Selective Sampling Strategies to Conserve Power in Context Aware Devices

Brian French¹, Daniel P. Siewiorek², Asim Smailagic², Michael Deisher³

¹School of Computer Science, Carnegie Mellon University, Pittsburgh
²Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh
³Intel Emerging Platforms Lab, Hillsboro, OR
bfrench@cmu.edu, {dps, asim}@cs.cmu.edu, michael.deisher@intel.com

Abstract

We analyze the use of selective sampling strategies to aid in power conservation in sensor platforms for context-aware systems. In particular, we study an activity-aware system based on the eWatch sensor and notification platform, developed at CMU. We collected 94 hours of self-annotated activity data from four subjects over several days each.

We compare sampling strategies according to several metrics, each of which satisfies a different set of application needs. These metrics include: accuracy as the percentage of time between samples that sampled activity matches true activity, average latency of detecting a change in activity, the percentage of missed activities, and the percentage of redundant samples. We consider both the performance differences between strategies as well as differences between subjects. Accuracies of over 95% were achievable using only 3% of the samples.

1. Introduction

Being able to identify a person's activity provides a high level of information about the state of the person, which can be exploited when constructing a context-aware system. Our model for a context-aware system is the eWatch, a multi-sensor platform developed at CMU [1].

Body-worn accelerometers have been used to recognize different activities [2][3]. However, the power resources of mobile platforms are limited, making the demands of continuous, on-line classification untenable.

In [4], the authors present a low-power sensor platform. They evaluated power requirements between different sampling rates of sensors and features for the classification task with different computational complexities. They do not consider selective sampling strategies and the only metric they consider in the trade-off is recognition accuracy.

In [5], the authors explore the trade-off between recognition accuracy and power consumption using a classifier based on the eWatch sensor platform. They explore reduction of sensor sampling rates, computational complexity of classification features and some selective sampling strategies. However, their selective sampling strategies are only tested on an extremely limited (4 subjects with one hour of data from each subject), artificial dataset (each subject was given a list of activities to perform, which they spaced randomly throughout an hour long session) and again recognition accuracy is the only metric used for comparison.

In Section 2, we discuss the trade-off between power consumption in more detail. In Section 3, we discuss our user study for collecting data. In Section 4, we discuss selective sampling experiments and the metrics used to compare them. In Section 5, we provide the results of the selective sampling experiments. Finally, in Section 6, we provide some conclusions and outline future work.

2. The power consumption trade-off

Continuous sensing and classification can quickly exhaust the battery. For example, continuous sampling of eWatch sensors at a rate of 20 Hz results in a battery life of under seven hours, and this does not even consider energy needed to perform classification.

As was discussed in [4] and [5], significant gains can be made in power consumption of a wearable activity recognition system simply through efficient choices of sensor sampling rates and feature selection. In [5], it was shown that eWatch's sampling rate could be reduced to as low as 6 Hz without affecting classification performance. However, these steps alone only increase battery life of the eWatch platform by a factor of four.
When the platform is not actively collecting samples and classifying the result, the system can be put into a low power state, which consumes the absolute minimum power achievable. Obviously, the more time the system can spend in this low power state the larger the positive effect on battery life. By selecting discrete windows, to collect sensor samples and classify activity, we can improve battery performance. So how can this goal of selectively sampling the sensors, while maintaining the effectiveness of the activity recognition be achieved?

3. User study

The goal of this data collection is two fold. First, the data collected should be representative of daily activities. Second, the data should be representative of a range of individuals.

Data was collected using the eWatch platform sampling the 2-axis accelerometer continuously at 6 Hz. The activities were annotated by the participants, using an annotation application provided on the eWatch. The annotation list included: sitting, sitting in a meeting, standing, walking, running, cooking, eating, driving, sitting on the bus, standing on the bus, and other.

An annotation reminder application ran in the background and enabled an on-board vibration motor to remind the participant if more than 15 minutes passed without an annotation event.

We have, to date, collected data from four members of our research group (three graduate students and one faculty member) during their normal daily activities. Each participant collected data over three days, averaging 7.8 hours per day. The dataset contains nearly 94 hours of annotated activity.

4. Selective sampling experiments

The optimal selective sampling strategy takes one sample immediately following every transition between activities. This results in an accuracy rate of 100 percent, an average latency of 0 seconds, 0 percent redundancy in the samples taken and 0 percent of missed activities. For the purposes of our experiments, we assume that the annotation provided by the participants satisfy the conditions of the optimal sampling strategy.

All of the sampling strategies that we studied have the properties that they require little or no overhead in terms of computation and memory requirements for an on-line classifier. The base strategy, against which all other strategies are compared, is the uniform sampling strategy. In [5], the authors showed that uniform sampling performed nearly as well as a computationally expensive method using an HMM to model the user activities. Therefore, the goal of these initial sampling experiments were to do better than uniform sampling, while not significantly increasing the overhead.

Observing that the duration and transition behavior of one activity is very different from another, one method to improve the selective sampling performance is to create a different sampling strategy for each activity. We developed sampling strategies based on two statistics calculated from the dataset: the distributions of activity durations and the self loop transition probabilities for each activity.

4.1 Distribution based sampling

In distribution based sampling, we sampled uniformly over the distribution of duration times for each activity, instead of sampling uniformly over time. Figure 1 shows an example sampling schedule on a Gaussian distribution where the x-axis represents the duration of the activity. Approximately 20% of the distribution lies between each sample time, represented by the vertical lines.

![Figure 1. Example distribution based sampling schedule.](image)

4.2 Transition probability based sampling

In transition probability based sampling, we construct a first-order Markov model of the transition probabilities of a subset of the data using a time-step length of five seconds. Assuming the probability of remaining in the same activity for n time-steps is (self-loop probability of activity)\(^n\), we can select sleep times such that the n time-step probability falls below a given threshold between 0 and 1. For example, a threshold of 0.5 would select sleep times such that there is a 50% probability the activity didn't change during the sleep time. The higher the threshold value, the shorter the sleep time for each activity and thus the larger the percentage of total samples taken (compared to continuous sampling).

4.3. Metrics

There were four metrics used in rating the effectiveness of sampling strategies. The first and most widely used metric is that of recognition
accuracy. Accuracy is defined as the percentage of time that the last sampled activity matches the current, true activity. This is a good metric for judging the quality of a classifier, but is limited by the fact that not all activities are equally represented in a day so high accuracy on the most prominent activities masks failures on less frequent activities. This can be addressed by calculating accuracy by activity, but this is reported elsewhere.

The second metric is the percentage of activities that were missed by the sampling strategy. This is a useful metric for systems in which high accuracy in detecting transitions between activities is more important than simply knowing what the current activity is.

The third metric is average latency of detecting that a transition occurred. This metric is useful for judging strategies for real-time context aware systems, where it is not enough to know that a transition occurred, but to know a transition occurred in a timely fashion.

The final metric is redundancy of samples. This is the percentage of samples that are from the same activity as the previous sample. This measures how much improvement is possible given a sampling strategy.

5. Results

Figures 2 through 5 show semi-log plots of the average accuracy, missed activities, latency, and redundancy of all strategies using annotations as a perfect classifier’s output. There are three line per plots shown using the Markov model derived sleep times: 'sameday' means the transition model was calculated using the same data as was plotted, 'samesubject' means the transition model was calculated using all other days from the same subject and 'nonsubject' means the transition model was calculated using only data for other subjects.

In Figure 2, the recognition accuracy for uniform sampling remains above 90% until 3% of the total samples. This is because the majority of the time for all subjects was spent in one of three activities: sitting, standing, or walking. These more likely activities have high accuracies that mask the lower accuracies for less frequent activities. The two best performing markov strategies 'sameday' and 'samesubject' continuously perform better than uniform. At 10% of samples used they perform 2-3% better on average, but when only 1% of samples are used the difference in performance compared to uniform has grown to 9%.

Similarly for Figure 3, 'sameday' and 'samesubject' performed the best out of all strategies. For 10% of samples used, these missed 3% fewer activities than uniform, while by 1%, they miss ~10% fewer activities than uniform sampling. However,
'samesubject' becomes much less reliable between 1% and 3% of samples used. On all measures, the distribution model is the worst performer. Even considering all 94 hours of data, the most frequent activities were represented by between 100 and 150 data points, while the others were represented by fewer than 20. There were simply not enough examples of each activity to learn a reliable distribution of activity lengths.

Summarizing the results, the Markov sampling strategies compared more favorably to the uniform strategy than distribution based sampling. As would be expected, the 'sameday' strategy is the best performing and provides an upper bound on how well the strategy can perform. This is obviously not a feasible strategy in practice. To see how the Markov strategy generalizes, we look at the other two Markov plots.

Through Figures 2 to 4, deriving the sleep model from other days for the same subject performs nearly as well as 'sameday' for sampling as little as 10% of the time and continues to perform better than uniform sampling down to 1-3% of samples.

The sleep models calculated using 'nonsubject' data do not perform quite so well, but performs better than uniform on accuracy and latency until around 10% of samples and on missed activities until around 20%. It is also interesting to note in Figure 5 that 'sameday' and 'samesubject' have significantly more redundancy than uniform over the entire range, while 'nonsubject' is more comparable to uniform in this respect.

6. Conclusions and future work

The Markov derived sampling strategies provide improvement over uniform sampling using the metrics of accuracy, missed activities and latency. Even using sleep times derived from other subjects, the Markov strategy performs better than uniform using only 10-20% of the samples. Because the Markov strategy produces one sleep time for each activity, there is essentially no computational or memory overhead when compared to uniform sampling.

We are continuing to add to our dataset and are exploring other metrics, such as activity-specific accuracies. Finally, we think that it is possible to improve performance for specific activities at the expense of others, which would be a useful strategy to take if only a subset of a persons activities are important to the application.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Nos. EEEC-540865 and 0205266, the Defense Advanced Project Agency (DARPA) under Contract No. NBCHD030010, Pennsylvania Infrastructure Technology Alliance (PITA) - a collaboration among Commonwealth of Pennsylvania, Carnegie Mellon and Lehigh University, and a grant from Intel Corporation.

Additionally, we would like to thank Uwe Maurer and Anthony Rowe for eWatch support.

References