Energy-Accuracy Trade-offs in Querying Sensor Data for Continuous Sensing Mobile Systems

Kiran K. Rachuri
Computer Laboratory
University of Cambridge
kkr27@cam.ac.uk

Mirco Musolesi
School of Computer Science
University of St. Andrews
mirco@cs.st-andrews.ac.uk

Cecilia Mascolo
Computer Laboratory
University of Cambridge
cecilia.mascolo@cl.cam.ac.uk

ABSTRACT

A large number of context-inference applications run on off-the-shelf smart-phones and infer context from the data acquired by means of the sensors embedded in these devices. The use of efficient and effective sampling technique is of key importance for these applications. Aggressive sampling can ensure a more fine-grained and accurate reconstruction of context information but, at the same time, continuous querying of sensor data might lead to rapid battery depletion.

In this paper, we present an adaptive sensor sampling methodology which relies on dynamic selection of sampling functions depending on history of context events. We also report on the experimental evaluation of a set of functions that control the rate at which the data are sensed from the Bluetooth device, accelerometer, and microphone sensors and we show that a dynamic adaptation mechanism provides a better trade-offs compared to simpler function based rate control methods. Furthermore, we show that the suitability of these mechanisms varies for each of the sensors, and the accuracy and energy consumption values stabilize after reaching a certain level.

INTRODUCTION

The development of mobile context-aware applications has always been limited by energy, processing, and memory constraints. With the advent of high-end phones like Google Nexus One, and HDC HD2, the processing and memory limitations are overcome to a great extent - indeed, these mobile phones are equipped with a 1 GHz processor and 512 MB RAM. However, energy is still a scarce resource and should be expended judiciously by the applications. For example, it is reported in [7] that the battery charge of a Nokia N95 smart phone lasts less than 5 hours when sensing data from GPS, accelerometer, and microphone sensors using a predefined static (and aggressive) sampling rate. In other words, context-aware applications are by definition resource intensive, since they continuously query data from sensors. There is a need for adaptive mechanisms for querying the sensor data in an energy efficient way by considering the application requirements in terms of energy and accuracy, and provide the sensor data to these applications.

As an initial step towards a larger framework, in this paper, we present a design methodology to evaluate and study the energy-accuracy trade-offs of rate control mechanisms for querying sensor data in continuous sensing mobile systems. These aspects are in some respects orthogonal to the problem of intelligent mechanisms for uploading data to a back-end [3], even if the sampling rate can also be tuned according to the corresponding transmission rate. However, some applications perform local computation on the phones and only then transmit the information to a remote server via GPRS or WiFi.

We present some preliminary results by evaluating a set of functions that control the rate at which the data should be sensed from the Bluetooth, accelerometer, and microphone sensors for a predefined set of classifiers used in various mobile context-aware applications. In this work, we focus on context events that are represented by a stream of states, such as streams of user activities, like walking, sitting, in conversation, and so on. We do not consider context information that is measured continuously in a given range, such as temperature. The contributions of this work can be summarized as follows:

- We propose a methodology for studying the energy-accuracy trade-offs for querying data in continuous sensing applications using a set of sampling functions selected dynamically according to the stream of context events. In particular, we discuss the choice of parameters of a dynamic adaptation algorithm that switches among a set of sampling functions based on the analysis of the stream of past events.
- We show experimentally that a dynamic adaptation mechanism provides a better trade-offs compared to simpler function based rate control methods. Furthermore, we show that the suitability of these mechanisms varies for each of the sensors, and the accuracy and energy consumption values stabilize after reaching a certain level.

UbiComp '10, Sep 26–29, 2010, Copenhagen, Denmark.

Table 1. Advance and back-off functions

Type	Back-off function	Advance function
Linear	$k \times x$	x/k
Quadratic	x^2	$\sqrt{x}$
Exponential	e^x	$\log_e x$
Minimum	N/A	*minSamplingInterval*
Maximum	*maxSamplingInterval*	N/A

ADAPTIVE SAMPLING BASED ON DYNAMIC FUNCTIONS

In order to address the energy-accuracy trade-offs of context-aware applications, we propose a methodology that uses a set of functions to adjust the sampling rate of sensors based on the current observed data. The sensor data are either queried periodically or aperiodically according to the sampling function used. We define two parameters *viz.*, *minSamplingInterval* and *maxSamplingInterval*. The former is the minimum sleep interval between two successive sensor readings and the later is the maximum sleep interval. If the sensor sampling interval for a sensor is always set to *minSamplingInterval*, then the accuracy of classifiers will be high (due to aggressive data sampling). However, the energy expended will also be considerable. On the other hand, if the sampling interval is always set to *maxSamplingInterval*, then the energy consumption is minimized but the accuracy decreases.

We classify context events into two classes. An *unmissable* event is an event of interest observed in the environment that should not be missed by the sensor. A *missable* event indicates that no interesting external phenomenon has happened and the corresponding sensor can sleep during this time. If there are no "interesting" events observed (*i.e.*, missable events), then the sampling interval increases from its current value to *maxSamplingInterval* based on a *back-off function*. Similarly, if the event is classified as unmissable, then the sampling interval decreases from its current value to *minSamplingInterval* based on an *advance function*. The classification of an event as missable or unmissable is application dependent. The choice of the advance and back-off functions and of the *minSamplingInterval* and *maxSamplingInterval* parameters play a crucial rule in the energy-accuracy trade-offs for the various context inference components. Examples of back-off and advance functions (also used in the evaluation section) are given in Table 1.

The functions take into consideration the previous state and not the full or partial history of the context events. Moreover, they can be considered static (no dynamic adaptation). One further step is to dynamically switch these functions based on past observations of sensor data. For our evaluation, we adopt the adaptive technique showed in Algorithm 1. The idea is to use the functions according to the consistency of the observed sensor data, *i.e.*, the function changes from least to most "aggressive" based on the number of consecutive sampling of the same state. By adopting this mechanism, small state changes do not have a large effect on the sampling interval. More refined techniques can be implemented. However, since the goal of this work is primarily to present a methodology for tuning the parameters of adaptive sampling functions by analyzing energy-accuracy trade-offs, we limit our discussion, methodological analysis, and performance evaluation to this simple mechanism.

Algorithm 1 Dynamic adaptation algorithm

```
sleep(sleepInterval)
interestingEvent = senseAndClassify(sensorId)
if (interestingEvent == TRUE) then
    uninterestingSequence = 0
    sequence = interestingSequence++
else
    interestingSequence = 0
    sequence = uninterestingSequence++
end if
if (sequence < linearThreshold) then
    function = linear
else if (sequence < quadraticThreshold) then
    function = quadratic
else
    function = exponential
end if
sleepInterval = update(function, sleepInterval, interestingEvent)
if (sleepInterval ≥ maxSamplingInterval) then
    sleepInterval = maxSamplingInterval
else if (sleepInterval ≤ minSamplingInterval) then
    sleepInterval = minSamplingInterval
end if
```

EVALUATION

In this section we describe the dataset used for the evaluation and then present the results of the performance evaluation of the proposed techniques considering Bluetooth, accelerometer, and microphone sensors.

Dataset

Trace files with ground-truth information for accelerometer, Bluetooth, and microphone sensors were collected from 10 users for 24 hours using the EmotionSense platform [5] running on Nokia 6210 phones. In order to extract the microphone sensor traces, audio samples of 5 seconds length were recorded continuously with a sleep period of 1 second between consecutive recordings. Co-location data for the Bluetooth sensor traces were queried continuously with a sleep duration of 3 seconds between successive queries. The accelerometer sensor was sampled continuously for movement information with an interval of 1 second.

As discussed above, the events generated from the data of each sensor can be of two types, *viz.*, "unmissable" and "missable" events. In the case of the microphone sensor, an unmissable event corresponds to some audible voice data being heard in the environment and a missable event corresponds to silence. These events are generated based on a GMM classifier [1] capable of classifying whether an audio trace contains any conversation. For the Bluetooth sensor traces, an unmissable event corresponds to a change in the number of co-located users, whereas a missable event indicates that there is no change. We assume reliable Bluetooth readings, however, techniques to identify outliers can also

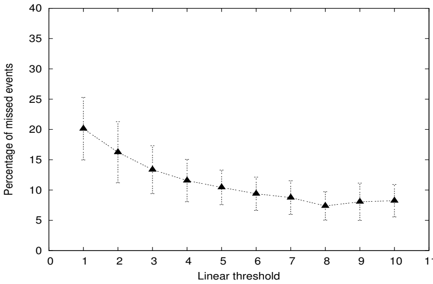

Figure 1. Percentage of missed events vs linear threshold for Bluetooth sensor.

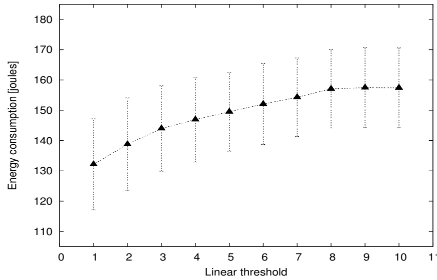

Figure 2. Energy consumption vs linear threshold for Bluetooth sensor.

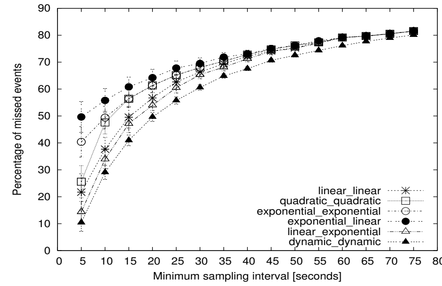

Figure 3. Percentage of missed events vs minimum sampling interval for Bluetooth sensor.

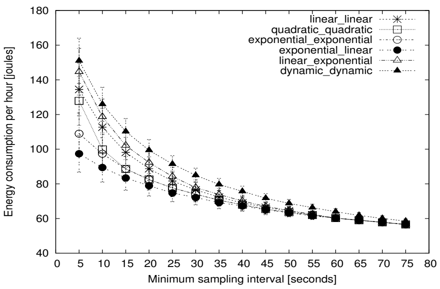

Figure 4. Energy consumption vs minimum sampling interval for Bluetooth sensor.

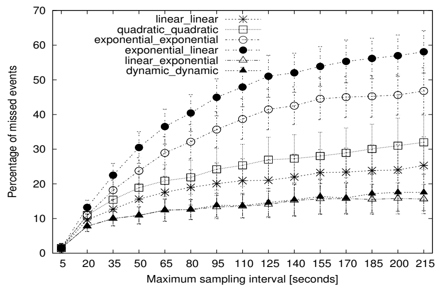

Figure 5. Percentage of missed events vs maximum sampling interval for Bluetooth sensor.

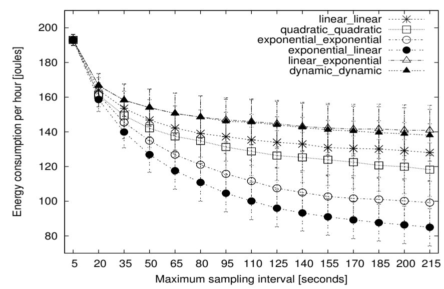

Figure 6. Energy consumption vs maximum sampling interval for Bluetooth sensor.

be applied. In the case of the accelerometer sensor, the unmissable event corresponds to movement of a user and a missable event indicates that the user is stationary. Although both of these events are unmissable, it is sufficient to detect just one of them since we have just two possible events, so we choose "user moving event" as unmissable. In future, we would like to consider unmissable events as transition events, *i.e.*, movement to stationary and vice versa, as it is more efficient. The accuracy is measured in terms of the percentage of missed events. An event is said to be missed when there is an unmissable event recorded in the trace file while the sensor is not actively queried. The energy consumption is measured using the Nokia Energy Profiler.

Results

In order to find optimal values of *linearThreshold* and *quadraticThreshold* for the Bluetooth sensor, we varied one of them by fixing the other. Figures 1 and 2 show the accuracy and energy consumption by varying the *linearThreshold* value. From these results, we selected a value of 3 for *linearThreshold* as the benefits in terms of accuracy after that are not high. We present all the results for Bluetooth sensor, but due to space constraints, we only show the variation of minimum sampling interval for accelerometer and microphone sensors. Note that the format of the legend in these plots is <*advance function*> _ <*backoff function*>. We can observe that all these curves stabilize at certain values. Therefore, high values of these intervals do not necessarily imply low accuracy and high savings in energy. It should suffice to use the values after which there are no considerable improvements in terms of performance. Dynamic adaption function is best in terms of accuracy compared to the other functions for most of the cases; however, it is not always the worst in terms of energy consumption (see Figure 8). With respect to the Bluetooth sensor (Figures 3 and 4), for a *minSamplingInterval* value of 5, the best performing function (dynamic adaptation) is more accurate than the worst (exponential_linear) by a factor of 5, whereas, in terms of energy consumption the gain ratio is 1.5. So, in this case, the gain in accuracy is much higher than the compromise in terms of energy consumption using the dynamic adaptation method. Figures 5 and 6 show the *maxSamplingInterval* variation for Bluetooth sensor.

With respect to the accelerometer sensor (Figures 7 and 8), the difference between the functions in terms of energy consumption is negligible, whereas the difference with respect to accuracy is not. For a *minSamplingInterval* value of 3, the best performing function is 20% more accurate than the worst, whereas the difference in terms of energy consumption is only 1%. Moreover, for the microphone sensor (Figures 9 and 10), we can observe that for a *minSamplingInterval* value of 25, the accuracy of *linear_exponential* is only 3% less than that of dynamic adaption method, however, the energy saving of the former is 11% better than the latter. This is due to high energy consumption for processing audio data locally on the phone. Therefore, *linear_exponential* is a better option for this sensor.

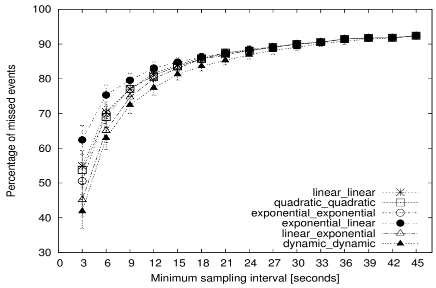

Figure 7. Percentage of missed events vs minimum sampling interval for accelerometer sensor.

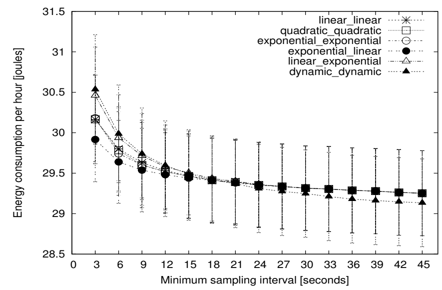

Figure 8. Energy consumption vs minimum sampling interval for accelerometer sensor.

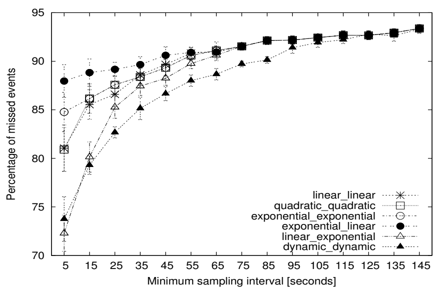

Figure 9. Percentage of missed events vs minimum sampling interval for microphone sensor.

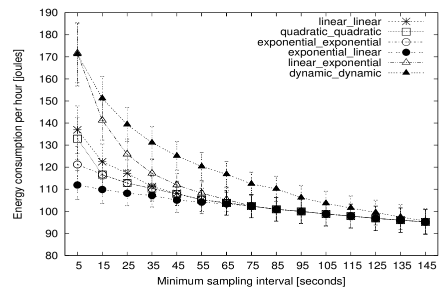

Figure 10. Energy consumption vs minimum sampling interval for microphone sensor.

RELATED WORK

Energy efficiency is a key issue in mobile sensing system design, and for this reason, it has been investigated in many recent works. In this section, we present a brief selection of relevant projects. The *EEMSS system* [7] is probably the most relevant work: this platform uses a hierarchical sensor management strategy to recognize user states as well as to detect state transitions. *SeeMon* [2] is a context monitoring service for mobile devices based on several sensors, and it achieves energy efficiency by performing context recognition only when the there is a change in the context. In [3] the authors show that continuous sensing is a viable option for mobile phones by adopting efficient data uploading (to a remote server) strategies. In [4] the authors address the problem of energy-delay trade-offs in smart phone applications. Finally, several energy saving schemes for mobile devices are discussed and compared in [6].

CONCLUSIONS

In this paper, we presented a design methodology to evaluate energy-accuracy trade-offs for querying sensor data in continuous sensing mobile systems, and also presented its evaluation with respect to a set of functions that control the rate at which the data should be sensed from the Bluetooth, accelerometer, and microphone sensors. We also presented a dynamic algorithm that switches among these functions based on the context history. The results show that the dynamic adaptation scheme is better in terms of accuracy, however, the suitability of these functions varies for each of the sensors.

We recently built a system [5] for sensing and analyzing user speech patterns and human emotions. We plan to integrate the current function based sampling into that system. Our future research agenda includes the design of techniques for intelligent sampling *and* uploading to back-end servers for further processing, *i.e.*, mechanisms that are able to optimize the sampling and uploading processes at the same time.

ACKNOWLEDGMENTS

This work was supported through Gates Cambridge Trust at the University of Cambridge, and EPSRC grants EP/C544773, EP/F033176, and EP/D077273.

REFERENCES

1. C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2007.
2. S. Kang, J. Lee, H. Jang, H. Lee, Y. Lee, S. Park, T. Park, and J. Song. SeeMon: Scalable and Energy-efficient Context Monitoring Framework for Sensor-rich Mobile Environments. In *Proceedings of MobiSys '08*, 2008.
3. M. Musolesi, M. Piraccini, K. Fodor, A. Corradi, and A. T. Campbell. Supporting Energy-Efficient Uploading Strategies for Continuous Sensing Applications on Mobile Phones. In *Proceedings of Pervasive '10*, 2010.
4. M.-R. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, and M. J. Neely. Energy-Delay Tradeoffs in Smartphone Applications. In *Proceedings of MobiSys '10*, 2010.
5. K. K. Rachuri, M. Musolesi, C. Mascolo, P. J. Rentfrow, C. Longworth, and A. Aucinas. EmotionSense: A Mobile Phones based Adaptive Platform for Experimental Social Psychology Research. In *Proceedings of UbiComp '10*, 2010.
6. M. A. Viredaz, L. S. Brakmo, and W. R. Hamburgen. Energy Management on Handheld Devices. *ACM Queue*, 1(7):44–52, 2003.
7. 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 '09*, 2009.