Markov chain model for energy-efficient context recognition

Vito Janko
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Jamova 39, 1000 Ljubljana, Slovenia
vito.janko@ijs.si

Mitja Luštrek
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Jamova 39, 1000 Ljubljana, Slovenia
mitja.lustrek@ijs.si

ABSTRACT
Continuous sensing of the user and subsequent context recognition using wearable sensors is a popular area of research. One of the big problems of such automatic context recognition is the battery life of the sensing device. To maximize the energy efficiency, the context recognition system should adapt its settings to the current situation. Choosing the appropriate setting for each situation usually requires either a lot of expert knowledge or extensive experimentation. We propose a method that simulates all possible combinations of contexts and settings using a Markov chain model, automating and speeding up the whole process. We show on a small example that the simulation is accurate, and that it allows us to quickly select best trade-off between the energy efficiency and context-recognition accuracy.

Categories and Subject Descriptors
G.3 [**Probability and statistics**]: Markov processes; H.2.8 [**Database Applications**]: Data mining

Keywords
Context recognition; continuous sensing; energy efficiency; Markov chain; acitivity recognition.

1. INTRODUCTION
Widespread accessibility of wearable sensing devices allows for 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. The recognition of user's context using sensors is a popular and mature area of research [2]. For example, using activity recognition, we can recognize "walking, "running", "resting" and similar activities from accelerometer data. This task was made easier and more practical with the increased use of smartphones, which have many sensors built in and are often carried. Sensing with multiple sensors, possibly at once, opens additional options for context recognition: detecting one's location, ambient sound level, or some higher level activities such as "shopping", "traveling" or "working".

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. 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.

Some solutions deal with this issue by optimizing the sampling rate or sensor duty cycle for the particular recognition task [1]. This alone however, might be suboptimal. We might for example want to have the GPS active at a high sampling rate while the user is driving, but at a very low rate when he is working in the office. This calls for dynamic changes in the settings for both the sensors and for the subsequent processing. Many adaptive approaches already exist [3][4][5][6].

A problem with these and similar solutions is that they have complex pipelines and/or require many parameters specific to the particular recognition problem. If we were to recognize different activities using different sensors we would have to adapt these parameters, requiring either a lot of experimentation or expert knowledge. It would therefore be useful to have a method that can be provided with a sensor-rich dataset, and it would be able to tell which sensors or which sensor setting to use in each context.

A relatively simple solution of the above problem can be found in the work of Yan, et al. [6]. They select the sensor and attribute settings based on the last classified activity. For each activity they find the setting that best recognizes it, by testing all the settings on their dataset, and then use it when that activity is detected. However, since the selection is made for each activity in isolation, the effect on the whole system can be unpredictable. To illustrate: an accelerometer is very good at recognizing walking and resting, while a GPS is very good at recognizing driving. However, if we have only the accelerometer active while walking, driving will never be detected and the sensor switch will never occur. To take such interactions into account, we would have to run an experiment that has a specific setting for each activity and switches between them in runtime. Since there are many (activity, setting) combinations, this process might be prohibitively time consuming, possibly taking months on large datasets with many possible settings.

We propose a Markov chain model to simulate runtime settings switching and predict what would happen if we used a particular setting for a particular context. Using this model requires only a small amount of actual experimentation. Since the experimentation is a lot more time consuming than the simulation, many more combinations can be tried out and a better combination is therefore expected to be found with the same resources. In this paper we explain the proposed Markov chain and try it on a simple activity recognition dataset.

2. METHOD DESCRIPTION

Suppose we have a sensing system that works as follows: the user is in one of predefined contexts (for example, the context can be the current activity) and each context is associated with some setting. Settings can be which sensors are in use, or what sampling frequency or duty-cycle policy is active, or which feature set is used for classification, and so on. Using the current setting, we watch for a possible context change, and if it happens, we switch the sensor setting to the one assigned to the new context. For example: if we are sitting we need a lower sensor frequency, compared to when we are walking. To determine the optimal parameters for such a system, we might want to try every combination of assignments of reasonable settings to possible contexts. If we have c contexts and s settings, we would have s^c different combinations. Our goal is to make only s experiments and simulate the rest, gaining a drastic increase in efficiency.

We begin by selecting all reasonable settings. For each of them, we make an experiment where the classification model is trained and tested with this particular setting. For each experiment, we calculate and remember the confusion matrix. Additionally we need the transition probability from one state to the next one; that can easily be inferred from the dataset. Finally we must make energy consumption estimation for each setting. Energy consumptions for most sensors at different configurations are known or can be estimated with simple measurements.

To simulate an experiment where the settings are dynamically switched depending on the current activity, we create a Markov chain. This model has a state for each possible (context, setting) combination. The Markov state (c,s) represents that we are currently in context c, with the system having setting s (which depends on which context the system believes we are currently in).

Next we have to calculate the transition probability from one Markov state to another. They can be calculated from the transition probabilities of contexts and data from the previous computed confusion matrices. Intuitively: we get in a state (c,s) if the context really changes to c and if the system classifies this instance into one of the contexts that have assigned setting s.

$$S(c_1, s_1) \to S(c_2, s_2) = T(c_1, c_2) \sum_{c \in C(s_2)} C_{s_1}(c|c_1)$$

$S(c_1, s1)$ - the Markov state with context c and setting s.
$T(a, b)$ - the probability in the dataset that the next context will be b given that the current one is a.
$C(s)$ - the set of all contexts that use setting s.
$C_s(c_1|c_2)$ - the probability that the classifier that works with setting s will classify an instance to c_1, if the true context is c_2.

Having all transition probabilities, we can use the basic Markov chain calculus to calculate the steady state of the Markov chain. This gives us the amount of time the system will be in each of the states. Since we know how much time any setting is active and how much energy this setting consumes per time unit, the energy consumption of the whole system can be estimated. Additionally, since confusion matrices give us the accuracy for each state, we can calculate the accuracy of the whole system.

$$\text{Energy estimation} = \sum_{m \in M} t(m) e(s(m))$$

M - the set of all states in the Markov chain.
$t(m)$ - the predicted proportion of time spent in state m.
$e(s)$ - the energy requirement of a setting s in a given time unit.
$s(m)$ - the setting corresponding to state m.

$$\text{Accuracy estimation} = \sum_{m \in M} t(m) acc(c(m), s(m))$$

$acc(c, s)$ - the accuracy of the classifier that works with setting s, if the true context is c.
$c(m)$ - the context corresponding to state m.

It should be noted that many other metrics can be determined from such a model. Example: the accuracy for a particular activity or the latency of activity change detection. They can be used instead of the accuracy when evaluating the performance.

Every simulated experiment represents a possible trade-off between the accuracy and energy consumption. Simulating all the combinations, the Pareto optimal trade-offs can than be presented to the application designer, which - knowing the energy and accuracy requirements of the application - can then choose the ideal settings for it.

3. EXPERIMENTAL SETUP

We will demonstrate our method on a simple example. We have a dataset of accelerometer data generated by a smartphone, and we want to classify basic activities (Table 1) from it. We can do this by using two different settings: a high frequency sampling (50 Hz) and a low frequency sampling (2 Hz). Two extremes were chosen for simplicity. In practice we would perhaps also try other frequencies (10 Hz, 20 Hz etc...) and duty cycles. We assign one of the two frequencies to each activity, generating 16 different combinations. All 16 combinations will be simulated by doing only 2 actual experiments. In this case we might expect that the best combination would be using the low frequency for the activity rest, and the high frequency for the other activities, but the trade-offs are not so clear in general.

The confusion matrices are in Table 2 and Table 3. States of the generated Markov chain are in Figure 1. Note that such a chain is in principle fully connected, including the connections of each state to itself.

s	w	r	c
62.7	21.1	10.4	5.8

Table 1: Proportion (in %) of each activity in the dataset. The values are in %. s - rest, w - walking, r - running, c - cycling

	s	w	r	c
s	97.6	1.4	0.0	1.0
w	5.6	86.2	5.5	2.7
r	0.7	10.0	89.1	0.2
c	22.9	16.4	0.0	60.7

Table 2: The confusion matrix using the high frequency. The values are in %.

	s	w	r	c
s	97.2	2.3	0.1	0.4
w	9.5	82.4	7.2	0.9
r	3.9	28.4	67.7	0.0
c	46.8	32.6	1.5	19.1

Table 3: The confusion matrix using low frequency. The values are in %. We can observe that the accuracy for rest does not change much compared to the high frequency case, while the accuracy loss when cycling is quite drastic.

We also did a simpler simulation in the spirit of Yan, et. al [6], where every activity was considered in isolation. In this case, the accuracy of the system was computed simply by computing the accuracy for each activity given its setting and then weighted by this activity's proportion in the dataset.

4. RESULTS

The method was evaluated in the following way. First all the simulation trade-offs were computed (both Markov chain simulation and the simple one). Then we ran the actual experiments we were simulating, switching classifiers and sampling frequencies during the runtime. All three sets of results were plotted in Figure 2.

The trade-offs in Figure 2 are marked with letters that correspond to the activities where the low frequency was used. The case where only rest was used with the low frequency (marked as 's') could be considered the best trade-off between the energy gained compared to the accuracy lost.

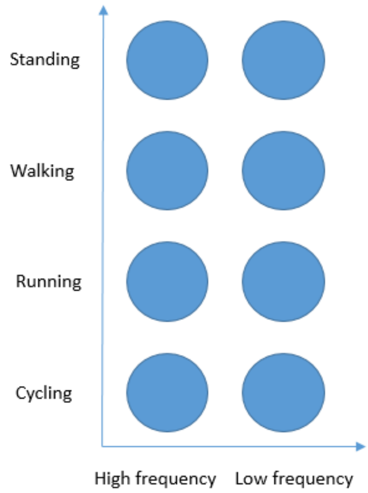

Figure 1: Markov chain states for our example. Vertical axis signifies the true activity, while the horizontal signifies the setting, which depends on the last classified activity.

We also plotted the Pareto front that shows the sensible solutions. We see that the Markov chain simulation points very closely correspond to the non-simulated ones. The simple simulation captures the general trend of the Pareto front, but makes substantial mistakes in predicting the actual values. If the interaction between sensors and activities were more complex, we expect the error to be be even greater. The error is numerically evaluated in Table 4.

	acc.	energy
Markov	2.03	0.35
Simple	12.82	1.83

Table 4: The average prediction mistake, made on energy consumption and classification error for the simple and Markov chain model. The values are given in % with respect to the maximal value for the corresponding axis.

We also explored what happens if the underlying activity distribution in the dataset changes. This can be easily simulated by modifying the transition probabilities of the Markov chains. It turns out that while the values themselves change drastically, the overall shape of the Pareto front remains similar. This means that the best simulated trade-off likely remains the best with most other activity distributions. Such simulation is also handy to see if the energy requirements exceed the application limits if some condition changes. Note that no additional experiments with the actual data were needed to generate this information, which is an additional benefit of our method.

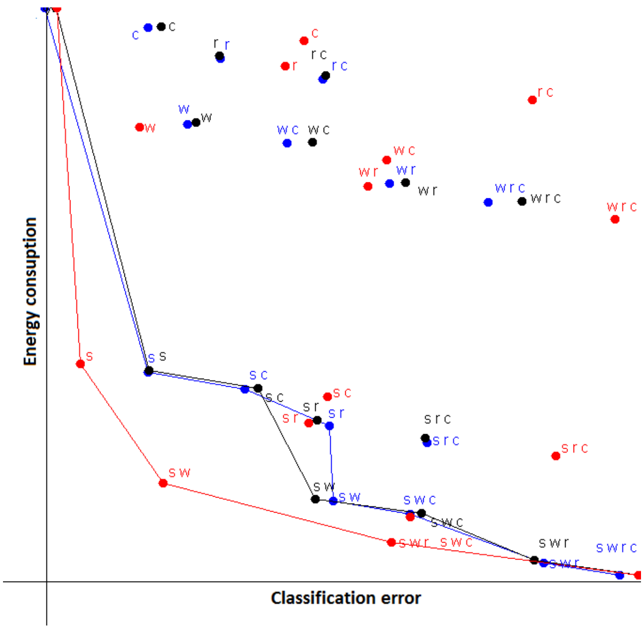

Figure 2: Black points are real trade-offs, blue points are simulated with Markov chains, red points are simulated with the simple model. The Pareto fronts are drawn using corresponding colors. The lower left corner represents the point with the lowest error and the lowest energy consumption. We can see that Markov chain simulation corresponds very closely to the values of real experiments.

5. CONCLUSION AND FUTURE WORK

The simulations display a very high decree of fidelity to the actual experiments and are very fast. Our method can thus effectively tackle the important but difficult task of selecting system settings that give us a good compromise between the accuracy and energy efficiency.

Future work on the topic will include testing the proposed method on a more complex dataset that contains more sensors and activities to see if the results still have the same fidelity. Another improvement will be to explore options for searching the settings combination space more efficiently. Since this is essentially a multi-objective optimization problem, many approaches from that area can be then used to further increase the effectiveness of our method.

6. REFERENCES

[1] Aftab Khan, Nils Hammerla, Sebastian Mellor, and Thomas Plötz. 2016. Optimising sampling rates for accelerometer-based human activity recognition. *Pattern Recognition Letters* (2016).

[2] Seon-Woo Lee and Kenji Mase. 2002. Activity and location recognition using wearable sensors. *IEEE pervasive computing* 1, 3 (2002), 24–32.

[3] Hong Lu, Jun Yang, Zhigang Liu, Nicholas D Lane, Tanzeem Choudhury, and Andrew T Campbell. 2010. The Jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM conference on embedded networked sensor systems.* ACM, 71–84.

[4] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. 2010. Energy-efficient rate-adaptive GPS-based positioning for smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services.* ACM, 299–314.

[5] Yi Wang, Jialiu Lin, Murali Annavaram, Quinn A Jacobson, Jason Hong, Bhaskar Krishnamachari, and Norman Sadeh. 2009. 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, 179–192.

[6] Zhixian Yan, Vigneshwaran Subbaraju, Dipanjan Chakraborty, Archan Misra, and Karl Aberer. 2012. Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach. In *Wearable Computers (ISWC), 2012 16th International Symposium on.* Ieee, 17–24.