

# ADAPTIVE SAMPLING IN CONTEXT-AWARE SYSTEMS: A MACHINE LEARNING APPROACH

Alex L. Wood, Geoff V. Merrett, Steve R. Gunn,  
Bashir M. Al-Hashimi, Nigel R. Shadbolt, Wendy Hall

Electronics and Computer Science, University of Southampton, UK, SO17 1BJ  
{alw306, gvm, srg, bmah, nrs, wh}@ecs.soton.ac.uk, +44 23 8059 4996

## Abstract

As computing systems become ever more pervasive, there is an increasing need for them to understand and adapt to the state of the environment around them: that is, their context. This understanding comes with considerable reliance on a range of sensors. However, portable devices are also very constrained in terms of power, and hence the amount of sensing must be minimised. In this paper, we present a machine learning architecture for context awareness which is designed to balance the sampling rates (and hence energy consumption) of individual sensors with the significance of the input from that sensor. This significance is based on predictions of the likely next context. The architecture is implemented using a selected range of user contexts from a collected data set. Simulation results show reliable context identification results. The proposed architecture is shown to significantly reduce the energy requirements of the sensors with minimal loss of accuracy in context identification.

**Keywords:** context awareness, energy efficiency, adaptive sampling

## 1 Introduction

Mobile computing has now reached such level of diffusion in everyday life that many of us are surrounded by computers everywhere we go [15]. As a result, an area of particular interest in pervasive computing is context awareness: that is, allowing systems to intelligently adapt to their changing environment. At the most basic level of context awareness, mobile phones dim or brighten their screens according to ambient light; at a more advanced level, smart homes employ a wide range of sensors and processing techniques to adapt to their inhabitants needs [18]. The recognition of these various *contexts* ('dark' or 'light' for the phone, 'empty', 'breakfast', 'inhabitants at work', etc. for the home) comes from the interpretation of a range of sensors. Generally stated, a widely-accepted definition of context is that it consists of information that can be used to characterise the state of anything relevant to the user [4]. Context awareness constitutes a fundamental component of many novel systems, such as smart homes which assist people living independently

[18], offering reminders at opportune moments [5], switching mobile phone profiles automatically [8], or even improving efficiency in industrial processes [6]. These improvements are only possible as a result of the improved understanding of the state of a system, in the form of context awareness. One can even go on to consider the likely next context [16], allowing systems to adapt proactively.

In the literature, a range of machine learning approaches have been demonstrated in identifying contexts. Many techniques use supervised methods, in which training data is collected and manually marked with the correct context. From this information, models can be built using a variety of statistical methods [7, 13]. Alternatively, unsupervised learning recognises emerging patterns in data without manually marked training data. In this category, approaches such as neural networks [17], nearest-neighbour algorithms and hidden Markov models have been explored [9]. The advantages of unsupervised learning are clear, in that the process is completely autonomic, but the trade-off is that the learnt contexts identified may not clearly map to user-interpretable situations. This is only a problem if the final output needs to be interpretable in this way. If the intention is purely to group together similar events in time, unsupervised learning can be well suited.

Recognition of contexts is, however, associated with energy costs entailed by the use of a number of heterogeneous sensors, and by costs associated with the transmission and processing of the data [9, 17]. There are various approaches to saving energy in these situations, such as the use of ultra-low power sensors. Considering the energy cost of data processing, approaches related to the sensor data itself have also been proposed, exploiting the redundancy in sensor systems to reduce the sampling rate, and hence power consumption, of the sensors themselves. Energy can be saved in this way even on a per-sensor basis [1]. It has also been demonstrated that a node may be able to accurately predict its samples based on knowledge of the readings of sensors around them [7, 12].

The problem of energy efficiency can be considered at a higher level than individual per-sensor optimisation particularly when context is the final output of the system (rather than in an application where numerical data is the final

system output, with context awareness purely a means of saving power). In a context-aware sensor system, selective activation of sensors according to the contexts to be sensed significantly reduces the energy costs of transmitting data [11]. This approach can be extended to make use of prediction in adjusting sensor configurations to be best suited to sensing the most likely next contexts [9]. Research has largely focussed on the communication energy costs of the data, rather than the sampling costs, although these costs are considered in [2]. This is not based on learnt prediction of contexts, however.

This paper proposes an architecture for context awareness which uses prediction of future contexts to minimise the energy required in sensing them. This is achieved by balancing the costs of sampling from each sensor against the improvement in context identification confidence expected. To the best of the authors' knowledge, this is the first work which balances these two factors to minimise energy expended.

## 2 An architecture for energy-efficient context recognition

An architecture is proposed for determining the current context, achieved by means of an arbitrary number of sensor inputs. This is illustrated in Figure 1. The architecture proposed is broadly applicable to any situation where there is a need to understand the user context by association with past contexts, rather than classifying user context into a predetermined set. A secondary, essential aim is for the context recognition to be performed in an energy-efficient manner.

The proposed architecture uses machine learning to identify contexts and adapt sampling rates. First, features are extracted from the raw data. These abstractions, such as averages of raw sensor streams, transform the data to such a level as to be more readily and informatively classified. Next, these features are grouped meaningfully, the groupings representing the user contexts. At this point, the current context can be identified based on sensor input data. However, the final context is also further used in tuning the sensing process to improve energy efficiency. Using the history of movements between the recognised contexts, predictions can be made of future context changes. Based on this, sample rates of the various sensors can be adjusted so as to maximise the sensitivity of the system to the most probable next contexts, thus saving energy for minimal loss in accuracy of context classification.

### 2.1 Feature extraction

The first stage of the system performs feature extraction. This reduces the computational requirements of machine learning by providing a relatively low-dimensional representation of the raw sensor data. This is guided by knowledge of the specific domain of interest: examples consist of mean sound level for a frequency range being extracted from audio data,

or acceleration data being filtered into frequency characteristic bands.

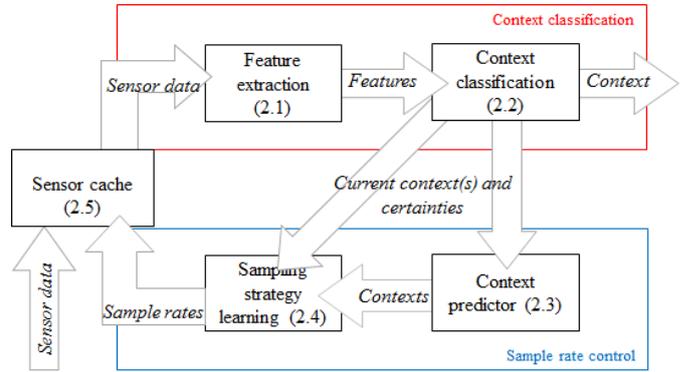


Figure 1: architecture for context recognition. Modules are described in detail in the numbered sections.

Consideration must be given to the utility of each feature introduced to the system. A common approach to this problem is to generate a wide range of features, and to then allow the system to select those which are most helpful in classifying the data [13]. This approach is used in this architecture, as covered in the next section.

### 2.2 Context classification

Once features have been extracted from the raw sensor data, context classification is performed. In the system, a number of classifiers are trained against existing data. This learning is an unsupervised process. Classifiers are trained to distinguish between every possible combination of pairs of contexts. In turn, for each combination, multiple classifiers are produced, one for each possible pair of features. This is a *one against one* style of classification as opposed to the alternative *one against all*, in which a single classifier is trained to distinguish all possible contexts. The use of multiple classifiers for different combinations of features means much simpler algorithms can be used in evaluating new data against the classifiers. The relationship between classifiers, contexts, and features is illustrated in Figure 2.

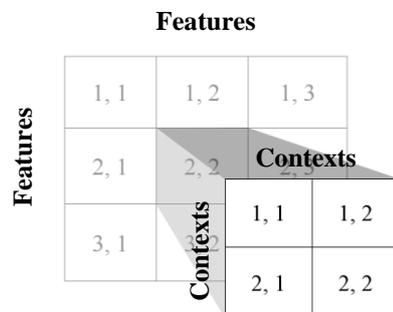


Figure 2: four classifiers are shown in the foreground, each for separating two contexts. All four use the same pair of features.

The algorithm which is used to generate the classifiers returns the data grouped into a number of clusters. By training with data which expected to contain precisely two contexts, these two clusters can then be assumed to represent these two contexts. Once all the classifiers have been trained, the ‘feature quality’, or utility of each pair of features in differentiating between the two contexts, can be calculated.

After training, new incoming sensor data can be processed and the likely current contexts identified. For each new item of sensor data, a feature vector is generated by the feature extraction module. This point is then fed into the classifiers, and the winning context in each is decided by geometric distance between the context cluster centres and the new data points. The results from the numerous classifiers are combined with a voting system. Each classifier returns one of its two possible contexts, whichever is most likely. Classifiers which are able to return correct result (that is, those which were built from the training data for the context in question) will reliably return a vote for this context, while those incapable will return votes for other contexts at approximately chance level. Hence, finally, a ranked list of likely current contexts is returned, with the most probable at the top. This output can be used by the context-aware application at this point, and represents the useful intended output of the system. A confidence metric is produced by consideration of the distribution of votes.

### 2.3 Context prediction

Based on the contexts identified in the new sensor data by the classification stage, a Markov chain is generated, representing the probability of a transition from any one context to any other. With the understanding of contexts provided by the Markov chain, it is then possible to prioritise the importance of data from different sensors. In the situation where there is only one possible next context, for instance, and where the difference between the current and next contexts is definitively represented by a change in sound level, only the microphone needs to be sampled.

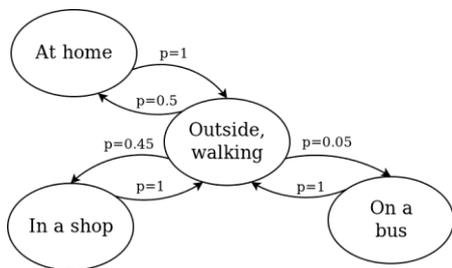


Figure 3: a Markov chain for some example contexts

In practice, the next context is unlikely to be certain. Instead, a ranked list of the likely next contexts is constructed initially.

### 2.4 Adaptation of sampling strategy

For each possible future context, a normalised vector is generated which represents the reliability of the different classifiers at distinguishing between this and the current

known context. This is based on the assessment of each classifier quality produced in the training phase. Based on these scores in conjunction with a table representing the cost of each feature, the features most suitable for correctly recognising the change to the next context can be selected. The scores are ranked and normalised such that the most effective and economic classifier has a score of 1, and the least, 0.

Note that the context classifications are based on features, not on the raw sensor data. As a result, these rankings of the significance of different aspects of the classification data relate to the importance of features, not sensors. A mapping is hence required from the ranking of features to the ranking of importance of the underlying sensors. Once a normalised vector list of sensor importance rankings has been generated, their sampling rates are adjusted accordingly by the sensor cache.

### 2.5 Updating sensor cache

Based on the sensor significance vector from the sampling strategy stage, the sample rates of the sensors are updated. This is done by multiplying the maximum sample rate of each sensor by the sensor score. The most important sensor will take readings at its maximum rate, and any which are scored zero will not be used at all.

## 3 Case study: a wearable camera

The motivation for the system in this paper is the need to detect the current context in a wearable camera [3]. The camera takes photos automatically based on the user’s judged level of interest in the current context. Photos are then transmitted over Bluetooth in real time. Capturing and transmitting photos is an expensive operation, and so the battery life is a pressing concern. By making intelligent decisions about when to capture photos, power is saved by taking the right number of photos for maximum benefit to the user; by making these decisions themselves in a power-efficient way, still more power can be saved.

### 3.1 Simulation of the architecture

The proposed architecture was simulated in MATLAB to evaluate the effectiveness. This has various benefits, such as making evaluation of new algorithms much more consistent, as a ready-captured data set can be re-used. The data set used was collected by means of a mobile sensor system worn around the neck for some hours; a subset of this data was selected by combining selected ranges of the full data set. Ultimately, contexts will be detected using unsupervised methods, but for the purposes of evaluation, supervised methods are used. The selection of regions of this subset was based on the crossover between features currently developed for the sensor data, and manually marked contexts. For example, as a feature was available for determining the light level, outdoor and indoor contexts were selected to form part of the test data.

Results were collected by means of five-fold leave-one-out cross-evaluation. The collected data set is divided into five parts, with four used for training and one for evaluation; the classifiers are trained on the former and their performance evaluated on the latter. The process is then repeated using each part as the evaluation subset. The whole data set is manually ground-truthed for the purposes of evaluation.

### 3.2 Feature extraction performance

The features used were based on two data sources: captured photos, and an accelerometer. The extracted features comprise, in full, the average red, green, and blue levels of the photos, the mean of these (or whiteness), the automatically-set ISO of the camera in each case (essentially a light level) and the dominant frequency detected by the accelerometer. An example of the features extracted from raw data is shown in Figure 4.

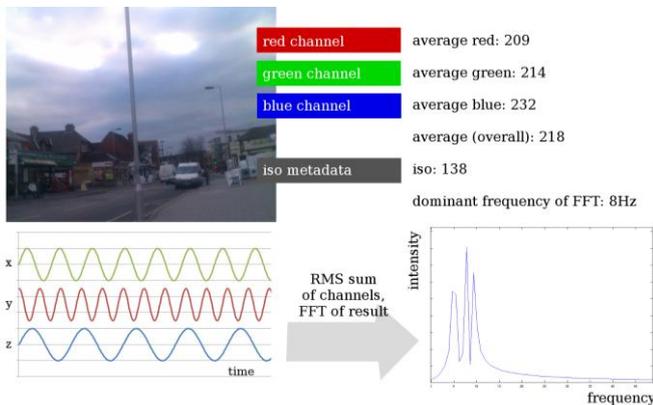


Figure 4: raw sensor data, and extracted features

Four contexts were used in training the classifiers: *asleep*, *cycling*, *in office*, and *walking outside*. For each pair of contexts, one classifier was produced for every possible pair of features.

### 3.3 Classification and adaptive sampling scheme

The learning algorithm used for training the classifiers was k-means, with the number of clusters set at two. The importance of each pair of features for a given pair of contexts is given by the quality metric produced at the time the classifier is constructed.

For a given list of probable next contexts, the mapping from a set of classifier quality metrics to a ranked sensor importance is performed by taking the average importance ranking of each feature in the classifier set. Then, based on the individual feature significances, the sensor significances are given by averaging the significances of all features derived from that sensor's data. In this evaluation, the average importances of two sensors are given: photo, and accelerometer. The quality is derived from a confusion matrix, and is calculated such that each classifier has a quality rating between 0 to 1, where 1

indicates perfect separation of the contexts into different clusters, and 0 represents performance at chance.

## 4 Results

By taking the average of all quality scores for individual classifiers, a measure is calculated for how well different pairs of features perform at separating different contexts. For instance, the classifiers based on comparing the average red and green features worked very well at distinguishing between sleep and being in the lab, with a separation of 91%. Meanwhile, classifiers based on the average photo green level and accelerometer dominant frequency features were 76% separated, again averaged across all contexts. Overall performance was more modest, at an average of 43%. It is clear, therefore, that features must be considered in conjunction.

### 4.1 Context recognition performance

An evaluation feature sequence was constructed using data from three of the four trained contexts. By running this sequence through the full collection of classifiers, the context at each point in time was identified, according to the machine learning system. Table 1 shows a confusion matrix showing the ground-truth and classified contexts. The overall accuracy in this evaluation, per classification, was 74.3%.

Ground truth context	Classified context			Accuracy
	Sleep	Office	Cycling	
Sleep	15	0	0	100%
Office	3	33	6	79%
Cycling	2	7	2	18%

Table 1: confusion matrix for contexts for one k-fold pass, with ground truth in rows and classification in columns.

The most difficult context to identify is clearly *cycling* with the current features: in fact, classification of cycling performs worse than at chance, persistently misclassifying in the *office* context. The features used at present are clearly unable to capture the key differences; this shows how important domain knowledge can be in designing features.

### 4.2 Context prediction performance

Taking the results from the classification stage, a Markov chain was then constructed. This is compared to a chain for the ground-truth data, both shown in Table 2. Discrepancies between the two chains are a result of imperfect classifications in the previous stage. Note that these chains consider context transitions only, so the diagonal will always be zero.

Current context	Next context		
	Sleep	In office	Cycling
Sleep	0	0.83 (1)	0.17 (0)
In office	0.40 (0.67)	0	0.60 (0.33)
Cycling	0.29 (0)	0.71 (1)	0

Table 2: Markov chain from context recognition data, with values from ground truth in brackets

### 4.3 Adaptive sampling performance

With the recognition and prediction stages in place, evaluation of the adaptive sampling performance is then possible. In initial experimentation, this is limited to sequences of contexts which do not repeat. The Markov chain used for this part is therefore simplified over that generated in the previous section. The most useful features for recognising the next context are identified as described in section 3, and sensor sampling schedules are sent to the sensor cache. The photo data was consistently ranked first in all cases, and so was always sampled at the full rate possible. The accelerometer sampling rate varied between 40.2% and 64.8%, depending on the training subset used. The relative performance was 85.1% accuracy compared to classification at the full sampling rates.

Furthermore, by manually setting the sensor sampling rate lower for all sensors, the performance of the context identification stage can be assessed at combinations of rates which the current mapping between feature and sensor importance does not reach. This is summarised in Table 3, and shows that there are still considerable further energy savings possible for a more complex mapping between feature significance and sensor sampling rates. For the camera module in the case study using 3.4J per photo at a maximum rate of two photos per minute, and accelerometer sampling at 20Hz [3], savings of 109mW (compared to 114mW at full sampling rate) are possible for only an 8% reduction in classification accuracy.

Controlled sampling rate	100%	40%	20%
Relative accuracy	100%	92.2%	51.3%
Energy use of camera	113mW	4.52mW	2.26mW
Accelerometer energy	0.77mW	0.31mW	0.16mW

Table 3: relative classification accuracy at different sampling rates

## 5 Conclusions

This paper has proposed an architecture for detecting context through machine learning. This architecture adapts the sampling rates of the underlying sensors according to the context sensed, thereby maximising energy efficiency. The architecture has been evaluated in MATLAB on a selection of data, and has achieved good results in recognising contexts, with average accuracy over five-fold cross-validation of 74.3%. With the output from the context classifiers, Markov models were constructed which described the probability of transitions between contexts. An adaptive sampling scheme has been demonstrated, with recognition performance evaluated at a range of different sampling rates, showing energy savings in sensing of around 60%, with only an 8% reduction in context classification accuracy. In future work, the system will be tested with a larger data set, and semi-supervised and unsupervised approaches will be evaluated.

## References

- [1] C. Alippi, G. Anastasi, M. di Francesco, M. Roveri. "Energy management in wireless sensor networks with energy-hungry sensors", *IEEE Instr. and Meas. Mag.*, **12**, pp. 16-23, (2009).
- [2] C. Alippi, G. Anastasi, M. di Francesco, M. Roveri. "An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors", *IEEE Trans. Instr. Meas.*, **59**, pp. 335-344, (2010).
- [3] D. de Jager, A. L. Wood, G. V. Merrett, et al. "A low-power, distributed, pervasive healthcare system for supporting memory", *Proceedings of MobiHoc '11*, (2011).
- [4] A. K. Dey, G. D. Abowd. "Towards a better understanding of context and context-awareness", *Lecture Notes in Computer Science*, **1707/1999**, pp. 304-307, (1999).
- [5] A. Dey, G. Abowd. "CybreMinder: a context-aware system for supporting reminders", *Proc. HUC2K*, pp. 172-186, (2000).
- [6] Y. Ding, H. R. Schmidtke, M. Beigl. "Beyond context-awareness: context prediction in an industrial application", *Proceedings of UbiComp '10, adjunct*, (2010).
- [7] E. Elnahrawy, B. Nath. "Context-aware sensors", *Wireless Sensor Networks*, Springer, **2920**, pp. 77-93, (2004).
- [8] H. W. Gellersen, A. Schmidt, M. Beigl. "Multi-sensor context awareness in mobile devices and smart artifacts", *Mobile Networks and Applications*, **7**, pp. 341-351, (2002).
- [9] D. Gordon, S. Sigg, Y. Ding, M. Beigl. "Using prediction to conserve energy in recognition on mobile devices", *Proc. PerCom '11 workshops*, pp. 364-367, (2000).
- [10] A. Jain, E. Y. Chang. "Adaptive sampling for sensor networks", *Proceedings of DMSN '04*, pp. 10-16, (2000).
- [11] S. Kang, J. Lee, H. Jang, et al. "SeeMon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments", *Proc. MobiSys '08*, pp. 267-280, (2008).
- [12] J. Kho, A. Rogers, N. R. Jennings. "Decentralized control of adaptive sampling in wireless sensor networks", *ACM Transactions on Sensor Networks*, **5**, pp. 19:1-19:35, (2009)
- [13] V. K on onen, J. M antym arvi, H. Simil a, et al. "Automatic feature selection for context recognition in mobile devices", *Pervasive and Mobile Computing*, **6**, pp. 181-197, (2010).
- [14] M. Meiners, S. Zaplata, W. Lamersdorf. "Structured context prediction: a generic approach", *Distributed Applications and Interoperable Systems*, Springer, **6115**, pp. 84-97, (2010).
- [15] F. Michahelles, S. Karpischek, A. Schmidt, "What can the Internet of Things do for the citizen?", *IEEE Pervasive Computing*, **9**, pp. 102-104, (2010).
- [16] B. Pietropaoli, M. Dominici, F. Weis. "Multi-sensor data fusion within the Belief Functions Framework", *Proceedings of NEW2AN '11*, pp. 123-134, (2011).
- [17] J. Podpora, L. Reznik, G. V. Pless. "Intelligent real-time adaptation for power efficiency in sensor networks", *IEEE Sensors Journal*, **8**, pp. 2066-2073, (2008).
- [18] A. D. Wood, A. J. Stankovic, G. Virone, et al. "Context-aware wireless sensor networks for assisted living and residential monitoring", *IEEE Network*, **22**, pp. 26-33, (2008).