Choosing Duty-Cycle Parameters for Context Recognition

Vito Janko
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
vito.janko@ijs.is

Mitja Luštrek
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
mitja.lustrek@ijs.is

***Abstract*—The recognition of the user's context with wearable sensing systems is a common problem in ubiquitous computing. However, the typically small battery of such systems often makes continuous recognition impractical. An efficient method to reduce the strain on the battery is to employ duty-cycling – periodically turning sensors on and off. Its benefits increase if the duration of those periods is tailored to each context. In this work we present a general mathematical model to predict the effect of different duty-cycle parameters on the system and discuss ways of selecting suitable ones. The methodology was tested on a real-life dataset where it accurately predicted the system performance and found duty-cycle parameters that were better than those found with expert knowledge alone.**

***Index Terms*—context-recognition, energy consumption, duty-cycling, Markov model, multi-objective optimization**

I. Introduction

Widespread accessibility of wearable sensing devices opens many possibilities for tracking the users who wear them. Possible applications range from measuring their exercise patterns and checking on their health, to giving them location-specific recommendations. For example, using accelerometer data we could recognize if the user is walking, running, resting or performing similar activities.

A major limitation of such continuous sensing and context recognition is its heavy toll on the sensing device's battery life. This is especially relevant for smartphones, which have a very limited battery that must be shared between many applications, but the same limitation applies to basically any wearable device. There is an inherent trade-off between a system's energy consumption and its recognition quality. Increasing energy savings decreases the recognition quality and vice versa. This issue is often neglected when discussing the design of context-recognition systems, however, it is an essential component if such systems are to be used in practice.

In our experience [1] the largest energy-consumption reductions were made by using duty-cycling, so this will be the main focus of the paper. Duty-cycling is a method where a system is active for a given time, than turns off its sensors for a given time ("sleeps"), than turns them on again and repeats. By classifying, for example, only each tenth instance and assuming that context has not changed in-between, one can get a roughly tenfold decrease in energy consumed; much greater than what can usually be achieved with other methods (e.g. reducing sensor sampling frequency).

When designing a context-recognition system that employs duty-cycling, one must choose its parameters – how many time periods the system is active, and how many time periods it sleeps (one time period represents one classification window). Furthermore, one can decide on the sleep time for each context individually. Contexts that seldom change can use long sleeping times, while quickly-changing contexts need short ones. In related work [2] [3] most authors solve the issue by empirically trying a few parameters, or use expert knowledge of the domain.

In this work we present a general methodology to determine sensible duty-cycle parameters. It exploits the structure of the data – how frequently do contexts change, in which sequence they appear, and how well we can classify them. We first present a mathematical model that can predict how duty-cycling with different parameters affects the system performance. This gives us the ability to try many different parameters to find the ones that best suit our needs. We discuss how to choose the parameters based on the required precision and recall with an iterative method, and how to choose them based on arbitrary criteria using multi-objective optimization. To the best of our knowledge, we are the first to tackle the problem of duty-cycle parameter selection in a general domain.

While we present our findings in relation to context recognition, they can be applied in any area where duty-cycling is sensible and useful.

II. Methodology

Suppose we have a context recognition system that classifies sequential time periods into contexts $c \in C$. The quality of this classification is given by the confusion matrix CM.

We then add the duty-cycling – with parameters a and len – to this system in the following way: sensors are working for a periods, classifying that many contexts. After a classifications, the last classified context c defines the length of the sleeping phase. The system then turns off the sensors for the next $len(c)$ periods, and classifies all contexts in this time as c; i.e., it assumes that the context has not changed.

Our goal is two-fold. First, to create a mathematical model that will predict the behavior of this system – the proportion of the time the sensors are sleeping and the confusion matrix of the newly classified activities. Second, to propose a way of selecting appropriate parameters a and $len(c)$ for each c.

A. Modeling duty-cycles

When building the model we assume that the context sequence has the Markov property – the probability of the next context depends only on the current one. Additionally, we assume that the classification errors happen randomly, with the distribution given by the confusion matrix.

Given the assumptions, the evaluation of a system with given duty-cycle parameters can be performed with the subsequently defined steps. This evaluation needs only a few matrix (of size $|C|$) multiplications and summations and can be therefore done almost instantly; in any case much more quickly than running a complete experiment (simulating duty-cycling on a dataset, classifying all the instances). This allows us to evaluate many different duty-cycle parameters and thus increases the chances of finding a good solution.

1) Duty-cycle types: We break the data into duty-cycles of different types that start with the last active period ("head") as shown in Figure 1. Cycle type depends on the true context of that period, in addition to the context that period is classified as – indices of these contexts are subsequently named t and p respectively. For each of those cycle types, we can use the following equations to estimate how many of each contexts appear in both the sleeping (3) and active (4) phase.

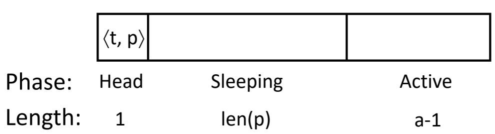

Fig. 1. Structure of a $\langle t,p\rangle$ duty-cycle. Starts with the context t classified as p, followed by the sleeping phase where all contexts are classified as p, and an active phase. The last period in the active phase belongs to the next cycle.

$$e_i = [0,..,0,\underbrace{1}_{i}0,...,0] \tag{1}$$

$$P_{t,i} = e_t T^i \tag{2}$$

$$ES_{t,p} = \sum_{i=0}^{len(p)} P_{t,i} \tag{3}$$

$$EA_{t,p} = \sum_{i=len(p)+1}^{len(p)+a-1} P_{t,i} \tag{4}$$

e_i – i-th base vector. Represents the context distribution in the first period of a cycle.
T – transition matrix, $T_{i,j}$ represents the probability of the next context c_j, given the current context c_i. Easily obtained from the dataset.
$P_{t,i}$ – probability distribution of contexts after i steps, given the current context c_t. Vector of size $|C|$.
$ES_{t,p}$ – average number of contexts that appear while sleeping, given that this sleeping period started with the true context c_t that was classified as c_p. Vector of size $|C|$.
$EA_{t,p}$ – average number of contexts that appear while active, given that this cycle started with the true context c_t that was classified as c_p. Vector of size $|C|$.

2) Distribution of duty-cycle types: We calculate proportionally how many cycles of each type there are. This can be done by estimating the probability that a cycle of one type is followed by a cycle of another type (5).

$$\langle t_1,p_1\rangle \rightarrow \langle t_2,p_2\rangle = (P_{t_1,len(p_1)+a})_{t_2} CM_{t_2,p_2} \tag{5}$$

$(P_{i,j})_k$ – k-th component of the vector $P_{i,j}$
$\langle t_1,p_1\rangle \rightarrow \langle t_2,p_2\rangle$ – probability of the next cycle type being $\langle t_2,p_2\rangle$, given the current cycle type $\langle t_1,p_1\rangle$. This happens if the last context of the previous cycle is c_{t_2}, classified as c_{p_2}.

Having the transitional probabilities, we can use basic Markov-chain calculus [4] to calculate the "steady state", which gives us the desired cycle-frequency proportions D:

D – Distribution of the cycle types. $D_{i,j}$ is the proportion of $\langle i,j\rangle$ cycles.

3) System evaluation: Combining points 1) and 2) we can estimate the performance of the system in the form of its confusion matrix (6) and its energy gain (7) compared to its sensors always being active.

$$CM^*_{t,p} = \sum_{i=0}^{|C|} D_{i,p}(ES_{i,p})_t + \sum_{i=0}^{|C|}\sum_{j=0}^{|C|} D_{i,j}(EA_{i,j})_t CM_{t,p} \tag{6}$$

$$gain = \frac{1}{a}\sum_{i=0}^{|C|}\sum_{j=0}^{|C|} len(j) D_{i,j} \tag{7}$$

$CM^*_{t,p}$ – Element (t,p) of the confusion matrix of the system that employs duty-cycling with given parameters. The first part of the equation is the performance in the sleeping phase, the second in the active phase. Note: The sum of all elements in this new matrix is different than in the original CM matrix; this can be easily solved with normalization if needed be.
$gain$ – Proportion of sleeping periods compared to the active ones. If a classifier classifies each tenth time period, $gain = 9$.

B. Choosing duty-cycle parameters

Section II-A gives a way to evaluate the performance of a system with given parameters, while here we discuss methods for efficiently selecting sensible parameters for evaluation. The space of possible parameters is usually large even if we limit the maximum cycle length: if we have $|C|$ contexts with $len \in [0,m_c]$ and $a \in [0,m_a]$, then there are $m_a m_c^{|C|}$ different parameters – making the brute-force approach infeasible.

The first insight is that according to the described model, there is no advantage in having the active period a longer than the minimum of 1 in most cases. Intuitively, if a context is assigned a sleeping phase with length l, then it is assumed that the context will not likely change in the next l periods; classifying multiple instances in a row is therefore unnecessary. The exception is when trying to get very small energy consumption reductions (to minimally decrease performance), as a $gain$ smaller than 1 is impossible to achieve with $a = 1$. There are two practical advantages, however, in having $a > 1$; a) if there is a cost in turning the sensors on or off (GPS is a notable example) and b) if the classification accuracy is poor, since one might consider classifying more than one instance and applying some smoothing before deciding on the next sleeping phase length.

To decide on the sleeping period lengths, we can resort to one of the following two methods.

1) Iterative method: Duty-cycling causes an error if a sleeping period coincides with a transition between two (or more) contexts: c_1 and c_2 for example. Two types of errors (8) happen: a) Precision of c_1 drops, as instances of c_2 are misclassified as c_1. The drop in precision depends only on $len(c_1)$ and the average length of c_1 streak. b) Recall of c_2 drops, and the drop depends on $len(c_1)$ and on the average length of c_2 streak.

$$c_1 \quad \underbrace{c_1 \quad c_1 \quad \overbrace{c_2 \quad c_2}^{\text{Precision of } c_1 \text{ and recall of } c_2 \text{ fall}}}_{\text{sleeping period}} \quad c_2 \tag{8}$$

Therefore, $len(c_1)$ should take into account both the streak length of c_1 and streak lengths of the contexts that are likely to follow it. If we have requirements on precision/recall for each context we can iteratively find the upper bounds for $len(c)$ for each c. This can be done by increasing the $len(c)$ for a given c (while not duty-cycling any other context) until either the precision or recall of some context drops below the required boundary; and then repeating the process for all other contexts. If we are only interested in precision, the found upper bound is tight, and ideal given requirements. This method is guaranteed to work only if we assume that the classifier works perfectly, as it does not model errors generated by misclassifications.

Using this method can serve as a semantic explanation for the sensible sleeping lengths – for example, it may show that some contexts shouldn't use long sleeping lengths because they are too short or because they are followed by another short context.

2) Multi-objective optimization: If we are optimizing some other criterion – e.g., the accuracy, we can use multi-objective optimization. The problem of choosing the parameters has two conflicting objectives: reducing the system's energy requirements, while retaining its classification quality. Both are easily calculable with the described model. We used NSGA-II [5], a genetic multi-objective algorithm, but we assume that any similar algorithm could be used for the task. The output of this algorithm is an approximation of the Pareto front, giving us sensible duty-cycle parameters. These parameters give us different trade-offs between the classification quality and consumed energy, from which the system designer, who knows the requirements of the system, can pick a suitable solution.

III. Experimental evaluation

A. Datasets

Our methodology was evaluated on two datasets. One completely artificial, and another from real life.

1) Artificial dataset: This dataset contains 5 different contexts, generated in sequence with the Markov property. Since our mathematical model assumes these properties, it should perform almost flawlessly given a large enough sequence. Transition probabilities were randomly selected, as were the confusion matrix values.

2) Commodity12 dataset: A real-life dataset, where a smartphone and a chest-worn heart-rate monitor were used to monitor 10 participants. Each participant captured continuous two weeks of data and hand-labeled the following contexts: sleep, work, home, eating, transport, exercise, out (out of house, but not in any of the previous contexts). The data came from ten different physical and virtual sensors: acceleration, pressure, light, GPS location, a list of visible WiFi networks, location description by the Foursquare web service, sound, time, heart rate and respiration rate. The first eight were measured with the smartphone, while the last two with the heart-rate monitor, which was connected to the smartphone via Bluetooth. Features were calculated for each minute of the data, and one minute became one learning instance. All details can be found in our previous work [6].

While the classification accuracy was reasonably high, energy consumption of the application prevented it from seeing any practical use. In our previous work we created a general pipeline for assigning sensor settings to contexts [1] and found that duty-cycling was the most efficient in reducing the system energy consumption. In that work we used hand-picked duty-cycle parameters that were the same for each context. We now use our new methodology to pick better parameters.

B. Evaluation of the duty-cycle modelling

First we evaluated our mathematical model for modelling duty cycles and evaluating their performance. To do so we sampled 100 random duty-cycle parameters and than evaluated their performance both with the model and by actually going through the dataset and simulating real duty-cycles. We then compared the two. This was done for both the artificial dataset and for the Commodity12 dataset. The results are listed in Table I and show a high fidelity of the values predicted by the model to the real ones.

TABLE I
PREDICTION ERROR OF OUR DUTY-CYCLE MODEL, PREDICTING ACCURACY AND ENERGY

Dataset	Accuracy [%]	Energy [gain]
Artificial	0.1	0.001
Commodity12	1.1	0.051

C. Evaluation of the duty-cycle parameters on the Commodity12 dataset

In the first experiment we used NSGA-II to calculate a Pareto front approximation for the Commodity12 dataset

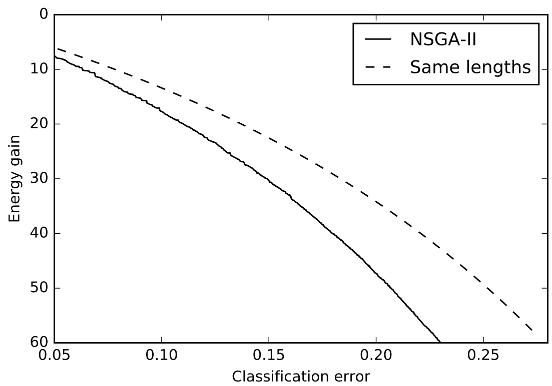

Fig. 2. Non-dominated set of solutions, found both with multi-objective optimization and by by setting the same sleeping period of the duty cycles for all contexts (up to the duration of 60 periods).

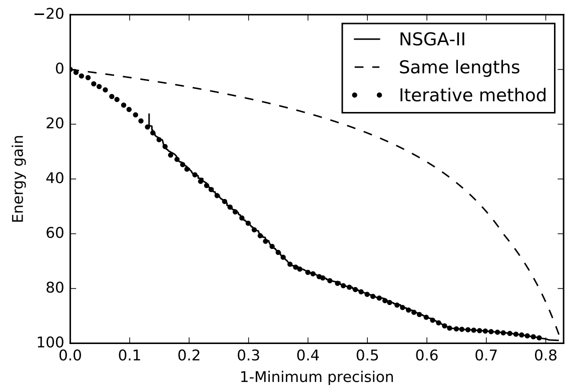

Fig. 3. Non-dominated set of solutions, found with multi-objective optimization, iterative method and by setting the same sleeping period of the duty cycles for all contexts (up to the duration of 100 periods).

(Figure 2), optimizing accuracy and energy gain. We compared it to solutions where all contexts had the same duty-cycle length. By not optimizing the sleeping length for each context, we can see that for the same energy gain, we lose points of accuracy. The error in the case of the optimized duty-cycling is up to 5 percentage points smaller (which represents roughly 20% of the total error) than in the "same lengths" case with the similar energy consumption gain. In both cases, we ignored the confusion matrix (replaced it with the identity matrix) to isolate the error generated solely as the result of duty-cycling. A similar shape of the solution set appears if we include confusion matrices, although the difference is slightly smaller, as the uncertainty of the classification discourages the use of highly unbalanced duty-cycle lengths.

In the second experiment, we tried to to select the parameters in a way that maximizes the precision of the context with the lowest precision. Solutions were found both with the iterative method as well as with multi-objective optimization. The results are shown in Figure 3. Both methods found mostly the same duty-cycle parameters (iterative method was roughly an order of magnitude faster), as they are optimal given the problem description, and are much better than those that use the same sleeping lengths for all contexts.

In all cases the longest duty-cycles were assigned to the contexts "Work" and "Home". This corresponds to our intuition, as those two contexts changed the least frequently. In the same spirit, the fast changing contexts such as "Eating" were assigned short cycles. The active period was chosen as 1 in all but a few border cases – as predicted in Section II-B.

IV. Conclusion

In this work we presented a mathematical model that allows us to evaluate the performance of a context-recognition system when using duty-cycling with different parameters. On the tested real-life dataset it was shown to work quickly and accurately. It needs no expert knowledge – it uses only the information on context sequencing, and their classification accuracy – and should work on any domain. In combination with the presented methods for efficiently selecting the duty-cycle parameters, it provides a useful tool for a system designer trying to make their system more energy efficient. In our case, we found duty-cycle parameters for the Commodity12 dataset that outperformed the ones we previously picked with expert knowledge alone.

The presented methodology can be used in conjunction with our other work on energy efficiency [1], which is able to optimize different sensor settings – which sensors to use, sampling frequency, duty-cycling... – and apply them to both individual contexts as well as to individual sensors, but can only work with fixed duty-cycle lengths. Future work on the topic will include more smoothly connecting the two, as well as testing our methodology on different domains.

V. Acknowledgement

The HeartMan project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 689660.

References

[1] V. Janko and M. Luštrek, "Energy-efficient data collection for context recognition," in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM, 2017, pp. 458–463.

[2] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The jigsaw continuous sensing engine for mobile phone applications," in *Proceedings of the 8th ACM conference on embedded networked sensor systems*. ACM, 2010, pp. 71–84.

[3] 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 the 7th international conference on Mobile systems, applications, and services*. ACM, 2009, pp. 179–192.

[4] "Finite-state markov chains," http://www.rle.mit.edu/rgallager/documents/6.262-4vaw.pdf, accessed: 2017-11-08.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[6] B. Cvetković, V. Janko, A. E. Romero, Ö. Kafalı, K. Stathis, and M. Luštrek, "Activity recognition for diabetic patients using a smartphone," *Journal of Medical Systems*, vol. 40, no. 12, p. 256, 2016.