However, due to energy consumption constraint, the sensor may have to stay idle sometimes in order to increase the device lifetime. As a result, there may exist time slots where no sensing is performed and the user state has to be estimated using available information, which is also illustrated by Figure 1. In particular, the unknown user state (state sequence) between two subsequent observations is going to be estimated using only the information from those two observations. We will present two mechanisms that estimate the missing state information between two subsequent observations in Section 3.3.

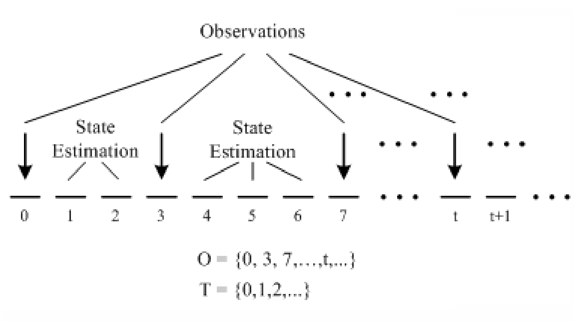

Fig. 1. Illustration of a discrete time Markov chain which is sampled at some time slots.

Our goal is to identify the tradeoff between the energy consumption and state detection (estimation) accuracy by quantifying both metrics as we vary the values of the idle periods I_i for each state i. The longer the sensor waits until making the next observation, the less energy the device consumes. However, a

[1] Note that the stationary deterministic sensor sampling policy studied in this paper is not claimed to be always optimal. In fact, a set of stationary randomized policies which select sensor duty cycle randomly may achieve higher state estimation accuracy than deterministic ones while preserving the same energy consumption constraint; however, in this paper we only focus on the analysis towards stationary deterministic policies and leave the study of randomized policies to future work.

longer waiting time also leads to more possibilities of state sequence taking place in that idle interval and thus more uncertainty about what "happened" between two observations.

Let s_t and s'_t denote the true user state and the estimated state at time slot t, respectively. The true state represents the user's real condition, whereas the estimated state is the "best guess" of user state which may be incorrect with certain probability. We assume that the sensor makes perfect observations, therefore if $t \in O$ we have $s'_t = s_t$ with probability 1.

We first derive the steady state probability of detecting a particular state i in an observation, under our deterministic sampling policy. We denote this probability as p_i. Since the state transition follows the Markovian rule, the current state observation result only depends on the previous observation status, thus:

$$p_i = P[observing\ state\ i] \tag{1}$$

$$= \sum_{j \in \{1,2,...,N\}} P[observing\ state\ j$$

$$I_j\ steps\ ago] \cdot P_{ji}^{(I_j)} \tag{2}$$

$$= \sum_j p_j \cdot P_{ji}^{(I_j)} \tag{3}$$

where $P_{ji}^{(I_j)}$ denotes the probability of state transition from j to i in exactly I_j time steps. Equation (3) accounts for all possible cases of last observation that may lead to state i in the current observation.

It is obvious to see that

$$\sum_i p_i = 1 \tag{4}$$

The values of $p_1, p_2, ...,$ and p_N can be obtained by solving the $N+1$ simultaneous equations in (3) and (4).

3.2 Expected sampling interval and expected energy cost

Given the probability of "seeing" each state when making an observation and the length of waiting time interval until next observation, the overall expected sampling interval $E[I]$ (idle time between two observations) can thus be expressed as:

$$E[I] = \sum_i p_i \cdot I_i \tag{5}$$

Let $E[C]$ be the overall expected energy cost whose value is defined as

$$E[C] = 1/E[I] \tag{6}$$

It can be seen that $E[C]$ characterizes the average per-slot energy cost. Note that since $I_i \geq 1$ and $\sum_i p_i = 1$, it is easy to conclude that $E[I] \geq 1$ and hence $E[C] \leq 1$. When $E[I] = E[C] = 1$, the Markov chain is observed in every time slot.

3.3 Estimation of missing state information and expected error

The problem of estimating missing state information between two subsequent observations is formulated as follows: given the fact that one observation detects state i at some time $t_m (t_m \in O)$ and the next observation detects state j at time $t_m + I_i (t_m + I_i \in O)$, what is the most likely state sequence that connects them?

In our study, we address this problem by two different methods:

3.3.1 Method 1: Pick the most likely state for each time slot and compute the expected per-slot error (utilizing the Forward-Backward Algorithm)

Given the sensor detects state i at time t_m, and detects state j at time $t_m + I_i$, the probability of being at state k at time t ($t_m < t < t_m + I_i$), which happens between the two observations, can be given by:

$$p[s_t = k | s_{t_m} = i, s_{t_m+I_i} = j] \tag{7}$$

$$= \frac{p[s_t = k, s_{t_m+I_i} = j | s_{t_m} = i]}{p[s_{t_m+I_i} = j | s_{t_m} = i]} \tag{8}$$

$$= \frac{p[s_t = k | s_{t_m} = i] \cdot p[s_{t_m+I_i} = j | s_t = k]}{p[s_{t_m+I_i} = j | s_{t_m} = i]} \tag{9}$$

$$= \frac{P_{ik}^{(t-t_m)} \cdot P_{kj}^{(t_m+I_i-t)}}{P_{ij}^{(I_i)}} \tag{10}$$

In order to obtain the "most likely" state at time t, the quantity given in equation 7 needs to be maximized. Thus, the estimated state s'_t can be chosen as:

$$s'_t = \underset{k}{\text{argmax}} \left\{ p[s_t = k | s_{t_m} = i, s_{t_m+I_i} = j] \right\} \tag{11}$$

$$= \underset{k}{\text{argmax}} \left\{ \frac{P_{ik}^{(t-t_m)} \cdot P_{kj}^{(t_m+I_i-t)}}{P_{ij}^{(I_i)}} \right\} \tag{12}$$

and the probability of correct estimation is given by:

$$p[s_t = s'_t | s_{t_m} = i, s_{t_m+I_i} = j] \tag{13}$$

$$= \max_k \left\{ p[s_t = k | s_{t_m} = i, s_{t_m+I_i} = j] \right\} \tag{14}$$

$$= \max_k \left\{ \frac{P_{ik}^{(t-t_m)} \cdot P_{kj}^{(t_m+I_i-t)}}{P_{ij}^{(I_i)}} \right\} \tag{15}$$

Having chosen the mostly likely state at time t, the corresponding expected estimation error for time slot t, denoted by e_t, is simply given by the probability that the true state is different from the estimation result:

$$e_t = 1 - p[s_t = s'_t | s_{t_m} = i, s_{t_m+I_i} = j] \tag{16}$$

$$= 1 - \max_k \left\{ \frac{P_{ik}^{(t-t_m)} \cdot P_{kj}^{(t_m+I_i-t)}}{P_{ij}^{(I_i)}} \right\} \tag{17}$$

Let e_{ij} be the aggregated expected estimation error between two consecutive samples that observe state i and j respectively. This e_{ij} accounts for the aggregated estimation error for all time slots between the two observations. Again, suppose the first observation is made at time t_m, e_{ij} can be expressed as

$$e_{ij} = \sum_{t=t_m+1}^{t_m+I_i-1} e_t \tag{18}$$

Now we derive the overall expected per-slot estimation error, denoted by $e_{perslot}$, which can be calculated by dividing the total number of time slots where incorrect estimation is made by the total number of time slots, i.e.:

$$e_{perslot} = \lim_{t\to\infty} \frac{\sum_{k=1}^{o(t)} W_k}{\sum_{k=1}^{o(t)} N_k} \tag{19}$$

$$= \lim_{t\to\infty} \frac{\frac{1}{o(t)} \cdot \sum_{k=1}^{o(t)} W_k}{\frac{1}{o(t)} \cdot \sum_{k=1}^{o(t)} N_k} \tag{20}$$

where $o(t)$, W_k, and N_k denote the total number of observation intervals till time t, the number of incorrect estimations in the k^{th} observation interval, and the number of time slots in the k^{th} observation interval, respectively. Considering the Law of Large Numbers, equation (20) can be further written as

$$e_{perslot} = \frac{E[W]}{E[N]} \tag{21}$$

which indicates that the overall expected per-slot estimation error is given by the ratio of expected number of estimation errors per observation interval and expected observation interval length. Considering all possible cases of a pair of subsequent observations and their corresponding aggregated estimation error, the overall expected per-slot estimation error is thus given by:

$$e_{perslot} = \frac{\sum_{ij}(e_{ij} \cdot p_i \cdot P_{ij}^{(I_i)})}{\sum_{ij}(p_i \cdot P_{ij}^{(I_i)} \cdot I_i)} \tag{22}$$

$$= \frac{\sum_{ij}(e_{ij} \cdot p_i \cdot P_{ij}^{(I_i)})}{\sum_i p_i \cdot I_i} \tag{23}$$

$$= \frac{\sum_{ij}(e_{ij} \cdot p_i \cdot P_{ij}^{(I_i)})}{E[I]} \tag{24}$$

in which the component $p_i \cdot P_{ij}^{(I_i)}$ accounts for the probability that observing state j in I_i time slots after observing state i.

3.3.2 Method 2: Estimate the most likely state sequence and compute the expected sequence estimation error (utilizing the Viterbi Algorithm)

Instead of selecting the most likely state for individual time slots, *Method 2* returns the most likely state sequence between two subsequent observations. We first construct a trellis which contains all possible paths from state i to state j in I_i time steps. Without loss of generality, again, let t_m and $t_m + I_i$ be the first and second observation time slots. Define the quantity

$$q_k^t = \max_{s_{t_m}=i, s_{t_m+1}, ..., s_{t-1}} P[s_{t_m} = i, s_{t_m+1}, ..., s_{t-1}, s_t = k] \tag{25}$$

where $k \in \{1, ..., N\}$ and $t_m < t$, i.e., q_k^t is the highest probability along a single path starting from state i at time t_m, and ends at state k at time t, which accounts for the first t observations after t_m.

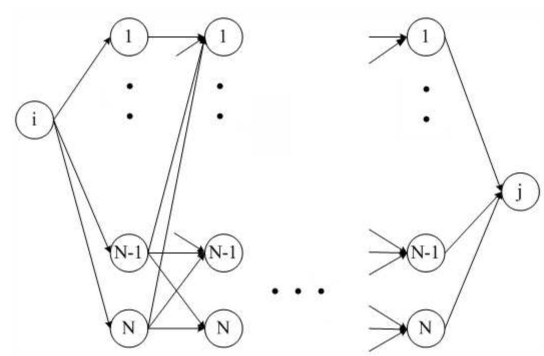

Fig. 2. All possible paths starting from state i at time 0 and ending at state j at time I_i.

In order to retrieve the most likely state sequence, we need to keep track of all the states that maximize the quantity given by equation (25), for each t and i. Thus, the complete process for finding the mostly likely state sequence and expected sequence error probability between two observations can be listed as follows:

– Initialization:

$$q_i^{t_m} = 1. \tag{26}$$

– Induction and state estimation:

$$q_k^t = \max_m (q_m^{t-1} \cdot p_{mk}), t_m < t \leq t_m + I_i - 1 \tag{27}$$

$$s_t' = \operatorname*{argmax}_k (q_k^t), t_m < t \leq t_m + I_i - 1 \tag{28}$$

– Construct the most likely estimation sequence:

$$S' = \{s'_{t_m+1}, s'_{t_m+2}, ..., s'_{t_m+I_i-1}\} \tag{29}$$

– Compute expected sequence error e'_{ij}:

$$e'_{ij} = 1 - \max_k (q_k^{t_m+I_i-1}) \tag{30}$$

It should be noted that q_k^t denotes a "path probability" and this method computes the most likely state sequence instead of estimating the most possible state for each individual time slots. The estimated sequence is considered to be incorrect even if it contains a single bit error. Therefore the expected sequence error probability can be computed based on equation (30), which is the complement of the maximum probability over all possible paths starting at state i at time t_m and ending at time $t_m + I_i - 1$. Finally, recall the fact that estimation may be performed within all possible subsequent observation pairs; hence the overall expected sequence error e_{seq}, which is the average probability that a sequence estimation is in error, can be thus expressed as:

$$e_{seq} = \sum_{ij} (e'_{ij} \cdot p_i \cdot P_{ij}^{(I_i)}) \tag{31}$$

It is worth noting that the expected energy consumption $E[C]$, the expected per-slot error $e_{perslot}$ (computed in *Method 1*), and the expected sequence estimation error e_{seq} (computed in *Method 2*) all depend on the choices of I_i ($i \in \{1, ..., N\}$) as well as the Markov chain transition probabilities $p_{ij}(i, j = 1, ...N)$.

3.3.3 Discussion of the Two Methods

As can be seen, the key difference between the two methods lies in the fact that *Method 1* requires two subsequent observation results and estimate user state for each time slot and finally construct the state sequence, whereas *Method 2* only requires a single observation and returns the most likely state sequence directly. The choice of algorithm would depend on which metric is most important to a given application. It may also be possible to consider the combination of the two objectives, but we have not studied this in our current work.

4 A Case Study: Two-state DTMC

In order to visualize the tradeoff between energy consumption and estimation error, we have conducted a case study where for simplicity, we consider a two-state discrete time Markov chain with states "1" and "2" and transition probability matrix

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \tag{32}$$

According to the deterministic sampling policy, if the sensor detects state "1", it will be re-sampled in I_1 time slots, and if the sensor detects state "2" it is re-sampled in I_2 time slots. As the potential possibilities of transition matrix is infinite, we select six representative ones, namely, $\begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$, $\begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}$, $\begin{bmatrix} 0.9 & 0.1 \\ 0.9 & 0.1 \end{bmatrix}$, $\begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$, $\begin{bmatrix} 0.5 & 0.5 \\ 0.9 & 0.1 \end{bmatrix}$, and $\begin{bmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{bmatrix}$. Recall that our goal is to identify the effect of selecting different sampling intervals on the performance including energy consumption and state estimation error. We examine all possible combinations of I_1 and I_2 with constraints $1 \leq I_1 \leq 30$ and $1 \leq I_2 \leq 30$.

4.1 Analytical results

The corresponding energy cost and expected errors are computed based on equation (6), (24), and (31). figure 3 shows the relationship between the expected per-slot estimation error $e_{perslot}$ and energy $E[C]$ based on the *Method 1*, whereas figure 4 shows the incorrect sequence estimation probability e_{seq} with respect to energy cost $E[C]$, computed based on *Method 2*. Note that these figures present a scatter plot of the tradeoffs obtained for all values of I_1 and I_2 considered. Of most interest is the lower envelope of these figures, which represent the optimal tradeoff between energy consumption and state estimation error.

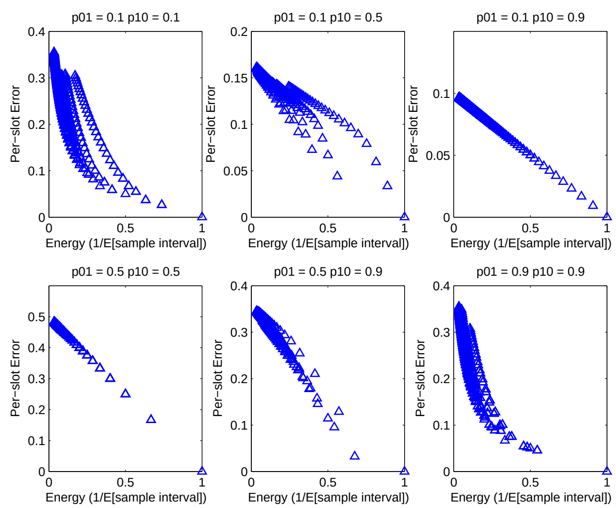

Fig. 3. Analysis result of expected per-slot estimation error with respect to energy usage based on *Method 1*.

It can be seen from figure 3 that the average per-slot state estimation error is within the range $[0, 1 - \max\{\pi_1, \pi_2\}]$ (π_1 and π_2 are the steady state probability

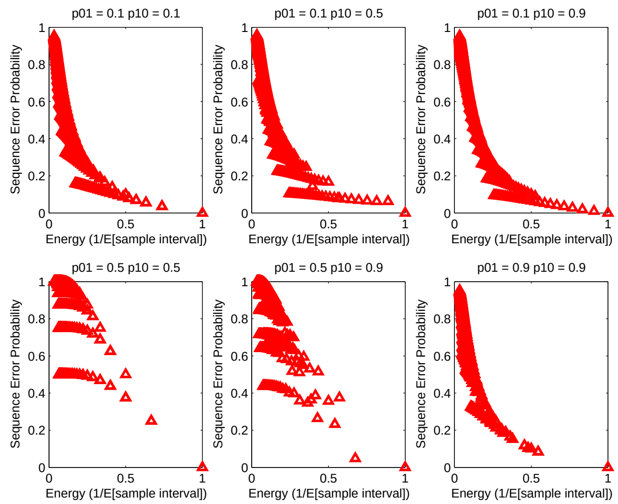

Fig. 4. Analysis result of expected sequence estimation error with respect to energy usage based on *Method 2*.

for state 1 and state 2 respectively) as the expected energy consumption changes within the range $[0, 1]$. This result is quite intuitive because when $E[C] = 1$, the sensor makes observation in each time slot therefore no estimation is needed and the observed user state sequence matches the true state sequence perfectly. On the other hand, if the sensor does not make any observation where $E[C] = 0$, the estimated state in each time slot is always going to be the one with larger steady state probability. Therefore the upper bound of expected per-slot state estimation error can be expressed as $1 - \max\{\pi_1, \pi_2\}$. For N-state DTMC, the expected per-slot state estimation error can be generalized as:

$$0 \leq e_{perslot} \leq 1 - \max\{\pi_1, \pi_2, ..., \pi_N\} \tag{33}$$

A similar explanation from above can be applied to figure 4, where the expected sequence estimation error $e_{seq} = 0$ whenever $E = 1$. However it can be seen that $e_{seq} = 1$ if no sensor observation is make. This is because in *Method 2* we examine the expected sequence error probability, and an estimated sequence is considered incorrect as long as it contains a single bit error.

It can be easily concluded from figure 3 that the energy consumption have a large impact on user state estimation accuracy. The results verify the intuition that the more energy utilized, the more accurate state estimation result will be, and vice versa. It can be also found out that given different user state transition probability matrices, the tradeoff curves between energy and error appear to have different characteristics. For all the tradeoff curves, it may be desirable to keep the system running at the knee of the curve which provides the best tradeoff between the two metrics. For example, if $p_{12} = p_{21} = 0.1$, if 10% error is allowed, the system energy consumption can be reduced by as much as 80% compared to a fully operating device.

4.2 Simulation results

In order to verify the analytical results presented in Section 4.1, we have conducted simulations on a 2-state discrete time Markov chain with same settings as discussed in Section 4.1. For each I_1 and I_2 combination, a duration of 5000 time slots is simulated. At the beginning of the simulation run the initial state of the Markov chain is arbitrarily set to "1" and the first sensor observation takes place at time slot 0. State estimation is performed right after a single observation is made and all the missing state information from last observation point are estimated using both methods. The system records the estimated state sequence as well as all the observation points in order to evaluate both the state estimation error and energy cost.

The expected energy consumption is measured by calculating the ratio between the total number of observations made and the length of simulation time (5000 time slots). For *Method 1*, we have found out the expected per-slot estimation error by dividing the number of incorrect estimations (as represented by the number of time slots where the estimation result does not match the true state) by the total number of time slots where the state needs to be estimated (i.e.: observation is missing). The results can be found in figure 5.

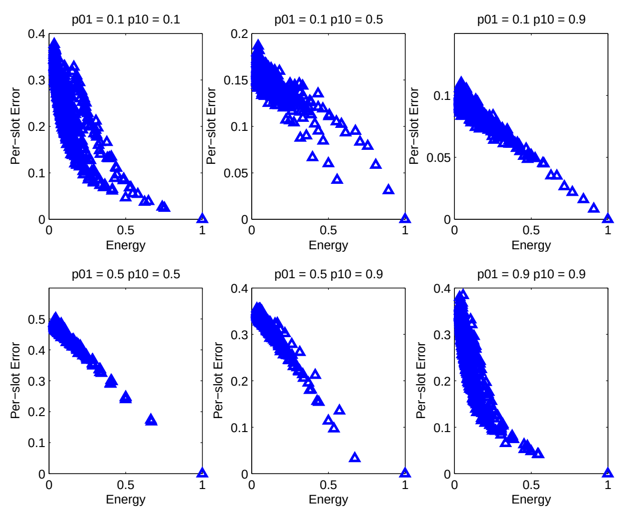

Fig. 5. Simulation result of per-slot error versus energy cost based on *Method 1*.

For *Method 2*, we examine the percentage of sequence estimations that are incorrect. Recall that the estimated state sequence is considered incorrect even if a single bit of estimation does not match the ground truth. The plot of percentage of incorrect sequence estimations with respect to energy cost can be found in figure 6. Moreover, since the system is able to record the estimated state sequence which can be further compared to the ground truth, we have also identified the per-slot error based on *Method 2*, and the results can be found in figure 7. Interestingly, we find that the per-slot error of *Method 2* appears to be comparable to that obtained by *Method 1*.

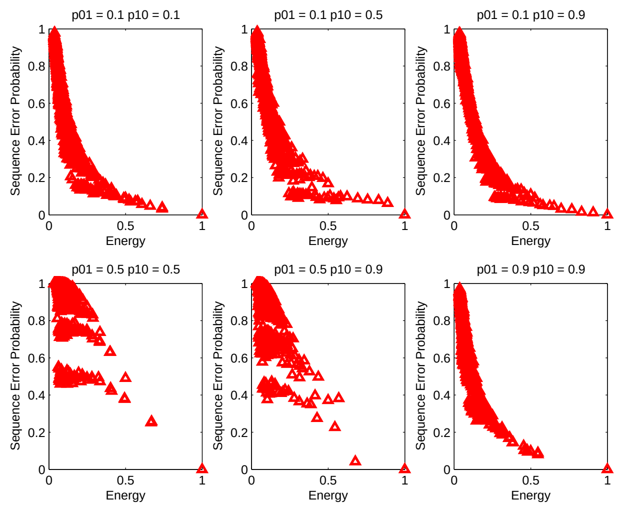

Fig. 6. Simulation result of expected sequence estimation error versus energy cost based on *Method 2*.

It can be seen that the energy-error tradeoff plot in figure 5 and 6 demonstrate a very close match to the numerical analysis result shown in figure 3 and 4, and this validates our analytical process and derivations of expected energy and state estimation error in section 3.

4.3 Optimal sensor idle intervals for different energy budget

In practical system designs, a device energy consumption budget is usually specified in order to ensure long enough device operating duration. We define ξ (where $0 \leq \xi \leq 1$) as the energy budget which is the maximum average energy cost allowed. We investigate the following problem: given an energy budget ξ, and user state transition probability matrix P, what is the optimal length of sensor idle interval at each state such that the state estimation error can be minimized?

All integer combinations of I_1 and I_2 have been tested and the one that leads to the minimum estimation error while satisfying the energy constraint is picked, and figure 8 shows the optimal I_1 and I_2 combinations to utilize in order to achieve the least per-slot estimation error, given a particular energy budget,

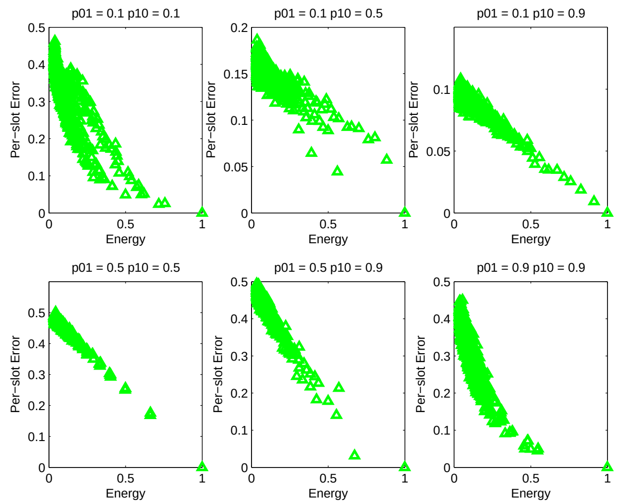

Fig. 7. Simulation result of per-slot error versus energy cost based on *Method 2*.

i.e., the sensor idle intervals must be such that the average energy cost is less than the budget value [2].

From figure 8 it can be seen that for symmetric user state transitions, i.e., when $p_{12} = p_{21}$, the optimal I_1, I_2 values are close to each other. This is due to the fact that the average duration spent in each state are the same. Therefore, the sensor should be assigned same duty cycles in each state as well. An interesting observation is the following: when $p_{12} = 0.1$ and $p_{21} = 0.5$, sampling the sensor more frequently when the user is at state 2 leads to the minimum state estimation error. Intuitively, if the average duration spent in a state is low (in this case, state 2), we prefer to sample the sensor more frequently in order to capture state transition quickly. However, an inverse observation is made for both cases when $p_{12} = 0.1, p_{21} = 0.9$ and $p_{12} = 0.5, p_{21} = 0.9$. In these cases sampling more often in user state 1 leads to the minimum state estimation error. This illustrates that the optimal sensing policy can be quite sensitive to the transition probabilities.

5 Conclusion and Future Work Directions

Mobile device based sensing of human user states is able to provide rich contextual information to applications such as social networking and health monitoring. However, energy management remains a critical problem due to limited battery capacity on mobile devices.

[2] A more general approach is to model the optimal sensor idle interval selection with energy constraint problem as a Constrained Markov Decision Process (CMDP) such that an optimal sensor sampling policy can be deduced. We leave this to future work.

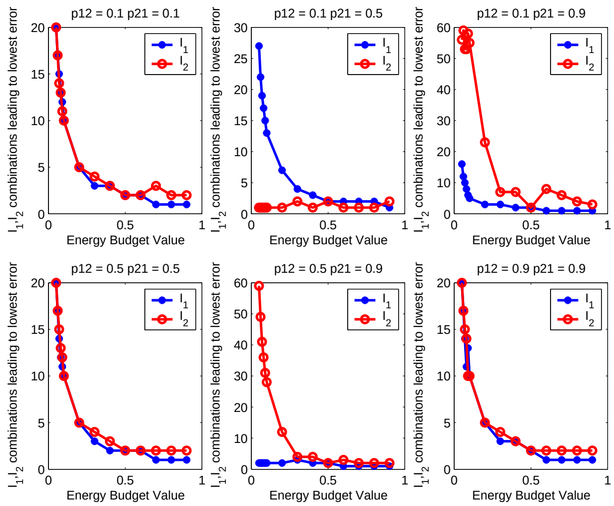

Fig. 8. The best I_1, I_2 combinations that lead to the lowest error given a particular energy budget.

In this paper we have studied the effect of applying different sensor duty cycles on the expected energy consumption as well as the expected user state estimation error, and have identified the tradeoff between the two metrics. In particular, we have modeled the user state as a hidden DTMC, and have proposed a stationary deterministic sensor sampling policy which assigns different sensor duty cycles for different user states. We introduced two state estimation methods in order to compute the most likely state sequence whenever sensors are idle. The tradeoff between the expected energy consumption and the expected user state estimation error have been visualized numerically.

In future work, we plan on modeling and studying the sensor duty cycle selection with energy constraint problem as a CMDP with stationary randomized sensor sampling policies and comparing them to our current stationary deterministic policies. We also plan to apply our sensor sampling policy and state estimation algorithms to real user data and identify the relationship between energy consumption and state estimation error.

References

1. M. Annavaram, N. Medvidovic, U. Mitra, S. Narayanan, D. Spruijt-Metz, G. Sukhatme, Z. Meng, S. Qiu, R. Kumar, and G. Thatte. Multimodal sensing for pediatric obesity applications. In *UrbanSense08 Workshop at SenSys*, Raleigh, NC, USA, November 2008.
2. D. Ashbrook and T. Starner. Learning significant locations and predicting user movement with GPS. In *IEEE International Symposium on Wearable Computers*, 2002.

3. L. R. Bahl, F. Jelinek, and R. L. Mercer. A maximum likelihood approach to continuous speech recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1983.
4. A. Bhattacharya and S. K. Das. Lezi-update: An information-theoretic approach to track mobile users in PCS networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, 1999.
5. S. Biswas and M. Quwaider. Body posture identification using hidden markov model with wearable sensor networks. In *BodyNets Workshop*, Tempe, AZ, USA, March 2008.
6. A. Krause, M. Ihmig, E. Rankin, S. Gupta, D. Leong, D. P. Siewiorek, A. Smailagic, M. Deisher, and U. Sengupta. Trading off prediction accuracy and power consumption for context-aware wearable computing. In *IEEE International Symposium on Wearable Computers*, 2005.
7. J. Lester, T. choudhury, G. Borriello, S. Consolvo, J. Landay, K. Everitt, and I. Smith. Sensing and modeling activities to support physical fitness. In *Proceedings of UbiComp*, Tokyo, Japan, 2005.
8. L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, 1989.
9. E. Shih, P. Bahl, and M. J. Sinclair. Wake on wireless: an event driven energy saving strategy for battery operated devices. In *Proceedings of MobiCom*, Atlanta, Georgia, USA, 2002.
10. Twitter. http://www.twitter.com.
11. A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. In *IEEE Transactions on Information Theory*, 1967.
12. Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. krishnamachari, and N. Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *Proceedings of MobiSys*, Krakow, Poland, June 2009.
13. S. Yu and H. Kobayashi. A hidden semi-markov model with missing data and multiple observation sequences for mobility tracking. In *Signal Processing*, 2003.