Trading off Prediction Accuracy and Power Consumption for Context-Aware Wearable Computing

Andreas Krause[1], Matthias Ihmig[3], Edward Rankin[2],
Derek Leong[2], Smriti Gupta[2], Daniel Siewiorek[1], Asim Smailagic[1], Michael Deisher[4], Uttam Sengupta[4]
[1]School of Computer Science, Carnegie Mellon University, Pittsburgh
[2]Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh
[3]Dept. of Electrical Engineering and Information Science, Technische Universität München, Germany
[4]Intel, Hillsboro, OR

Abstract

Context-aware mobile computing requires wearable sensors to acquire information about the user. Continuous sensing rapidly depletes the wearable system's energy, which is a critically constrained resource. In this paper, we analyze the trade-off between power consumption and prediction accuracy of context classifiers working on dual-axis accelerometer data collected from the eWatch sensing and notification platform. We improve power consumption techniques by providing competitive classification performance even in the low frequency region of 1-10 Hz and for the highly erratic wrist based sensing location. Furthermore, we propose and analyze a collection of selective sampling strategies in order to reduce the number of required sensor readings and the computation cycles even further. Our results indicate that optimized sampling schemes can increase the deployment lifetime of a wearable computing platform by a factor of four without a significant loss in prediction accuracy.

1. Introduction

Context-aware mobile computing uses wearable sensors to acquire information about the user, without requiring the user's attention, in order to enable proactive system behavior [8]. Continuous sensing and data processing requires the wearable computer to constantly run at full power, thereby rapidly depleting the system's energy, which is a critically constrained resource [10]. In [3], the decrease of energy consumption for varying sampling and bit rates was examined for a wearable sensing platform consisting of two hip- and knee-worn accelerometers. It was shown that a reasonable trade-off between prediction accuracy and sampling rate was achieved at a sampling rate of 20 Hz and 2 bit resolution. Lowering the sampling rate led to a drastic decrease in prediction accuracy. Sampling rates of 20 Hz pose a significant computational problem, and a memory consumption problem for the data collection scenario. In particular, the deployment of a knee worn accelerometer in a wearable computer system is difficult to realize on a larger scale.

In this paper, we analyze the trade-off between power consumption and prediction accuracy for the eWatch [9], a context-aware wearable platform, described in Section 2. It acquires activity context information from dual-axis accelerometer data. We improve on the results of [3] by achieving competitive classification accuracy with continuous monitoring even for the highly erratic and ambiguous wrist-based sensing location, while presenting a finer-grained resolution at the lower frequency region of 1-10 Hz. We also provide detailed estimates on the deployment lifetime of the eWatch depending on the selected sampling rates. While these results allow us to reduce the processor duty cycles by a significant amount, we additionally propose four selective sampling strategies in order to reduce the number of required sensor and computation duty cycles. We empirically analyze our methods for a real-world deployment of the eWatch devices. Our results indicate that optimized sampling schemes can increase the deployment lifetime of a wearable computing platform by a factor of four without a significant loss in prediction accuracy.

2. Wearable platform

We developed the *eWatch*, a wearable sensor and notification prototype in a wrist-mounted form factor [9], which offers a versatile platform with LCD, LED, vibration motor, and speaker for notification; Bluetooth for wireless communication; and buttons for interacting with the user. It perceives the user's context using a two-axis accelerometer, microphone, light and temperature sensors in combination

Proceedings of the 2005 Ninth IEEE International Symposium on Wearable Computers (ISWC'05)
0-7695-2419-2/05 $20.00 © 2005 IEEE

Figure 1. eWatch sensor platform.

with machine learning algorithms.

The eWatch hardware is based on a 32-bit ARM7 CPU with 64kb RAM running at 80 MHz. The acceleration sensor ADXL202 from Analog Devices has been used to capture the movements and orientation of the forearm including forces of +/- 2g.

The two accelerometer axes are aligned to be parallel to the LCD plane, which captures all "horizontal" movements. Sampling the third axis is not necessary, as the user experiments showed little extra movements along this axis that could increase classification accuracy. The eWatch is worn as shown in Figure 1 during the different user studies.

The eWatch is powered by a Li-Ion battery with a capacity of 700mAh. The ARM7 processor allows optimizing for power consumption through explicitly switching off the processor core clock and certain peripherals when they are not required. We assumed the following three different power states for estimating the power consumption of classifiers running on the eWatch:

(1) *Full Power*: eWatch runs at full-speed with all peripherals enabled. This state is used for acquiring sensor values from the accelerometer and processing them. In the current hardware and software configuration, this takes about 30ms, but that time can be further reduced with improved interrupt handling. The duration for classification depends on the type and number of feature vectors used.

(2) *Idle State*: The core clock is turned off, but all processor internal and external peripherals keep running. This state is used while waiting for the next sample to be collected and its time interval varies with the selected sampling frequency. For a sampling rate of 6 Hz, this is 166ms minus the time required for acquiring and processing sensor values.

(3) *Low Power*: The state is active most of the time when selective sampling strategies are used. The processor and its peripherals are shut down except for a real-time clock which schedules the next wake-up, as determined previously by the selective sampling algorithm. This reduces the power consumption to the lowest value possible. The remaining power goes into external components like the voltage regulator. The average length of this interval depends on how often selective sampling is conducted.

Table 1. eWatch power consumption

Power state	Current	Battery lifetime
Full Power	108 mA	6.5 h
Idle State	24 mA	29 h
Low Power	9 mA	78 h

Table 1 shows the power consumption for the whole eWatch device and how long it would run with that average current. Power consumption during the Full Power state can be further improved by reducing clock speed and fine-tuning the use of CPU internal peripherals. The current values correspond to the different power states described above and form the basis for later calculation of the average current and battery lifetime.

For continuous sampling with varying sampling rates, only *Full Power* and *Idle State* is used, whereas for selective sampling, all three power states are included in the calculation.

To determine the battery lifetime, the three power states are averaged, with parameters such as sample rate, number of selective sampling actions and the number of FLOPS influencing the time intervals of each power state.

For lower sampling rates, the time interval for the *Idle State* increases reciprocally, while the time interval for the *Full Power* state slightly decreases as the number of samples in a block also diminishes. This results in an improved battery lifetime, as illustrated in Figure 2. The lifetime is slightly higher for a time domain analysis, as no FFT is required and the vector dimensions are smaller.

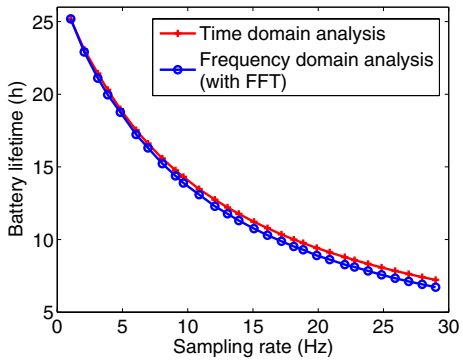


Figure 2. Battery lifetime for various sample rates

Proceedings of the 2005 Ninth IEEE International Symposium on Wearable Computers (ISWC'05)
0-7695-2419-2/05 $20.00 © 2005 IEEE

Additional power is required if the data is transmitted over Bluetooth. This module consumes an 8.5 mA on average when activated. Later calculations assume a classifier algorithm running locally on the eWatch with Bluetooth switched off.

To estimate the power consumption for running data processing and classification on the eWatch, we assumed a calculation throughput of 250 kFLOPS/s based on [1]. A real ARM-based implementation would work in integer arithmetic, which results in an even higher speed. We show an upper bound for calculation power, and the following estimates are rather pessimistic as they are based on the use of software floating point.

For an analysis in the frequency domain with a Support Vector Machine (SVM) at 25 Hz sensor sample rate collecting two-axis data for 5 seconds, we can estimate the number of FLOPS as follows:

- *FFT* with $n = 128$ values:
  $c_{\text{dual-axis}} \cdot c_{\text{impl. FFT}} \cdot n \cdot log_2(n)$
  $= 2 \cdot 5 \cdot 128 \cdot 7 = 8960$ FLOPS
- *SVM* with $D = 128$ dimensions and $n = 300$ vectors:
  $c_{\text{impl. SVM}} \cdot n \cdot D + n \cdot c_{\text{impl. exp}}$
  $= 2 \cdot 300 \cdot 128 + 300 \cdot 10 = 79800$ FLOPS

About 89 kFLOPS are required for one classification. For a continuous classification at 25 Hz, this reduces the running time by nearly 30 minutes from 8.1 to 7.6 hours; hence, the computation itself is a visible but not a dominant factor in reducing energy consumption.

The sampling interval is always five seconds, so a reduced sample rate will result in a lower number of samples for each block and in a lower number of required FLOPS. Together with appropriate power levels, this leads to greatly reduced power consumption.

3. Power-optimized classification

In this section, we describe the classification method we used and explore the impact of varying the sampling rate and choice of feature vectors on the classification accuracy and power consumption.

The first task in classification is to decide on how the raw signal $(x_t)_t$ should be represented in feature space. Motivated by the observation that human movement often appears periodic, *frequency spectrum*-based classification is commonly used for classifying accelerometer data [6, 3]. In this approach, short-term Fast Fourier Transformations are computed for short time windows from the collected samples. For very low sampling rates, however, one might expect that the frequency spectrum becomes less expressive, because high dominant frequencies might be lost in reflection. So in addition to using frequency based features, we also investigated *time domain*-based features. Guided by results from [6], we used five second windows for computing the features used in the classification process for both approaches.

For *frequency-domain-based* classification, we computed the absolute values of the frequency components using the FFT for both accelerometer axes separately. For *time-domain-based* classification, we computed the empirical means, variances, square root of the uncentered second moment and the median absolute differences. For both approaches, the labeled examples used to train the classifier were centered and normalized to have unit variance, as suggested in [6].

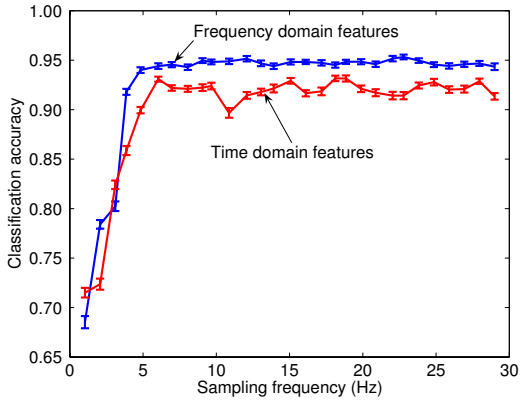
For the actual classification task, we used a multi-class Support Vector Machine (SVM) [11], which is a classification algorithm that often provides competitive or superior accuracy for a large variety of real-world classification tasks [7]. While training an SVM is computationally complex, the evaluation of the classification rule is very efficient and can easily be deployed on wearable hardware. To handle non-linear classification problems, SVMs can make use of the kernel trick, which allows implicit and non-linear embedding of the data in high-dimensional spaces, effectively allowing non-linear classification while avoiding overfitting problems. In our experiments, the Gaussian Radial Basis Function (RBF) kernel exhibited the best classification accuracy during cross-validation. Invoked with a set $(x_i, y_i)_i$ of labeled examples, the training procedure identifies a set of support vectors v_i among the labeled training, which support the maximum margin slab in the feature space induced by the Gaussian kernel $k(x, x') = \exp(-\frac{\|x-x'\|_2^2}{h})$. We employed the LIBSVM [2] implementation for our experiments as this library is widely used and often cited as reference.

3.1. Experiment Design

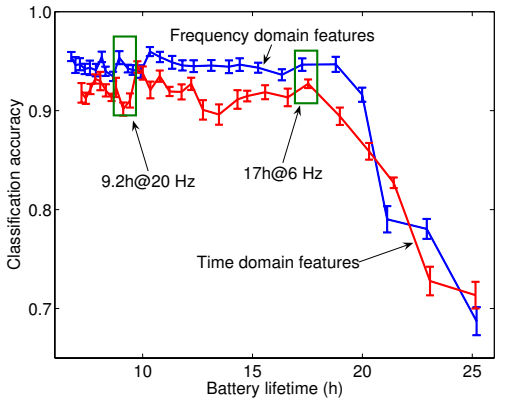
To construct the classifiers, training data was captured by three different test participants. The following five predefined activities were performed consecutively: *walking*, *running or jogging*, *standing*, *sitting or working* and *climbing* or *descending stairs*. Each activity was recorded for 10 minutes.

The data stream was first resampled at different target frequencies without using a low-pass filter. The resulting alias effects due to sub-nyquist sampling are actually desired since they help to at least capture some of the higher frequency effects even using low sampling rates. The data was then split into the different recorded activities and further partitioned into blocks of *five seconds*. From these five-second windows, we extracted the time- and frequency domain features as described above.

The labeled examples were subsequently used for train-



(a) *Dependence of accuracy on sampling rate*



(b) *Lifetime/accuracy trade-off for varying sampling rate*

Figure 3. Classifier performance.

ing multi-class SVMs with Gaussian Radial Basis Function kernels. The kernel width was determined using cross-validation, with 0.01 being the optimal choice for the unit variance normalized data. The generalization errors of the SVMs were computed using six-fold cross-validation.

In addition to computing the prediction accuracy for varying sampling rates, we estimated the power consumption based on the data described in Section 2.

3.2. Results

Figure 3(a) presents the dependency of the classification accuracy on the chosen sampling rate, using the frequency and time domain features. It can be seen that for all but the extremely low frequency ranges the frequency-based features perform superiorly. On the other hand, if power consumption is of utmost importance and only sampling rates of 1 Hz or less can be afforded, the time domain features achieve significantly higher classification accuracy. A main result of this analysis is that a sampling frequency of 6 Hz gives almost the same accuracy as achieved with much higher sampling rates. This significantly improves on the results from [3]. Figure 3(b) shows that a decrease of the sampling frequency from 20 Hz to 6 Hz increases the battery lifetime from 9.2 to 17 hours, which is an 85 percent increase. The relationship between sampling frequency and battery lifetime can be seen in Figure 2.

4. Selective sampling vs. prediction accuracy

To further reduce energy consumption and increase the deployment lifetime of the wearable platform, we observe that human activity is a rather continuous process: observed at a particular point in time, a person involved in one activity will be more likely to continue the activity than to change to another. We exploit this continuity by proposing a collection of selective sampling strategies. These schemes can be considered to *schedule* the use of the classifier presented above. Thus, selective sampling is a method of reducing the number of observations to save energy, from a continuous monitoring to selected time points, while keeping the accuracy of tracking the user's current activity as high as possible.

We can model the selective sampling problem as a sequential decision-making process, where we decide at each time step whether we want to sample and use the classifier to estimate the user's activity. We will model the activity sequence along with our notion of continuity as a (potentially instationary) Markov chain $(A_t)_t$, where A_t describes the activity of the user at time t. An example of such a Markov chain estimated from our data is shown in Figure 4. The selective sampling problem is then to select a set of observation times, subject to certain constraints, in order to maximize the probability of correctly predicting the user's activity at time steps when the sampler and classifier are not invoked. We can formalize the problem as follows: We want to minimize the expected loss

$$\min_{\pi,|\pi|\leq B} \mathbb{E}\left[\sum_{t=1}^{T} \mathcal{L}(P(A_t = \cdot|\pi(\mathbf{a})))\right], \qquad (1)$$

where the optimization is over conditional plans π, which are allowed to make at most B observations. Here, the expectation is taken over all activity sequences $\mathbf{a}$. A conditional plan π is a sequence of decisions, which, depending on the outcome of the observations made so far, decides when the next observation should be made. For each activity sequence $\mathbf{a}$, the selected observation times are denoted by $\pi(\mathbf{a})$. The objective function is the expected loss $\mathcal{L}$ incurred by only observing the sensor measurements $\pi(\mathbf{a})$ selected by the conditional plan, where the expectation is taken over all possible sequences of activities $\mathbf{a}$. The empirical loss measures the amount of uncertainty in

Proceedings of the 2005 Ninth IEEE International Symposium on Wearable Computers (ISWC'05)
0-7695-2419-2/05 $20.00 © 2005 IEEE

the probability distribution $P(A_t = \cdot \mid \pi(\mathbf{a}))$ of activity variable A_t conditioned on the observations $\pi(\mathbf{a})$ up to time t. We chose the entropy of the marginal distributions $\mathcal{L}(P(A_t = \cdot \mid \pi(\mathbf{a}))) = H(A_t \mid \pi(\mathbf{a})) = -\sum_x P(A_t = x \mid \pi(\mathbf{a})) \log P(A_t = x \mid \pi(\mathbf{a}))$. This objective function has been studied in [5] with applications to sensor networks. There it was proven that the optimal conditional plan π can be computed and represented in polynomial time using a dynamic programming approach.

In our experiments described below, we used four approaches for selecting the conditional plan π:

- *Uniform spacing* selects observation times at equally spaced intervals, starting at time 1.
- In *random spacing*, B observation times are selected uniformly at random, leading to time intervals of random length independent of the current activity. At time 1, a sample was guaranteed to be made.
- *Exponential backoff* samples at time 1, and maintains a maximum step size $\Delta_{\max}$. If at any time, the current activity is identical to the last detected activity, the maximum step size is multiplied by a fixed parameter $\alpha > 1$. If the activity is different, it resets $\Delta_{\max}$ to 1. The actual step size Δ is selected uniformly at random from the interval $[1, \Delta_{\max}]$, and the next observation is made at time $\lceil t + \Delta \rceil$. [4]. This strategy has been successfully employed in the Ethernet standard.
- The *entropy based* method acquires data according to a conditional plan π as to minimize the uncertainty in the marginal probability distributions $P(A_t = a \mid \pi(\mathbf{a}))$, measured using the entropy criterion. This strategy directly aims at improving the classification accuracy of the different activities by solving the optimization problem (1) as defined above and presented in [5]. Unlike the previous schemes, this method takes transition probabilities of the states into account, as can be seen in Figure 4. If the current state is known to have a short duration, the interval to the next sampling time is set shorter than for states with longer duration.

4.1. Experiment design

We performed one-hour user studies with four subjects wearing the eWatch. These subjects were different from those who collected data to train the classifiers in order to obtain person-independent classification results. In contrast to the previous experiment, it was not explicitly stated what activities should be performed and when.

The goal was to record a representative snapshot of user activity within one hour. To achieve this, the instructions were given in an indirect way. The subjects were asked to perform typical tasks such as "Go to the soda machine in the basement, get something to drink." or "Run to the bus stop and wait a few minutes." The user was asked to perform such activities at random times and return to the desk to continue working afterwards.

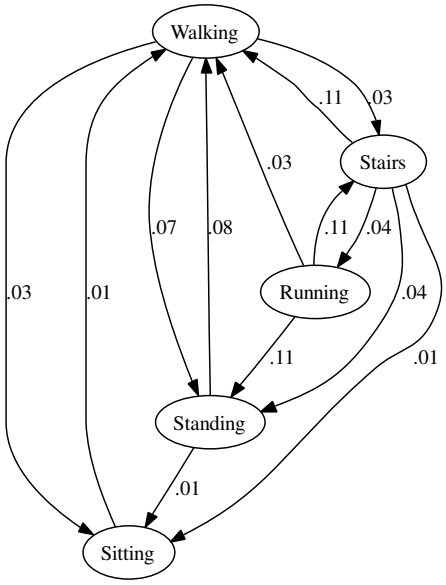


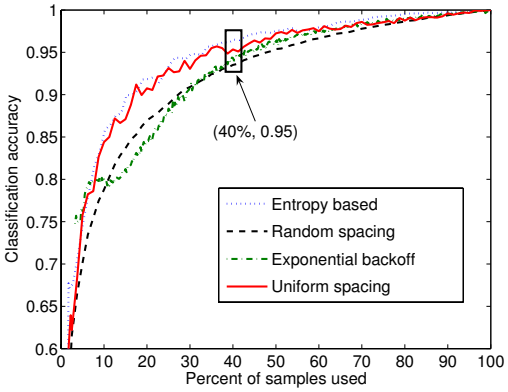
Figure 4. Transition probabilities estimated from annotated data

A second person was constantly following the subject and manually annotating the actual activities, as listed in Section 3.1. This procedure resulted in an exact list of activities such as *walking* and their respective length in seconds, which was used later in evaluating the different strategies for selective sampling. Sampled with a rate of 6 Hz, the recorded data from the acceleration sensors was then partitioned into sequences of five second blocks. These blocks were labeled according to the annotations and classified using the pre-trained classifiers in the frequency domain.

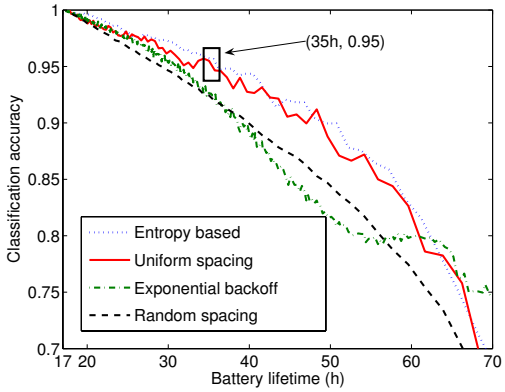
4.2. Results

Figure 5(a) presents the results of our selective sampling experiment, using the annotation as "exact classification" output. Using 100% of samples is equivalent to continuous sampling, resulting in 17 hours battery lifetime. Due to the dependence on random values, every trial for the exponential backoff and random spacing strategies results in different sampling points even for the same data. To get a representative result, the means from 100 random trials are shown. The uniform spacing and entropy-based strategies are deterministic, thus only one trial is shown.

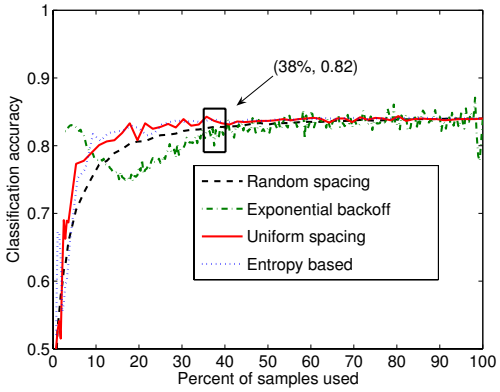
It can be seen that the random spacing strategy performs almost consistently worst. Exponential backoff, which was executed with the multiplication parameter α varying from 1.01 through 5.00, is very competitive if not superior for



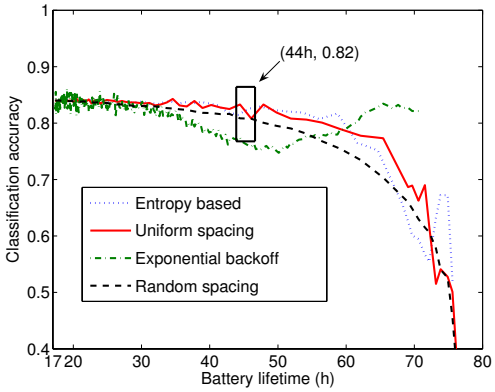
(a) *Effects of selective sampling strategies (annotated data)*



(b) *Lifetime/accuracy trade-off for selective sampling (annotated data)*



(c) *Effects of selective sampling strategies (classified data)*



(d) *Lifetime/accuracy trade-off for selective sampling (classified data)*

Figure 5. Selective sampling results.

very low limits on the number of observations (less than 10 percent of the potential 5 second windows used). In general, the entropy-based approach of finding an optimal conditional plan shows superior performance, improving the classification accuracy by several percent. The optimal plan uses activity transition probabilities estimated from the annotations, as presented in Figure 4. The uniform spacing strategy, which is much easier to implement, almost achieves the same performance, especially with increasing the number of observations B.

For example, Figure 5(a) shows that a reduction to 40% of the number of samples used still results in a 95% classification accuracy. The corresponding battery lifetime has increased by a factor of two from 17 to 35 hours, as seen in Figure 5(b).

We performed the same experiment again, using the output of the classifier, described in Section 3, instead of the annotations. The annotation was used only to compute the classification accuracy for each selected block of samples. In comparison to the "perfect classification" experiment described above, it can be seen that now the overall error is dominated by the classification error from the support vector machine, diminishing the error induced by the selective sampling strategy.

The classification accuracy is in general slightly lower than the results reported in Section 3, since the classifier was used for people it had not been trained for. Also, the recording was performed in a noisy real-world environment. The qualitative characteristics of the four selective sampling schemes are similar to the "perfect data" experiment, except that the exponential backoff strategy exhibits interesting performance: For a very small number of observations, it outperformed the other methods, consistently over all study participants. For the usage range of 10 through 40 percent of all possible observations, it performs worse than all other strategies. We plan to perform further studies to investigate whether these characteristics can be reproduced for a larger number of study participants. Figure 6 shows

Proceedings of the 2005 Ninth IEEE International Symposium on Wearable Computers (ISWC'05)
0-7695-2419-2/05 $20.00 © 2005 IEEE

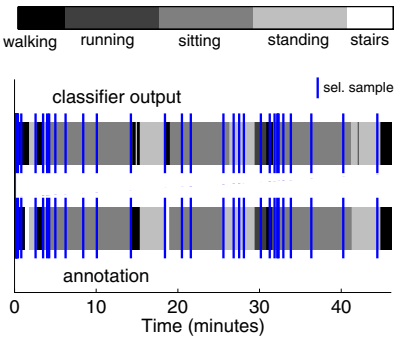


Figure 6. Example of exponential backoff sampling strategy

the observations selected by the exponential backoff strategy for a value of $\alpha = 4$. Each shade of gray indicates one of five different activities, and the vertical lines show the moments at which one block of samples is acquired. This example illustrates the effective use of a small number of observations to more densely cover regions of high activity than long regions of a single activity.

As can be seen in Figure 5(c), the classification accuracy is about 82% when only 38% of the samples are used at a sample rate of 6 Hz. Figure 5(d) shows that the corresponding battery lifetime increased from 17 to 44 hours, which is a factor of about 2.5.

5. Conclusion

In this paper, we explored options for trading off power consumption for classification accuracy in context-aware wearable computing. We showed how even very low frequency sampling of accelerometer data collected from the highly erratic wrist location can lead to classification results competitive with previous results for much higher sampling rates, and more inconvenient wearing locations. This approach effectively doubles the battery lifetime for our context-aware eWatch platform. We furthermore proposed a collection of selective sampling strategies which decrease the power consumption even further by exploiting continuity in human activity. Using this approach, we can again approximately double the lifetime of our wearable platform. We believe that our analysis can significantly benefit the real-world deployment of context-aware wearable computing devices.

Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010, the National Science Foundation under Grant Nos. 0205266 and 0203448, a grant from Intel Corporation, and the Pennsylvania Infrastructure Technology Alliance.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA), or the Department of Interior-National Business Center (DOI-NBC).

The authors would like to thank Anthony Rowe and Uwe Maurer for their contributions to the hardware and software design of the eWatch.

References

[1] Advanced RISC Machines (Ltd). Application note 55 – floating point performance. Technical report, http://www.arm.com/pdfs/DAI0055A_fp_performance.pdf, 1998.

[2] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[3] H. Junker, P. Lukowicz, and G. Tröster. Sampling frequency, signal resolution and the accuracy of wearable context recognition systems. In *Proc. 8th International Symposium on Wearable Computers (ISWC)*. IEEE Computer Society Press, 2004.

[4] L. Kleinrock and S. S. Lam. Packet-switching in a slotted satellite channel. In *AFIPS Conference Proceedings*, 1973.

[5] A. Krause and C. Guestrin. Optimal nonmyopic value of information in graphical models - efficient algorithms and theoretical limits. In *Proc. of IJCAI*, 2005.

[6] A. Krause, D. P. Siewiorek, A. Smailagic, and J. Farringdon. Unsupervised, dynamic identification of physiological and activity context in wearable computing. In *Proc. 7th International Symposium on Wearable Computers (ISWC)*. IEEE Computer Society Press, 2003.

[7] D. Meyer, F. Leisch, and K. Hornik. Benchmarking support vector machines. Report Series 78, Vienna University of Economics and Business Administration, November 2002.

[8] A. Smailagic and D. P. Siewiorek. Application design for wearable and context-aware computers. *IEEE Pervasive Computing*, 1(4):20–29, 2002.

[9] A. Smailagic, D. P. Siewiorek, U. Maurer, A. Rowe, and K. Tang. eWatch: Context-Sensitive Design Case Study. In *Proc. of the IEEE Annual VLSI Symposium*, pages 98–103. IEEE Computer Society Press, May 2005.

[10] M. Stäger, P. Lukowicz, and G. Tröster. Implementation and evaluation of a low-power sound-based user activity recognition system. In *Proc. 8th International Symposium on Wearable Computers (ISWC)*, 2004.

[11] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.