Accuracy and Timeliness in ML Based Activity Recognition

Robert J. Ross and **John Kelleher**
Applied Intelligence Research Group
Dublin Institute of Technology
Ireland

Abstract

While recent Machine Learning (ML) based techniques for activity recognition show great promise, there remain a number of questions with respect to the relative merits of these techniques. To provide a better understanding of the relative strengths of contemporary Activity Recognition methods, in this paper we present a comparative analysis of Hidden Markov Model, Bayesian, and Support Vector Machine based human activity recognition models. The study builds on both pre-existing and newly annotated data which includes interleaved activities. Results demonstrate that while Support Vector Machine based techniques perform well for all data sets considered, simple representations of sensor histories regularly outperform more complex count based models.

Introduction

Although there has been a clear movement from deterministic to probabilistic activity recognition methods in the last 10 years, there remains a lack of consensus on the most appropriate techniques for achieving robust activity recognition across multiple data sets. Naturally the choice of a suitable activity recognition method is primarily centred on the accuracy of activity identification. However, we argue that raw prediction accuracy on isolated datasets is not the only measure of appropriateness which should be employed. For example, in the ambient assistive living domain, the ability to identify missing actions or incomplete activities is essential. Similarly, in order to provide users with assistance in a timely fashion, the activity recognition method must provide an accurate activity description as early as possible in the execution of the activity. We view this issue of *timeliness* as key to online and real-time plan recognition methods. In the case of plan recognition in real situated environments, we also assume that plan recognition methods must be robust both to noise in data and to the partial loss of data.

Given the above difficulties and considerations, in this paper we present a study of the utility of three well regarded activity recognition techniques. We evaluate utility not only in terms of raw accuracy results, but also in terms of the robustness of the techniques to loss of data, and the timeliness of accurate classification responses. The activity recognition techniques which we consider are those based on Hidden Markov Models (Rabiner 1989), Naive Bayes Classifiers (Rish 2001), and Support Vector Machines (Burges 1998). We will begin the main body of this paper with a brief overview of related work in the area of Human Activity Recognition with a particular focus on those efforts which have considered the techniques just mentioned. We then introduce and detail the testbeds that we have made use of in this study. In addition to describing the testbeds in terms of the raw data, we also detail all essential data pre-processing steps and the representation choices which we adopted. Following this, we present the results of our analysis in detail. Finally we draw conclusions and outline proposed future work.

Related Work

Unlike purely logical and inferential approaches to plan recognition, likelihood based Activity Recognition determines to what extent evidence supports competing plan hypotheses. Of likelihood based methods, we see those based on Hidden Markov Models, Bayesian techniques, and Support Vector Machines as best placed to provide robust and timely plan recognition.

The Hidden Markov Model (see Rabiner for a useful tutorial introduction) is a modeling technique which can be used to capture processes that cannot be observed directly. This feature, most famously leveraged for identifying underlying models in the case of noisy channels, corresponds well to the problem of activity recognition in real world environments. Specifically, in the case of activity recognition individual actions may be observable but it is not always possible to know the activity being pursued by the user – or even the procedural structure of a plan as understood by a user. For this reason a number of researchers have investigated the use of Hidden Markov Models and their variants in the plan recognition task. Bui for example has applied a variant on Hidden Markov Modelling to recognize behaviour in noisy domains and across multiple levels of abstraction (Bui, Venkatesh, and West 2002; Bui 2003).

Also in the activity recognition domain, in a range of papers Singla et al. investigated the use of both Hidden Markov Models and the more straightforward Markov Model for Activity Recognition. With respect to the use of Hidden Markov Models, Singla, Cook, and Schmitter-

Edgecombe(2009) analysed a dataset of users performing an extended number of Activities of Daily Living (ADL) where users were free to interleave activities. Whereas in previous work with Markov Models Singla et al. trained one model per ADL, in this work on HMMs, a single Hidden Markov Model was instead trained for all activities to be performed. The authors argument for this change was that this allows for the modelling of potential sequences of activities without worrying about the particular sequences of events which together compose individual activities. While the authors openly state that they have no interest in the sequential nature of the activities, we see this as a sacrifice of information that may not be necessary.

In addition to Hidden Markov Models, classifier based techniques may also be used for activity recognition. Of the classifier based techniques, those based on Bayes Theorem have arguably been the most extensively applied to activity recognition in the last ten years. While some early application of Bayes theory to activity and plan recognition looked at the use of Bayesian Networks (See e.g., Charniak and Goldman(1993) and Pynadath and Wellman(1995)), more recent work has focused on the use of the naive Bayes classifier which is a Bayesian inference model that has a strong independence assumption. In addition to applying Markov based methods to the CASAS smart home datasets, Singla, Cook, and Schmitter-Edgecombe (2009) have also for example made use of a naive bayes classifier to identify activities in the case of interleaved data. Singla's approach to using the Naive Based classifier centres on using each sensor event as a feature type, and learning the probability distributions over features for each event type. Using this simple approach, Singla reports relatively poor accuracy results of approximately 66% for the interleaved data set considered. Such a result is in contrast with the usually well accepted performance for Naive Bayes

An increasingly popular classification technique is the Support Vector Machine (Das and Cook. 2011). Essentially Support Vector Machines build on the observation that the essential job for any classifier is to create suitable decision boundaries between different classes based on observed instances. In a brief comparison of SVM methods to neural network and boosting methods, Chen, Das, and Cook(2010) applied Support Vector Machines to the human activity recognition problem. In this work a large number of features are used to identify the activity currently being performed. The primary focus of this work was on the benefits of boosting methods and on the the description of the features employed. As such limited detail was given on the relative performance of the methods considered, or of any indication of their relative performance with respect to Bayesian or Markov based techniques. Nevertheless this work and follow up work such as the more recent related study by Krishnan and Cook(2012) highlights the utility of applying machine learning methods other than bayesian techniques to the activity recognition problem,

It is clear from the literature that each of Bayesian, Markov based, and SVM based activity recognition methods have shown considerable promise for human activity recognition in the case of non-trivial data. Yet it could be argued that no single method has emerged as the definitive method of choice for activity recognition. Moreover while there are many studies which have investigated raw accuracy performance, we argue that there is a dearth of investigation of other key performance metrics such as the performance robustness in the case of missing data and the timeliness of prediction. We argue that performance metrics such as these are essential in the case of real time activity recognition in real environments.

Testbeds

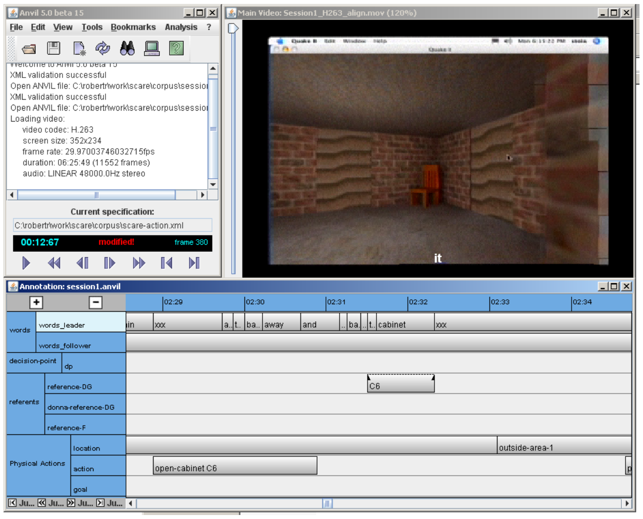
Given the relative importance of these methods, we investigated the relative performance of HMM based, Naive Bayes based, and SVM based methods in activity recognition. Specifically in the following we outline a study to evaluate the relative utility of these methods with respect to accuracy, robustness to missing data, and timeliness. For this work we made use of three human activity-centric testbeds. In the following we give background on these testbeds and describe the representations which we adopted for each of the Activity Recognition methods that were investigated.

The SCARE Testbed

The SCARE multimodal corpus of situated dialogues is a collection of annotated videos and audio recordings of participant pairs performing joint tasks in a simulated 3D environment (Stoia et al. 2006). In total 15 session recordings are included in the corpus. For each of these sessions, two participants were recorded while performing 5 distinct activities. Each activity in turn involved moving an item from one location to another in a virtual environment. One of the two session participants, the Instruction Giver was provided with a schematic map of the environment and was informed of the five activities that were to be performed. Specifically, the Instruction Giver was aware of: (a) which objects were to be moved; (b) where these objects were to be moved from; and (c) where these objects were to be moved to. The Instruction Follower meanwhile could maneuver around in the virtual environment and manipulate that environment by moving objects, opening and closing containers, as well as picking up and placing down items. In the real environment, the Instruction Giver and Instruction Follower were placed in separate rooms and verbally communicated via headsets to jointly complete all five tasks.

The SCARE corpus was selected as a testbed for a number of reasons. First, instruction givers and instruction followers were not constrained in terms of the sequence in which activities were to be performed, and whether or not these activities could be performed in parallel. Thus there was wide variation in terms of the interleaving of activity completion in this tesbed. Second, the activities seen in the data are also varied in length with both very long and short activities present. Third, the SCARE dataset came pre-annotated with respect to a number of features which will be beneficial to our long term studies, i.e.., the SCARE corpus is provided with time-aligned speech recognition transcriptions and reference information.

Although the SCARE corpus includes a number of useful



```
Welcome to Anvil 5.0 beta 15
XML validation successful
Open ANVIL file: C:\robertr\work\scare\corpus\sessio
XML validation successful
Open ANVIL file: C:\robertr\work\scare\corpus\sessio
Loading video:
    video codec: H.263
    screen size: 352x234
    frame rate: 29.97003746032715fps
    duration: 06:25:49 (11552 frames)
    audio: LINEAR 48000.0Hz stereo
```

Current specification: C:\robertr\work\scare\corpus\scare-action.xml

00:12:67 modified! frame 380

Figure 1: Annotation of the SCARE corpus with activity information in the ANVIL tool.

annotations such as time aligned speech transcriptions, Activity Recognition requires further annotation of the data. We therefore developed and applied an annotation scheme for activity recognition with the SCARE corpus. The annotation scheme included three distinct layers which labeled the data with respect to: (a) location of the instruction follower; (b) actions being performed by the instruction follower; and (c) activities or goals currently being pursued. Specifically, the *location* layer denoted which of 19 possible area of the environment that the Instruction Follower was currently located in. The *action* layer on the other hand denoted what physical manipulation actions – if any – were currently being performed by the Instruction Follower. In total there were six action types where these types could in turn be parameterised. Examples of action types include *Pickup(Silencer)* and *Open-Cabinet(C1)* where *Silencer* was an object in the environment and *C1* was a named container in that environment. The *activity* layer was a more course grained annotation layer that denoted what activities were currently being pursued by the participant pairing. In total 5 activities were pursued by each dyad, and these activities were frequently interleaved by participants. Thus the annotation scheme was defined to mark only the start and end of each activity; thus allowing for activities to be concurrently active.

The annotation scheme was applied to each of the 15 SCARE session recordings. For annotation we made use of the ANVIL multimodal annotation tool (Kipp 2001). For illustration, Figure 1 shows the in-world view as seen by the instruction follower and the instruction giver; and an excerpt of our extended activity-oriented SCARE annotation as conducted with the ANVIL tool.

The Washington State University Testbeds

The second and third testbeds that we consider have been sourced from the collection of daily living activity corpora that were collected and annotated by the CASAS Smart Home Project at Washington State University (Cook and Schmitter-Edgecombe 2009; Singla, Cook, and Schmitter-Edgecombe 2009). The corpora collection provides a wide range of data sets which record activities performed by participants in smart home and smart apartment settings. Over 20 individual corpora are provided, and these vary in terms of whether the data was recorded for single versus multiple participants; interleaved versus non-interleaved activities; and whether the data was recorded in an apartment, workplace or real home setting.

```
2008-02-27 12:52:44.712481 M17 ON
2008-02-27 12:52:46.943816 M17 OFF
2008-02-27 12:52:48.29525 M17 ON
2008-02-27 12:52:52.22381 M17 OFF
2008-02-27 12:53:03.940956 M17 ON
2008-02-27 12:53:04.74972 D01 OPEN
2008-02-27 12:53:07.523938 D01 CLOSE
2008-02-27 12:53:19.6455 AD1-A 3.28802
2008-02-27 12:53:26.812782 M17 OFF
2008-02-27 12:53:27.142702 M16 ON
2008-02-27 12:53:34.467035 M16 OFF
```

Figure 2: Sensor log extract from the Kyoto-ADL Data Set.

For our current studies we used the Kyoto ADL (*Activities of Daily Living*) and Kyoto Interleaved ADL data sets[1]. Kyoto-ADL is a labelled dataset which captures the performance of 5 distinct activities by 24 individuals. Each individual performed each of the 5 activities in sequence and without allowing the activities to overlap. For each activity performance, the dataset records a sequence of timestamped events. The timestamps for these events is the absolute date and time at which the event was recorded. The events themselves may be of two different types. Location events record the likely position of the participant as indicated by a set of motion sensors. Activation events on the other hand record when the participant has triggered one of a range of embedded environmental sensors that record the opening and closing of kitchen cabinet doors and so forth. To illustrate, Figure 2 provides a sensor log extract from the Kyoto ADL Data Set. Note that each sensor event is generally associated with a parameter. Motion detection sensor events (labelled M17, M16 in the excerpt) can be either ON or OFF. Other sensor events can have either categorial parameters, e.g., such as the sensor events associated with the Door D01, or can be real valued as is the case with sensor AD1-A.

The Kyoto-ADL corpus is useful as a baseline testbed, but is limited by a small number of performed activities and non-interleaving of activity execution. In light of this we also made use of the Kyoto-Interleaved dataset. As with Kyoto-ADL, the Kyoto-Interleaved dataset is a collection of recordings of individuals as they perform a range of activities of daily living in an apartment setting. However in this case the recorded data is more complex in that each participant performs 8 rather than 5 activities, and crucially, par-

[1] From this point on we will refer to these testbeds simply as *Kyoto-ADL* and *Kyoto-Interleaved* respectively.

ticipants first perform all activities sequentially, but then are asked to perform the activities again in any order or interleaved manner they see fit. Otherwise the datasets are otherwise similar to those recorded for Kyoto-ADL, with each activity recorded as a sequence of sensor events that are both time stamped and may be parametrized with categorial or real values. While data for 20 individuals are published for the Kyoto-Interleaved corpus, two of these are partial with recordings of certain activities absent. We therefore omitted the data for these two individuals from subsequent analysis.

Data Preparation

HMMs operate over a sequence of observations which in our case are low-level actions or events. For the Kyoto-ADL and Kyoto-Interleaved datasets this sequence of events was derived straightforwardly from the raw data sequences. Specifically each event corresponded to a sensor type and the parameter which was applied to it. Thus the sensor event for the first line in Figure 2 was simply M17_ON. In the case of real valued parameters we truncated parameter information to a single ACTIVE class. While we acknowledge that this leads to a loss of information, we believe that at least in the case of the datasets under consideration that this is an acceptable simplification. Thus, in the case of the eighth event in Figure 2, the sensor event was simply AD1-A_ACTIVE.

In the case of the SCARE dataset the construction of event sequences for HMMs was also straightforward. For each activity an event stream was algorithmically constructed from raw annotated data. The event stream consisted of either place events which were triggered by a new place annotation in the *Location* annotation layer, or action events which were triggered by annotations in the *Action* layer.

For classification with either SVM or Naive Bayes techniques we require a suitable set of features for analysis. We adopted a similar modelling approach for both the SCARE and Kyoto datasets where a baseline set of almost historyless features were augmented with a suitable representation of event history. Namely as the base set of features for the SCARE dataset the following six features were used:

- **Step** - The number of observations since the beginning of this task.
- **Time** - The amount of time that has passed since the beginning of this task.
- **Place** - The current location of the instruction follower in the virtual environment.
- **Activity** - The current activity being performed by the instruction follower. Possibly none or null if no action is being performed.
- **LastPlace** - The last location of the instruction follower.
- **LastActivity** - The last action performed by the instruction follower - possibly none if no action has yet been performed by the instruction follower for this activity.

For the Kyoto-ADL and Kyoto-Interleaved datasets a similar baseline set of features was used to characterize each recorded event. However in the case of the Kyoto datasets both place and activity were conflated into a single stream in the raw data, i.e., the four base features for the Kyoto datasets were:

- **Step** - The number of observations since the beginning of this task.
- **Time** - The amount of time that has passed since the beginning of this task.
- **Event** - The latest location or activity event that has been observed.
- **LastEvent** - The last location or activity event that has been observed.

Although the baseline features capture a minimal history by including the *LastEvent* or *LastPlace* and *LastActivity* features, a more complex model which includes histories for individual features is essential for modelling medium and long range dependencies between events. For such a model we model one feature per sensor and have that feature capture the activation history for that sensor. While this method and its motivation are straightforward, some variability is possible in terms of how we choose to capture sensor history in each feature. In this analysis we apply and test two different modelling approaches. For the first model, we adopted an approach where independent binary features were created for each possible level associated with the original location and action/event features. These binary features captured whether or not that particular event had been observed in the data stream during a rolling history of n observations. As outlined later in the results section, we tested for values of n ranging from 0 (where a feature was only active if the event type had occurred during the current observation) to values such as 150 (where a feature was active if the event type had been seen in any of the previous 150 observations).

For the second set of models, a similar rolling window method was adopted. However, in this case the binary features were replaced with simple counts of how many times the particular event type had been seen in the past n observations. In practice the former set of models (based on binary features) were derived from the latter set of models by replacing all values greater than 0 with TRUE, and replacing all 0 values with FALSE. Both binary feature based (Model-**B**) and count feature (Model-**C**) based models were developed for each of the Scare, Kyoto-ADL and Kyoto-Interleaved datasets.

Results

Following pre-processing of the data, we applied Support Vector Machine and Naive Bayes classifiers to both binary and count based models for the SCARE and Kyoto datasets. Similarly we applied Hidden Markov Model based techniques to the raw data for all three datasets. In the following we present the results of that analysis with respect to our key metrics of accuracy, timeliness, and robustness to missing data.

Accuracy

To perform activity recognition with Hidden Markov Models there is flexibility in terms of how we choose to model the domain. For the analysis presented here we adopted a

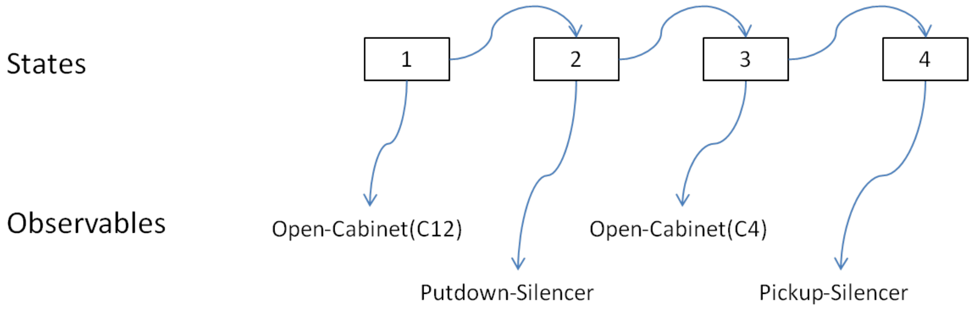


Figure 3: Illustration of the a 4-state HMM model for the SCARE activity Goal-Move-Silencer.

modelling approach where one HMM was trained per activity. Thus, events were observables in this modelling approach, and one latent variable was assumed per activity, where states of this variable corresponded to latent states of the given activity. This modelling approach is in contrast with the approach previously adopted by Singla, Cook, and Schmitter-Edgecombe(2009) where a single HMM was trained such that internal states of that single HMM modelled specific activities. To illustrate our modelling approach, Figure 3 presents an illustration of a 4-state HMM for the SCARE dataset activity *Goal-Move-Silencer*. While our modelling approach makes no claims about the specific internal states of a given activity HMM, we believe that our approach is a more natural approach to modelling activities with HMMs since we are taking advantage of the natural partial ordering of events in a given activity. Moreover, unlike in Singla's approach we need not assume any particular orderings between the pursuit of individual activities of daily living.

Since HMMs must be parameterized with a specific number of internal states, we trained HMMs with a variety of numbers of internal states for each of the three datasets. HMMs with 4 internal states were found to be optimal for our purposes, and we thus present performance measures with respect to these 4 state models. To avoid overfitting to the dataset we applied a k-fold training and testing strategy over each dataset. Namely, for each observation sequence to be tested with, a new HMM was built using a training data set which consisted of the complete dataset minus the target data sequence. Moreover since HMM instantiation is partially dependent on an initial probabilistic distribution, we also ran each epoch of testing 10 times, and took average accuracy values for each of the 10 iterations.

Given a set of HMMs trained for each activity in the dataset, we determine the most likely activity for a given event observation by applying the forward algorithm to each HMM and selecting the HMM with the highest likelihood as the prediction at that point in the event sequence. Thus for each observed event we have a specific activity prediction. Table 1 summarizes results for HMM based activity prediction for the Scare, Kyoto-ADL, and Kyoto-Interleaved datasets. Moreover, since the activity recognition problem is essentially a multi-class classification problem we report prediction accuracy both in terms of raw accuracy and also in terms of Cohen's Kappa Coefficient κ^2. While the HMM based detection performed relatively well on the non-interleaved Kyoto-ADL dataset, it can be seen that the method performed poorly on both of the interleaved activity datasets.

	Scare	Kyoto-ADL	Kyoto-Interleaved
accuracy	0.541	0.803	0.445
kappa statistic	0.401	0.742	0.346

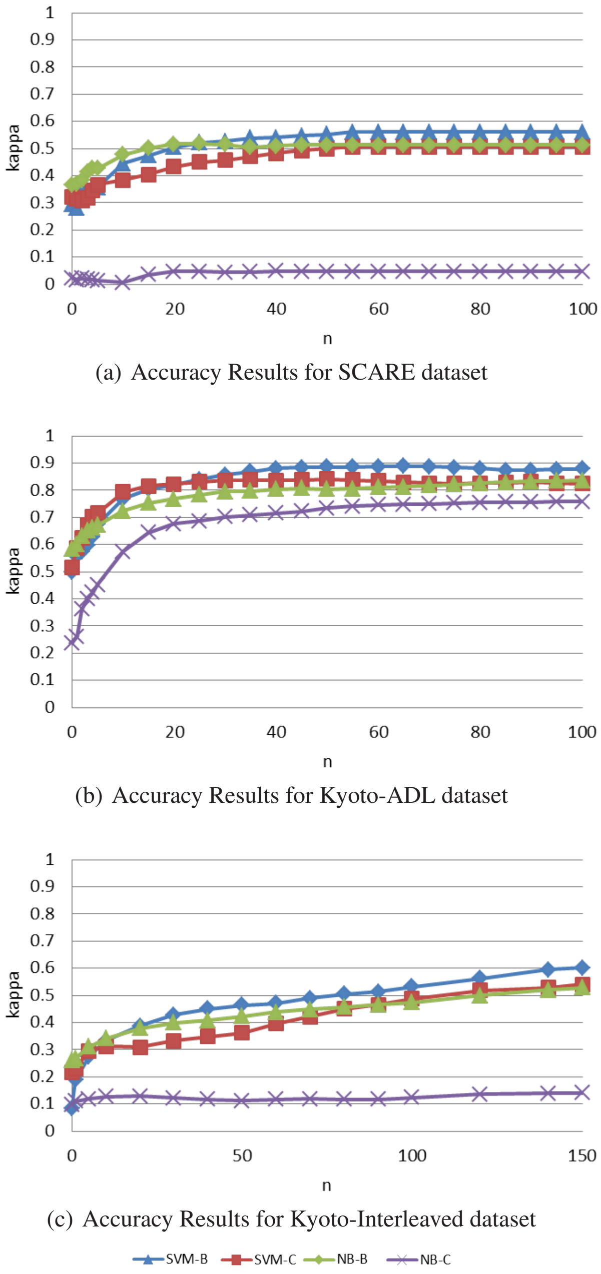
Table 1: Activity prediction accuracy and Cohen's Kappa statistic measures for HMM model with 4 states. Results presented for SCARE, Kyoto-ADL, and Kyoto-Interleaved datasets.

For SVM and Naive Bayes based prediction we developed classifiers for a range of history window lengths for each dataset. In addition, to avoid over-fitting we also performed k-fold classification within each dataset. Figure 4 summarizes the kappa coefficient scores for each of the four classifier based prediction models considered with respect ot the Scare, Kyoto-ADL, and Kyoto-Interleaved datasets. Unsurprisingly, accuracy – as measured in terms of Cohen's kappa coefficient here – rises as our models allow a greater window of history to be included in the defining features; here the SVMs prediction accuracy for Kyoto-ADL at n=100 is 87.48%. Also unsurprisingly, the we see that as with the case of HMMs we obtain significantly better activity prediction results for the non-interleaved Kyoto-ADL dataset than we do for the interleaved Scare and Kyoto-Interleaved datasets. Somewhat surprisingly however, it can be seen that the SVMs trained on binary features (SVM-B) perform consistently better than the SVM classifiers trained on count features, and better than both Naive Bayes based classification models. It is also significant to note that the Naive Bayes classifier performed exceptionally poorly on count based features in both interleaved datasets.

Comparing the classifier based model results to those for the HMM based prediction methods presented in Table 1, we see that the classifier based models generally outperformed

[2] Cohen's kappa is a conservative measure of accuracy which takes into account the possibility of class agreement happening by chance.

the HMM based models. The one exception to this was in the case of the interleaved Scare and Kyoto-Interleaved datasets where the count based Naive Bayes classifier performed considerably poorer than the HMM based method.



(a) Accuracy Results for SCARE dataset

(b) Accuracy Results for Kyoto-ADL dataset

(c) Accuracy Results for Kyoto-Interleaved dataset

Figure 4: Accuracy results for SCARE, Kyoto-ADL and Kyoto-Interleaved datasets. For each of the datasets, we plot kappa accuracy measure against n – the history window length used in building models B and C. Results are plotted for both SVM and Naive Bayes based classifiers on both the binary (B) and count (C) based models.

Timeliness

We also measured the relative *timeliness* of accurate prediction for both the classifier model variants and the HMM model. Here we define timeliness as the relative time at which the classifier obtains an accurate prediction of the activity which does not switch to an inaccurate prediction before the end of all events in that sequence. To illustrate with an example consider a sequence of 200 events for a given activity. If the correct activity was detected for the 40^{th} event and the class prediction then remains steady until the end of the event sequence, the timeliness measure for this event is 0.2. If instead the class prediction deviated from the correct activity after the 40^{th} event and only returned to the correct activity and become stable after 100 events, then our timeliness measure would be 0.5.

We calculated the timeliness of accurate class prediction for each target sequence in the case of both HMM and classifier based activity prediction. An overall timeliness value was then calculated for each prediction technique by taking the mean over all test sequences. In the case of prediction failure, i.e., where the HMM method or classifiers failed to successfully predict the correct target activity type by the end of a given test event sequence, we assumed an NA value for the timeliness measure for this activity instance with this classifier.

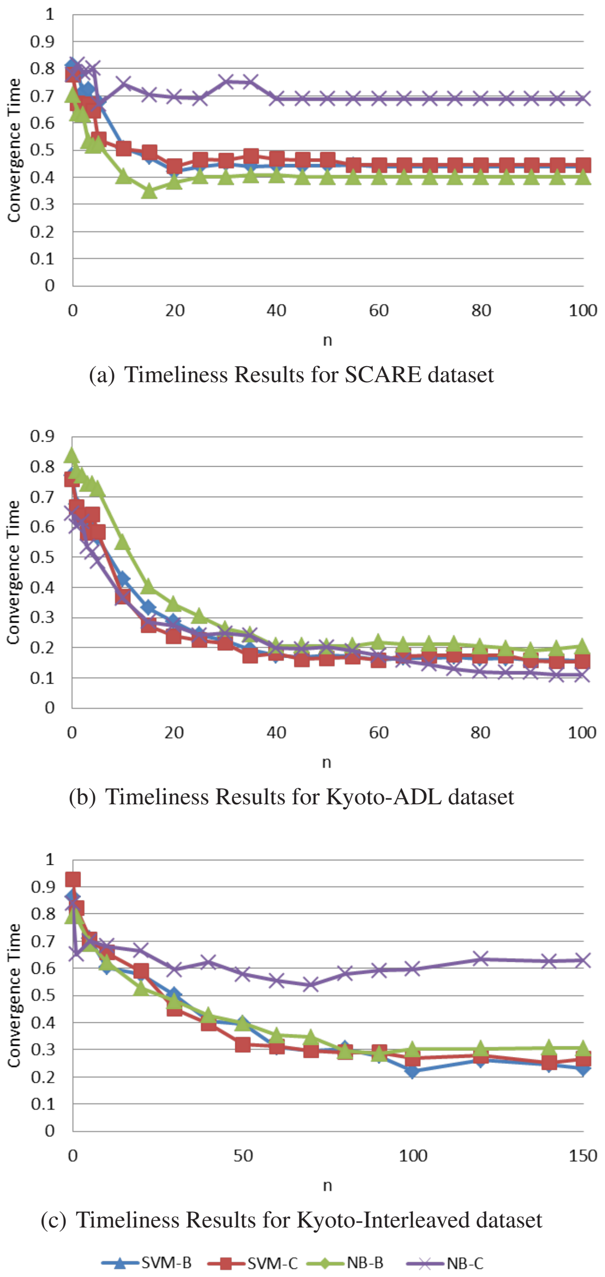
	Scare	Kyoto-ADL	Kyoto-Interleaved
timeliness	0.343	0.204	0.412

Table 2: Timeliness measures for HMM model with 4 states. Results presented for SCARE, Kyoto-ADL, and Kyoto-Interleaved datasets.

Table 2 presents timeliness measure for HMM based prediction while Figure 5 presents timeliness measures for each of the 4 classifier based models. From the results we can see that there is a strong correlation between accuracy results and timeliness. This can be understood to be due to the fact that since if a classifier is performing poorly it is not likely that the correct answer is converged on until a reasonable amount of data has been processed. Despite this, the relationship between classification accuracy and timeliness is not direct. For example, the Naive Bayes classifier when applied to count based data (NB-C) performs best on the Kyoto-ADL data in terms of timeliness, but is not the best performer for cases of measurement accuracy.

Robustness

For sensor based event and activity detection, there is a very real reality of incomplete and noisy data. Even if algorithms perform well for ideal data, there is no guarantee that they will perform well for noisy data or missing data. In order to contrast the relative robustness of SVM, Naive Bayes, and HMM based activity detection we performed one final test in which each activity detection algorithm was evaluated against randomly selected portions of the recorded data. Specifically for each of the datasets we removed between 5% and 45% of the original data during n-fold testing. The



(a) Timeliness Results for SCARE dataset

(b) Timeliness Results for Kyoto-ADL dataset

(c) Timeliness Results for Kyoto-Interleaved dataset

Figure 5: Timeliness results for SCARE, Kyoto-ADL and Kyoto-Interleaved datasets. For each of the datasets, we plot timelinesss measure τ against n – the history window length used in building models B and C. Results are plotted for both SVM and Naive Bayes based classifiers on both the binary (B) and count (C) based models.

full data set was however used for training each of the activity recognition models. It should be noted that in the case of the classifier based techniques, it was necessary to regenerate the feature sets associated with each remaining event once a proportion of the data had been eliminated from the dataset.

Figure 6 presents the results of the robustness tests for each of the 5 activity recognition algorithms considered against each of the three datasets. In general we can see that each of the algorithms performed well in all cases. With respect to SCARE and Kyoto-ADL we see that the Naive Bayes classifier operating on the binary data held up particularly well as the percentage of data presented was reduced towards the 55% mark. In the case of the Kyoto-Interleaved dataset meanwhile we see that very little degradation in performance is observed even after almost 50% of the test data in held back during evaluation. Particularly in the case of Kyoto-ADL we see this as being due simply the the high volume of initial data, i.e., even after 45% of data had been discarded from a test set, there remained a very large quantity of data for each activity, thus meaning there was still a high probability of characteristic features occurring in the test data.

Conclusions

The studies presented in this workshop paper aim to provide a modest review of the relative merits of three prominent and competing Machine Learning approaches to the activity recognition problem. We believe our findings confirm the increasingly ubiquitous dominance of kernel methods such as Support Vector Machines in general and specifically confirm their use in the field of activity recognition. While this is true, the results also point out that considerable more care and investigation will be required to determine just what features are most appropriate for capturing the event history such to optimize classifier performance. We saw from our results that a simple set of binary features outperforms a more information rich count based approach.

As indicated, we present the current study as an initial study into the relative merits of likelihood based activity recognition techniques. In future work we aim to build upon the work here by investigating representation models for sensor histories.

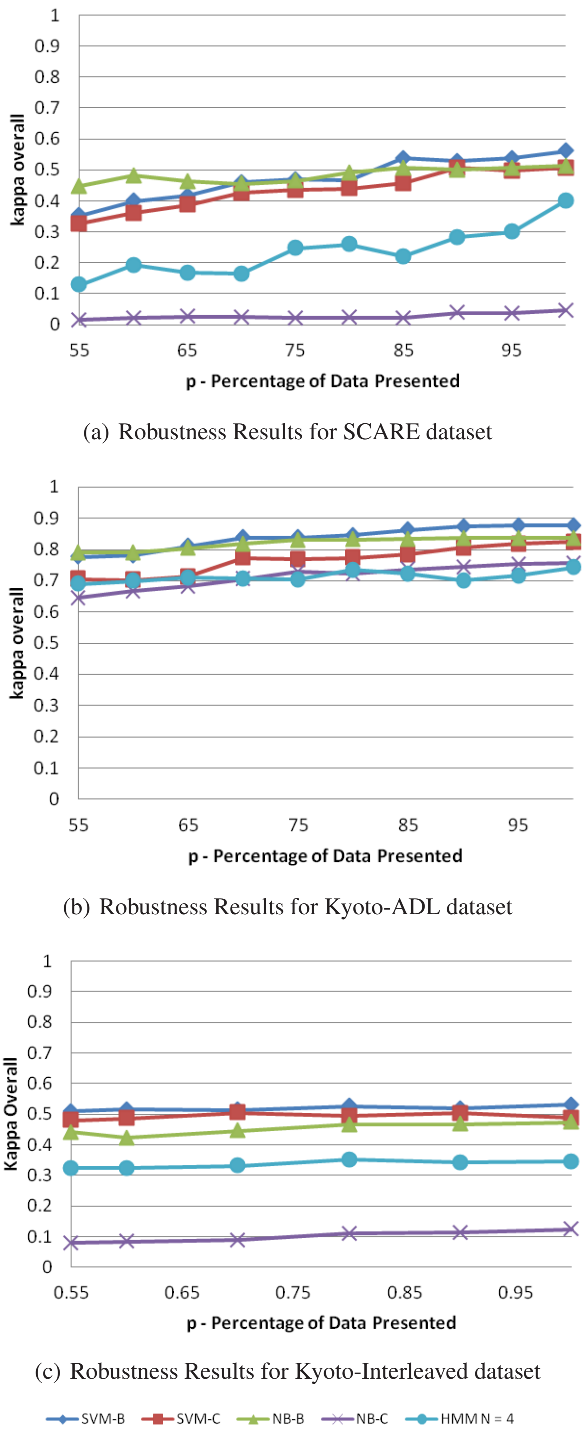
Acknowledgements

The authors wish to thank the Irish Research Council for Science Engineering and Technology (IRCSET) for their support through the IRCSET EMPOWER funding programme.

References

Bui, H.; Venkatesh, S.; and West, G. A. W. 2002. Policy recognition in the abstract hidden markov model. *Journal of Artificial Intelligence Research* 17:451–499.

Bui, H. 2003. A general model for online probabilistic plan recognition. In *In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI*, 1309–1315.



(a) Robustness Results for SCARE dataset

(b) Robustness Results for Kyoto-ADL dataset

(c) Robustness Results for Kyoto-Interleaved dataset

Figure 6: Robustness results for SCARE, Kyoto-ADL and Kyoto-Interleaved datasets. For each of the datasets, we plot the overall kappa statistic score for accuracy against p – the percentage of original data that was presented for testing.

Burges, C. J. C. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2:121–167.

Charniak, E., and Goldman, R. P. 1993. A bayesian model of plan recognition. *Artificial Intelligence* 64:53–79.

Chen, C.; Das, B.; and Cook, D. J. 2010. A data mining framework for activity recognition in smart environments. In *Proceedings of the 2010 Sixth International Conference on Intelligent Environments*, IE '10, 80–83. Washington, DC, USA: IEEE Computer Society.

Cook, D., and Schmitter-Edgecombe, M. 2009. Assessing the quality of activities in a smart environment. *Methods of Information in Medicine.* 48(5):480–485.

Das, B., and Cook., D. 2011. Data mining challenges in automated prompting systems. In *Workshop on Interacting with Smart Objects.*

Kipp, M. 2001. ANVIL - a generic annotation tool for multimodal dialogue. In Dalsgaard, P.; Lindberg, B.; Benner, H.; and Tan, Z.-H., eds., *INTERSPEECH*, 1367–1370. ISCA.

Krishnan, N. C., and Cook, D. J. 2012. Activity recognition on streaming sensor data. *Pervasive and Mobile Computing*.

Pynadath, D. V., and Wellman, M. P. 1995. Accounting for context in plan recognition, with application to traffic monitoring. In Besnard, Philippe, and Hanks, S., eds., *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI'95)*, 472–481. San Francisco, CA, USA: Morgan Kaufmann Publishers.

Rabiner, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2):257–285.

Rish, I. 2001. An empirical study of the naive Bayes classifier. In *IJCAI-01 workshop on "Empirical Methods in AI"*.

Singla, G.; Cook, D.; and Schmitter-Edgecombe, M. 2009. Tracking activities in complex settings using smart environment technologies. *International Journal of BioSciences, Psychiatry and Technology* 1(1):25–35.

Stoia, L.; Shockley, D. M.; Byron, D. K.; and Fosler-Lussier, E. 2006. Noun phrase generation for situated dialogs. In *Proceedings of the Fourth International Natural Language Generation Conference*, 81–88. Sydney, Australia: Association for Computational Linguistics.