Noname manuscript No.
(will be inserted by the editor)

Unobstrusive human activity recognition using smartphones and Hidden Markov Models

Adriana Wilde · Robert Streeting · Ed Zaluska

Received: 12 June 2013 / Accepted: date

Abstract Accelerometer data is sufficient to compute human activity recognition, even with only a single accelerometer in use. Such data can be used for many pervasive computing applications, user activity being interpreted as real-time contextual information. This paper investigates activity recognition on smartphones, as they are a suitable platform for the implementation of context-aware pervasive systems. Many machine learning algorithms are suitable for this purpose, but Hidden Markov Models (HMMs) are particularly appropriate for their ability to exploit the sequential and temporal nature of data. This paper evaluates HMMs in unobstrusive activity recognition with the added restrictions resulting from the use of the smartphone platform.

Keywords Human activity recognition · accelerometer data · pervasive computing · Hidden Markov Models.

1 Introduction

Pervasive (or ubiquitous) computing is a post-desktop computing model which seeks to enrich user experience with technology supporting their everyday lives, whilst doing it in an unobstrusive manner, i.e. "fading into the background" [1]. It spans three main areas of concern: natural interfaces, context-aware computing and automated capture/access of data [2]. *Natural interfaces* aims to minimise the obtrusiveness of the interaction with devices, *context-aware computing* aims to interpret the user's context so that systems can adapt to it, and *automated capture/access of data* aims to minimise human intervention in the process.

A. Wilde, R. Streeting, E. Zaluska
Electronics and Computer Science, University of Southampton, United Kingdom.
Tel.: +44-23-8059-8787, +44-23-8059-3379
E-mail: {agw106, rsws1g09, ejz}@ecs.soton.ac.uk

Smartphones are mobile devices which support significant processing power together with a variety of sensors in addition to their communication capabilities. As an example of their ability to be context-aware, many smartphone applications use location to provide contextual information. Similarly, other sensors commonly included in smartphones, such as accelerometers, could be used to acquire aspects of user context. We have previously shown that using data from a single tri-axial accelerometer can be used to recognise user activities accurately [3]. However, the choice of the most suitable classification algorithm suitable for smartphones still remains an open question.

Hidden Markov Models (HMMs) have been previously chosen for activity recognition because their ability to exploit the sequential nature of activities. An evaluation of HMMs both in terms of their suitability as activity recognition classifiers and the constraints on the smartphone platform is necessary to assess their suitability for context-aware pervasive systems. In particular, the additional constraints facing smartphone software (user interfaces aside) are *energy consumption* and *memory usage*, in addition to the typical contraints inherent to all pervasive computing applications. These include *sensor constraints*, such as the need to overcome the limited sensing capability available (i.e. a single accelerometer) and the possibility that the orientation and position of the device is not fixed. In addition to these smartphone-specific issues, *real-time processing* is desirable, as contextual data may lose its relevance to a software agent if it takes too long to process. Finally, the system should have an acceptable *classification accuracy.*

We consider all of these issues in our evaluation of HMMs for activity recognition in ubiquitous computing using smartphones, starting from a review of HMM-based methods from the literature that tackle each of the individual problems faced in this domain, together with some preliminary experiments.

Section 2 contains a brief introduction to HMMs. We then explore the relevant issues to consider when using HMMs in this context, namely: energy consumption (Section 3), memory usage (Section 4), accelerometer constraints (Section 5) and real-time processing requirements (Section 6). Then, we evaluate classification accuracy (Section 7) across the literature. To complement this review, Section 8 discusses preliminary experiments conducted in order to investigate less obvious issues. Section 9 summarizes the feasibility of HMMs in this context. Finally, Section 10 provides concluding remarks and directions for further work.

2 Hidden Markov Models

Hidden Markov Models (HMMs) can be considered as a set of states which are traversed in a sequence hidden to an observer. The only thing that is visible is a sequence of observed symbols, emitted by each of the hidden states whilst they are traversed. The model defines a probability of moving between states (the *transition* probability) and the probability of each state emitting each particular symbol (the *emission* probability). Using this model and an observed sequence of symbols, a system can determine three key things: the probability of the observed symbol sequence, the most probable hidden state sequence, and the parameters of the model that maximise the probability of the observed symbol sequence [4]. The following algorithms are used to make these inferences:

– **Forward-Backward Algorithm** - In this context this is used for determining the probability that an emission sequence was generated by a given HMM. The forward pass computes the probability of being in a state at a particular time given the observation sequence up to that time, so summing over all states at the end is required.
– **Baum-Welch Algorithm**[1] - Used for estimating transition and emission probabilities of a HMM given an observation sequence and initial "guesses" for these values. As an *expectation-maximisation* algorithm, it uses an iterative search for the parameters with the highest likelihood.
– **Viterbi Algorithm**[2] - Used to find the most likely sequence of hidden states given a HMM and an observation sequence. It computes the recursive likelihood of being in each state at the next time step until the end of the sequence, at which point, the algorithm backtracks to give the most likely state sequence.

3 Energy consumption

Due to the limited battery space available in a mobile device, reducing energy consumption is one of the most important considerations when developing smartphone software. It is therefore relevant to evaluate the feasibility of using HMMs when energy usage has to be kept to a minimum.

Pathak et al. [6] give a taxonomy for what it labels as *ebugs*; bugs in software that cause excessive energy drain. Although for evaluation purposes we can assume that the HMM implementation would be bug-free, this taxonomy does identify the leading causes of energy drain in mobile applications. Many of the classes of ebugs presented relate to applications over-using hardware components which use a significant amount of energy to run. Of particular interest are *no-sleep* bugs, which involve keeping processes "awake" for longer than they need to be, suggesting that long-running passive processes are likely to be a problem. Unfortunately, pervasive agents need to be constantly active to access contextual data and make decisions in real-time, regardless of the machine learning technique being used to classify activities.

Boyd and Sundaram [7] present a conceptual framework for activity recognition that claims to reduce energy usage. The framework defines a passive *sleep* state where the system remains running but uses as little energy as possible to listen for new activities, and then an *active* state where the activities are actually classified. New activities are found using *transition detection*,

[1] The forward-backward algorithm is sometimes referred to as the Baum-Welch algorithm; the term here is used as in Rabiner and Juang [4].

[2] Viterbi decoding algorithm, not to be confused with the training algorithm.

which simply looks for a change in activity. The research shows that this can be achieved with a very low sample rate on the accelerometer, reducing the amount of energy used. When a transition is detected the system enters the active state, where HMMs are used to classify the activity at a high sample rate. When the activity has been determined, the system goes back into a sleep state.

Although HMMs are used for activity recognition in this particular example, this method should apply to any machine learning algorithm because the method of saving energy is independent of the machine learning technique used. This result would suggest that HMMs are not significantly better or worse than any other machine learning algorithm when it comes to energy consumption, assuming this method were to be used. If there are differences, they would be made insignificant by the amount of energy saved using the sleep state, which effectively removes the need to run the algorithm continuously and reduces the need to access the accelerometer.

4 Memory usage

Given the limited hardware available in a smartphone, reducing memory usage as much as possible has a wide range of benefits. In order to analyse the effectiveness of HMMs in this respect, the most relevant factor is the *space complexity*.

The space complexity of the forward-backward and Viterbi algorithms are both about $O(NT)$, where N is the number of states and T is the length of the state sequence [8]. This can be confirmed intuitively by considering the storage requirements for a HMM. In both algorithms, a matrix is built up that stores the probability of either the state sequence or the observed sequence at each state at each point in time. The columns of the matrix represent the time steps in the sequence and the rows represent the states. This matrix therefore contains $N * T$ elements, and so the space complexity of the algorithms is $O(NT)$. This is about the same as the K-Nearest Neighbours algorithm (the simplest of the effective algorithms explored in Wilde [3]), suggesting that HMMs are as space conservative as possible for a machine learning algorithm.

Lee and Cho [9] argue that a hierarchical HMM reduces the computational complexity of the process. This method involves classifying short segments of accelerometer data using one layer of HMMs, then feeding the results of these individual classifications into another high-level HMM which can classify more complex activities as compositions of low-level ones. This divide-and-conquer approach avoids having to deal with large amounts of state transitions - if a single-layer HMM was used, multiplying many probabilities together eventually results in arithmetic underflow which means that the data has to be transformed using a log function. Instead, small and simple HMM sequences are processed and none of these issues are present.

Although hierarchical HMMs do not improve the worst case space complexity, they reduce the length of the sequence and the amount of hidden states due to the division of the problem. After the result for one level of HMMs is obtained, the memory used to obtain that result can be cleared, meaning that the amount of memory needed is only the amount needed by the most complex sub-model.

The method described in Section 3 reduces energy consumption in two ways: reducing the amount of access to the accelerometer and reducing the amount of processing that has to be done. Therefore, it can also be applied to memory consumption, and the conclusions made can be carried over; slight differences in the memory used by different machine learning algorithms are not significant next to the amount of memory freed by "switching off" the algorithm when it is finished.

Overall, the conclusion is that HMMs are viable for use on smartphones in terms of memory usage. Even before the application of extra techniques, they have a relatively small space complexity.

5 Sensor constraints

5.1 "Near-static" and similar activities

As discussed by Quwaider and Biswas [10] and demonstrated by Kunze and Lukowicz [11], accelerometer data struggles to classify accurately what are termed "near-static postures" (e.g. standing or sitting); this is intuitive because the lack of movement means that the accelerometer signal remains very similar over a range of static activities. This issue is not just limited to near-static activities however; these can also become confused with other low-energy activities (e.g. writing on a chalkboard). Additionally, similar dynamic activities can be also be confused (e.g. walking and going up stairs) [3][9][12][13]. Distinguishing between similar activities using accelerometer data is still clearly a difficult issue to resolve.

Li et al. [14] presents a method that can solve this issue by making use of the ability of the HMM to classify sequences. Activities are represented as hidden states, discretised features obtained from the accelerometer signal are the observed sequence of symbols. Given this sequence, it is possible to calculate the most likely hidden state sequence (and therefore sequence of activities) using the Viterbi algorithm. The actual algorithm used in this method is a slightly modified version of this algorithm that caters for the fact that activities can last for a long time.

As this method classifies sequences of activities rather than single activities, that means that the transitions between activities are taken into account. This allows static activities and similar activities to be classified with more accuracy because the previous and following activities are considered; for example, if a static activity comes after a walking activity, it can be inferred that it might be a standing still activity. This ability to classify sequences is unique to HMMs when compared to other common machine learning algorithms, so clearly they are well suited to activity recognition using just accelerometer data in this respect.

5.2 Accelerometer positioning

In addition to limiting the amount of sensors available, using a smartphone platform also means that certain assumptions can be made about the position of the accelerometer on the body. For example, it may be assumed that most of the time the device is kept at the hip. However, there are obvious exceptions to this; users also may keep their mobile phone in a handbag or in a loose pocket [15]. This means that no assumptions can be made about the orientation of the device, which may change the accelerometer signal significantly enough that activities become unclassifiable.

Kunze [16] provides a detailed analysis of this problem and suggests multiple solutions for tackling its individual components. The main problem areas are detecting whether the device is on the body at all and detecting where the device is placed on the body. Over time, the system can also detect changes in the orientation of the device and displacement of the device whilst remaining on the same body part. Detecting where the device is placed on the body and its orientation only requires the accelerometer, making it feasible for use in modern smartphones. However, other parts of the solution require other sensors such as a gyroscope.

Using these solutions, it would be possible to classify the body part on which the device was present and therefore use different HMMs based on this information to improve the classification accuracy. Not only this, but the accelerometer position can be used as further contextual information for activity classification - for example, if the device is strapped to the wrist or arm, it might be inferred that the user is more likely to be doing physical activity [16]. Orientation invariance can also be achieved by estimating the device's orientation using the gravitational ('static') acceleration from the accelerometer data, as described further by Mizell [17].

Because the accelerometer position and orientation can be obtained from the data separately to the current activity, this means that it causes no further constraints on the activity recognition process. HMMs are therefore completely feasible for the problem regardless of the accelerometer positioning issue, assuming that the accelerometer position is collected in this way.

5.3 Limited hardware

It has been shown that accelerometers positioned at different points on the body provide different information. As discussed by Kunze [16], some activities are associated with certain body parts, meaning some activities are unclassifiable when the accelerometer is positioned in a particular place, regardless of whether the accelerometer position is known or not. However, with multiple accelerometers in different places this problem can be overcome. Limiting activity recognition systems down to the hardware commonly available in smartphones, realistically only one accelerometer is available because other sensors placed around the body are too obtrusive to be an effective pervasive system [13][18].

However, it has been shown that a single accelerometer is enough to detect a range of activities in various research, regardless of the machine learning technique [3][18][9]. This suggests that this issue is non-existent for the majority of activities, meaning that HMMs are no better or worse than other techniques in this regard. For the activities that are not classifiable, there could be benefits for using different machine learning techniques.

As shown before, the main benefit of HMMs is their ability to analyse sequences, and therefore compute the most likely activity occurring at a certain time without having quality data for that activity. The technique discussed by Li et al. [14] can be applied here in a similar way to the way it was applied for static activities. If most of the activities in a sequence are classifiable, then the unclassifiable gaps can be filled by looking at the most likely activity that fits in the sequence. In this regard, HMMs have the potential to overcome this issue where other machine learning algorithms may not be able to. However, there are other issues with this approach which are discussed in the next section.

6 Real-time processing

In the context of pervasive systems, it is important for any contextual data to be processed immediately as it may become irrelevant later on; for example, if the goal is to detect an elderly person falling over to alert the health services, it is important that this activity is classified when it happens as opposed to later on. This suggests that real-time classification of activities is a required ability of any machine learning algorithm that is being used for this purpose.

Although the method previously proposed by Li et al. [14] shows how HMMs can be used to detect certain kinds of activities that cannot be detected as easily by other methods, it is not ideal for real-time processing. Since an entire observed sequence of activities is necessary before classification can begin, by the time the activities have been recognised the data may no longer be relevant [19]. The actual method used involves segmenting the data into parts on which to apply the Viterbi algorithm separately; although this is not exploited for real-time use in the research, it could make the latency between data collection and classification shorter. A similar method in principle is described in more detail by Eickeler and Rigoll [20]. However, the shorter the sequence becomes, the less benefit the process receives from the extra contextual data. The true nature of this trade-off is a subject for future research.

Although real-time processing is ideal for a pervasive application and essential to many application areas, there are certain activities that do not have this requirement. For example, Chang et al. [21] describes the use of the Viterbi approach for tracking free-weight exercises at the gym after a workout, suggesting that real-time processing would be beneficial in this case, but not necessary. Yang [22] shows how the Viterbi approach can be useful as a post-processing method to correct errors made by other classifiers, showing that it is possible to allow for less accurate real-time classification in the short term and correct any errors made later on.

An alternative method widely used with HMMs for activity recognition makes use of the forward algorithm for determining the likelihood of the observed sequence (the accelerometer data) [23][24][7][13]. A separate HMM is trained for each individual activity (using the Baum-Welch algorithm) off-line and the likelihood of the observed sequence of data for each activity is calculated in real-time. The activity that is chosen is the one represented by the HMM that produced the highest likelihood for the given sequence of data. In order to do this, the data has to be split into overlapping *sliding windows*; each window is then classified individually as it is collected.

Although this shows that HMMs promise to deliver in a real-time pervasive system, there are a number of disadvantages to this approach. The clearest issue is that classification of static/similar activities is no longer possible as it was in the Viterbi approach, so it seems that the ability to classify these types of activity and the ability to process in real-time are mutually exclusive in this respect. This approach also uses multiple HMMs as opposed to just one, all of which have to be run for each sliding window of data. This means that the amount of processing required increases with the array of activities that need to be recognised. Finally, the amount of hidden states needs to be decided, and it is not immediately obvious how to do this; this is implicit with the Viterbi approach because each state represents an activity, but the mapping is not quite as direct in this context.

Processing in real-time also has many of its own inherent issues. As mentioned previously, long-running processes are a major cause of energy drain, and constant processing can put an unnecessary load on memory. Boyd and Sundaram [7] showed that these issues can be solved without the need for compromising the real-time ability of the system by a significant amount. Amma et al. [24] also explored the benefits of attempting to “spot” useful data in the feed before processing it with HMMs, and gives further evidence that this is a feasible approach for processing data in real-time.

Overall, it is evident that there is a compromise between classification accuracy and scope, and feasibility of real-time processing. The segmented Viterbi approach allows the classification of static and similar activities and arguably increases the accuracy due to the consideration of inter-activity transitions, but the larger the segments are the greater the latency between the activity and its classification. The HMM-per-activity approach allows the minimal latency, but lacks the scope of the Viterbi approach. Therefore real-time classification with HMMs is a possibility, but a decision has to be made on how far to sacrifice scope and accuracy of activity classification.

7 Classification accuracy

Accuracy of classification is an important issue for all machine learning algorithms, although it still needs to be considered alongside the domain-specific issues with activity recognition on smartphones. The classification accuracy is the proportion of correct classifications made by the system over the entire data set. Although accuracy is a used as a standard measure of how well a machine learning system is performing, it must be interpreted carefully. If the test sample is too small or not comprised of unseen data, the accuracy can be skewed up or down from the actual value.

Most of the methods presented so far have been practically tested and the classification accuracy reported. These results are summarised in Table 1.

Overall, the research for all of the methods show impressive accuracy, with most giving a 90%+ accuracy. However, many of the experiments carried out also used very small sample sizes of participants from which to gather data, which means the accuracy could be worse than it appears. Other issues found included accelerometers placed in unconventional positions (the effects of which are explored in Section 5.2), incomplete results (missing data), parameters that are chosen without any explanation (arbitrary parameters), and use of extra sensors in the classification such as gyroscopes or additional accelerometers. Only Lee and Cho [9] attempted to classify high-level activities (such as riding a bus or going shopping), and it is assumed that this is the reason for the reduced accuracy observed.

Despite the issues presented for each of the methods, overall HMMs appear to be a satisfactory solution in terms of the recognition accuracy alone, with regard to low-level and high-level activities. However, Cao et al. [25] suggests that HMMs may not be well suited to activity recognition in terms of accuracy of recognition. Along with a non-sequential framework devised by the researchers and *Conditional Random Fields* (a generalisation of HMMs), HMMs are tested on three activity datasets, each of which have distinctive features. The first contains data for 18 different activities, the second contains data for these activities and a large amount of data for a null activity, and the final dataset has four activities, which is more typical of the experiments shown so far. Although the HMM performs fairly well on the final dataset, it struggles with the other two, suggesting that the main issue is the large amount of classes present. Even though the data considered is not limited to accelerometer data, given the nature of the problem it appears that it may be an inherent issue with the algorithm rather than the type of data. Unfortunately the actual classification accuracy is left out of the results given, but it can be seen that the non-sequential framework is clearly more accurate and more robust in this case.

If it is assumed that HMMs are limited to a small number of activities, there are ways to improve the classification accuracy even further. Mannil et al. [19] provides a method for reducing the false positives given by attempting to eliminate data containing activities that were not seen in the training phase. This achieves a similar result to the "spotting" stage seen in Amma et al. [24]

Table 1 Experimental conditions and accuracy of existing HMMs

Experimental Conditions	Segmented Viterbi [14]	HMM-per-activity [23]	HMM-per-activity [13]	Energy-saving [7]	Hierarchical HMM [9]	Filtered Viterbi [19]
Hip-mounted sensor	-	X	X	-	-	X
Missing data	-	X	-	-	-	-
Small participant sample size	-	X	-	-	X	-
Arbitrary parameters	-	X	-	-	X	-
Use of gyroscopes	-	-	-	X	-	X
Multiple accelerometers	-	-	-	-	-	X
High-level activity detection	-	-	-	-	X	-
Experimented with smartphone	-	-	-	-	X	-
Recognition accuracy(%)	99.6	99	98.7	94.4	83.5	94

but with the purpose of reducing false positives in dense activity data rather than identifying classifiable data amongst meaningless background data. The main algorithm is based on a segmented Viterbi approach like Li et al. [14], but implements an additional "sanity check" after each second of data. At the beginning, a threshold is calculated using sample data that is used during the sanity check to decide whether data relates to a known activity or an unknown activity, and if unknown the data is classified as such. This prevents the Viterbi process from trying to classify it and prevents false context from arising in the sequence, thereby improving the accuracy all round. The paper also gives a method for dealing with varying length activities, which simply normalises the probability based on the duration of the activity. The reason this is necessary is due to the nature of the Viterbi process, which causes shorter sequences to have higher probabilities.

Overall it appears that HMMs can provide a good classification accuracy for a small set of target activities, but it is likely that this does not scale well when the amount of activities increases. This result does not necessarily rule out HMMs entirely; many applications may only have the need to recognise specific events (for example, detecting an elderly person falling or an athlete starting a run). There are also ways to improve the accuracy of HMMs given "real-world" data. Small sample sizes were common within the surveyed literature. This could have artificially increased the classification accuracy beyond levels that could be expected for a real audience. However, there are ways to improve the accuracy if necessary [19], although solutions are not scalable with the number of activities to classify, which appears to be an inherent problem with using HMMs in this context [25].

8 Experimentation

We performed preliminary experiments using Matlab, implementing HMM classification on features extracted from the COSAR data set [3,5]. This allowed the identification of further issues on the use of HMMs for accelerometer data.

Firstly, HMMs are designed to deal with discrete observation data, but accelerometer data is inherently continuous. Therefore, either the data must be discretised or the model must be altered to process continuous data. On the one hand, discretising the data is simple, however, the precision is lost and determining the thresholds for discretisation to minimise errors can prove problematic.

On the other hand, continuous density HMMs (CDHMMs) extend HMMs to allow output of continuous symbols. Rather than having an emission probability for each of a set of discrete symbols, the emission probability is described as a *mixture model* (commonly a Gaussian mixture model). Although this method avoids the loss of precision, it can be more computationally intensive due to the added complexity.

Not emphasised in the literature is that using a large number of features can be detrimental to the performance. As HMMs only deal with one emission per time step, features must be combined into one emission symbol. When many features are used, the space of possible emissions becomes very large. The training data rarely covers this whole space, meaning that during training the probability of emitting many of these symbols is zero. If one of these symbols appears in the test sequence, the probability of the whole sequence becomes zero. This becomes less of an issue with CDHMMs due to the nature of the emission probabilities, making CDHMMs a much more attractive solution where multiple features need to be considered.

The choice between continuous and discrete HMMs is only one of the many choices to be made. For example, HMMs can be *ergodic* (any state can transition to any other state at any time) or *left-to-right* (after leaving a state, it can never be returned to). This choice is mainly based on the complexity of the activity - for example, an activity such as walking that has no grand structure may suit an ergodic HMM. However, an activity such as brushing teeth may benefit better from a left-to-right HMM, with states representing picking up the toothbrush, brushing, spitting etc. which happen in a particular order. The number of distinct hidden states to have is also a key decision.

In summary, there are implementation details that make HMMs less well suited to accelerometer data than other machine learning algorithms. However, for all of these limitations there are also unique advantages to using HMMs, which are presented in the following section.

9 Discussion

With regards to the large majority of issues presented, HMMs appear offer a satisfactory solution on the whole. The smartphone platform does not seem to introduce problems that could reduce the feasibility of HMMs, and it appears likely that they may actually be ideal with respect to memory usage and classifying near-static activities using a single accelerometer. However, in order to work effectively in real-time, the classification scope needs to be compromised partly, but the true nature of this trade-off is not clear at present. Even so, the Viterbi approach (that allows the classification of static/similar activities) can be used as a post-processing method to classify activities that failed to be classified in real-time [22], making it attractive for pervasive computing. It is also difficult to assess the generalisation ability of the system without assessing the classification accuracy on a larger sample of users. Given the research that has been conducted however, the results are promising and indicate that HMMs perform well on real data. The particular machine learning algorithm used is often irrelevant; methods in the literature that reduce power consumption and provide smartphone position/orientation invariance are independent of the algorithm used. In some cases, the HMM is particularly well suited; it has a relatively low space complexity making it ideal for reducing memory consumption, and use of the Viterbi algorithm allows the classification of static/similar activities that would be near-impossible for most other machine learning algorithms. However, these gains have to be sacrificed somewhat in order to process the accelerometer data in real-time. Which of these is more important depends mainly on the particular application domain.

Although HMMs have been shown as suitable for activity recognition, there are nevertheless some limitations. The literature suggests that HMMs may fail when a large number of activities are classified, being more appropriate to applications where only a small set are to be classified. Other considerations include the choice between continuous and discrete HMMs and the need for scalar emissions.

10 Conclusion and Future Work

As noted previously, HMMs are suitable for activity recognition using accelerometer data on smartphones [26], however a number of other issues, such as energy consumption, memory usage and sensor constraints are also important. HMMs have a relatively low space complexity compared to other machine learning algorithms. In addition to this, there are methods aiming to reduce energy consumption which also reduce memory usage.

The constraints imposed by using the smartphone as a platform are mainly the limited hardware and accelerometer positioning. Smartphones today typically include only one accelerometer, which may impair the discrimination between similar activities (e.g. sitting versus standing still) using traditional machine learning methods. The Viterbi algorithm can be more effective in discriminating these activities due to its ability to exploit the sequential context.

This presents a strong argument for using HMMs for activity recognition in preference to other machine learning techniques. However, since the Viterbi algorithm requires a sequence of observations, at least two activities must take place before the algorithm can begin classification, degrading the real-time capability of the system. The forward-backward algorithm can be used to maximise the real-time capability of the system, but the ability to classify static activities is lost. Ultimately there is a trade-off between real-time capability and classification scope; the decision of which is more important is entirely dependent on the application domain.

With regards to the accelerometer positioning, there is evidence that it is possible to obtain the position of the accelerometer on the body and its orientation using only one accelerometer, suggesting that these issues could be solved in pre-processing (before any classification takes place).

Although the focus of this research is to evaluate the use of HMMs with respect to the smartphone platform, more general issues in pervasive computing and activity recognition were also investigated as it is still important that the system performs well in these areas.

Overall, we have established cogent arguments for the use of HMMs in activity recognition – its unique features provide a number of benefits over other machine learning algorithms. This explains its adoption in previous research and also shows that it is a promising contender for use in future pervasive smartphone applications. It has also exposed some of the limitations of HMMs, which will be useful when considering such developments.

However, several unanswered questions remain. Areas of future work include the relationship between the length of the data sequence adopted and the classification accuracy of static/similar activities. This will make it easier to decide whether to use the Viterbi algorithm or the HMM-per-activity approach when designing future applications. The capability of discrete HMMs versus continuous HMMs in this context is also be a valuable subject for future research.

References

1. M. Weiser, "The Computer for the 21st Century." *Scientific American* vol. 265, no. 3, pp. 94–101 (1991)
2. G. D. Abowd and E. D. Mynatt, "Charting past, present, and future research in ubiquitous computing," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 7, no. 1, pp. 29–58 (2000)
3. A. G. Wilde, "Activity recognition for motion-aware pervasive systems," Master's thesis, University of Fribourg, Switzerland (2011)
4. L. Rabiner and B. Juang, "An introduction to Hidden Markov Models," *ASSP Magazine, IEEE*, vol. 3, no. 1, pp. 4–16 (1986)
5. D. Riboni and C. Bettini, "COSAR: hybrid reasoning for context-aware activity recognition," *Personal Ubiquitous Computing*, vol. 15, no. 3, pp. 271–289 (2011)
6. A. Pathak, Y. C. Hu, and M. Zhang, "Bootstrapping energy debugging on smartphones: a first look at energy bugs in mobile devices," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, pp. 5:1–5:6 (2011)

7. J. Boyd and H. Sundaram, "A framework to detect and classify activity transitions in low-power applications," in *Proceedings of the 2009 IEEE international conference on Multimedia and Expo*, ICME'09, pp. 1712–1715 (2009)
8. S. R. Eddy, "Profile Hidden Markov Models." vol. 14, no. 9, pp. 755–763 (1998)
9. Y.-S. Lee and S.-B. Cho, "Activity recognition using hierarchical Hidden Markov Models on a smartphone with 3D accelerometer," in *Proceedings of the 6th international conference on hybrid artificial intelligent systems - Vol. I*, HAIS'11, pp. 460–467 (2011)
10. M. Quwaider and S. Biswas, "Body posture identification using Hidden Markov Model with a wearable sensor network," in *Proceedings of the ICST 3rd international conference on Body area networks*, BodyNets '08, pp. 19:1–19:8 (2008)
11. K. Kunze and P. Lukowicz, "Using acceleration signatures from everyday activities for on-body device location," in *Proceedings of the 11th IEEE International Symposium on Wearable Computers*, pp. 115–116 (2007)
12. S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer, and R. Crompton, "Activity identification using body-mounted sensors – a review of classification techniques," *Physiological Measurement*, vol. 30, no. 4, pp. R1–33.
13. J. Wang, R. Chen, X. Sun, M. She, and L. Kong, "Generative models for automatic recognition of human daily activities from a single triaxial accelerometer," in *Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6 (2012)
14. A. Li, L. Ji, S. Wang, and J. Wu, "Physical activity classification using a single triaxial accelerometer based on hmm," in *Proceedings of the IET International Conference on Wireless Sensor Network, IET-WSN*, Nov., pp. 155–160.
15. F. Ichikawa, J. Chipchase, and R. Grignani, "Where's the phone? a study of mobile phone location in public spaces," in *Proceedings of the 2nd International Conference on Mobile Technology, Applications and Systems*, pp. 1–8 (2005)
16. K. Kunze, "Compensating for on-body placement effects in activity recognition," Ph.D. dissertation, Universität Passau (2011).
17. D. Mizell, "Using gravity to estimate accelerometer orientation," in *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, pp. 252–253, (2003)
18. X. Long, B. Yin, and R. Aarts, "Single-accelerometer-based daily physical activity classification," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2009*, pp. 6107–6110 (2009)
19. J. Mannil, M.-M. Bidmeshki, and R. Jafari, "Rejection of irrelevant human actions in real-time Hidden Markov Model based recognition systems for wearable computers," in *Proceedings of the 2nd Conference on Wireless Health, WH '11*, New York, NY, USA: ACM, pp. 8:1–8:10 (2011)
20. S. Eickeler and G. Rigoll, "Continuous online gesture recognition based on Hidden Markov Models," in *Proceedings of ICPR*, vol. 98, 1998, pp. 1206–1208.
21. K.-H. Chang, M. Y. Chen, and J. Canny, "Tracking free-weight exercises," in *Proceedings of the 9th international conference on Ubiquitous computing, UbiComp '07*, 2007, pp. 19–37.
22. J. Yang, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," in *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics*, ser. IMCE '09, 2009, pp. 1–10.
23. T. Al-ani, Q. T. L. Ba, and E. Monacelli, "On-line automatic detection of human activity in home using wavelet and hidden markov models scilab toolkits," in *Control Applications, 2007. CCA 2007. IEEE International Conference on*, Oct., pp. 485–490.
24. C. Amma, M. Georgi, and T. Schultz, "Airwriting: Hands-free mobile text input by spotting and continuous recognition of 3d-space handwriting with inertial sensors," in *Wearable Computers (ISWC), 2012 16th International Symposium on*, June, pp. 52–59.
25. H. Cao, M. N. Nguyen, C. Phua, S. Krishnaswamy, and X.-L. Li, "An integrated framework for human activity classification," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pp. 331–340 (2012)
26. A. Wilde, R. Streeting, and E. Zaluska, "Using Hidden Markov Models for human activity recognition in motion-aware pervasive systems on smartphones," Submitted to the *7th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM 2013* (2013)