A Hybrid Discriminative/Generative Approach for Modeling Human Activities

Jonathan Lester[1], Tanzeem Choudhury[2], Nicky Kern[3], Gaetano Borriello[2,4] and Blake Hannaford[1]
[1]Department of Electrical Engineering, University of Washington, Seattle WA 98195, USA
[2]Intel Research Seattle, Seattle, WA 98105, USA
[3]Department of Computer Science, Darmstadt University of Technology, Darmstadt, Germany
[4]Department of Computer Science, University of Washington, Seattle WA 98195, USA

Abstract

Accurate recognition and tracking of human activities is an important goal of ubiquitous computing. Recent advances in the development of multi-modal wearable sensors enable us to gather rich datasets of human activities. However, the problem of automatically identifying the most useful features for modeling such activities remains largely unsolved. In this paper we present a hybrid approach to recognizing activities, which combines boosting to discriminatively select useful features and learn an ensemble of static classifiers to recognize different activities, with hidden Markov models (HMMs) to capture the temporal regularities and smoothness of activities. We tested the activity recognition system using over 12 hours of wearable-sensor data collected by volunteers in natural unconstrained environments. The models succeeded in identifying a small set of maximally informative features, and were able identify ten different human activities with an accuracy of 95%.

1 Introduction

The task of modeling human activities from body-worn sensors has received increasing attention in recent years, especially in the ubiquitous computing (Ubicomp) field [Bao and Intille, 2004; Lukowicz *et al.*, 2004; Patterson *et al.*, 2003]. Although originally most of the research in activity recognition was done using vision sensors [Gavrila, 1999; Pentland, 1996], it has increasingly become dominated by various types of wearable sensors, like accelerometers and audio. A fertile application domain for activity recognition is in the health care arena, especially in elder care support, long-term health-monitoring, and assisting those with cognitive disorders. In addition, activity recognition is an important component for modeling higher level human behavior, tracking routines, rituals, and social interactions.

The majority of research using wearable devices, has concentrated on using multiple sensors of a single modality, typically accelerometers on several locations on the body [Bao and Intille, 2004; Kern *et al.*, 2003]. The placement of sensors in multiple pre-defined locations can be quite obtrusive and is one of the limitations of such an approach. While the ultimate goal is to embed these devices into clothing, this technology is far from being commercially available and widely accepted. As a result, a single sensing device that can be integrated into existing mobile platforms, such as a cell phone, will be more appealing to users and is likely to garner greater user acceptance. Work by [Bao and Intille, 2004] has shown that an appropriate sensor subset (two locations), does not effect the recognition scores significantly (by less than 5%) compared to a system with five sensors; whereas the use of a single sensor reduced the average accuracy by 35%. Our hypothesis is that incorporating multiple sensor modalities will offset the information lost by using a single sensing device. Furthermore, multiple modalities will be better suited to record the rich perceptual cues that are present in the environment, cues that a single modality often fails to capture. Multiple modalities have already shown promise in earlier activity recognition experiments [Lukowicz *et al.*, 2002].

To capture the diverse cues from movement, sound, light, etc., about ongoing activities, we have built a very small sensing unit (2.53 sq. in.) that includes eight different sensors: accelerometer, audio, IR/visible light, high-frequency light, barometric pressure, humidity, temperature, and compass. Using these sensors, we have collected a large annotated dataset of various human activities from two volunteers over a period of six weeks. We compute over six hundred different features from these eight sensor modalities, which attempt to capture various attributes of the raw signal.

Often in activity recognition, the choice of sensors and the features derived from them are driven by human intuition and by what is easily available, rather than by performance or practicality. Using the right features is crucial for recognition. We are working towards developing a framework that allows us to systematically identify modalities and features that are most useful for machine recognition and discrimination of natural human activities.

In the end, we want models that accurately recognize and track a variety of activities and a system that is lightweight enough to run on devices like cell phones, which many people already carry. Thus, minimizing the computation cost of our recognition system is also an important goal.

The two main approaches that are used for classification in machine learning are: (i) generative techniques that model the underlying distributions of the classes and (ii) discriminative techniques that only focus on learning the class boundaries [Rubinstein and Hastie, 1997]. Both of these approaches have been used extensively in the vision and the wearable sensing communities for recognizing various human behavior and activities. The work presented in this paper is a hybrid approach that combines the two techniques. First, a modified version of AdaBoost proposed by [Viola and Jones, 2001], is used to automatically select the best features and to learn an ensemble of discriminative static classifiers for the activities we wish to recognize. Second, the classification margins from the static classifiers are used to compute the posterior probabilities, which are then used as inputs into HMM models. The discriminative classifiers are tuned to make different activities more distinguishable from each other, while the HMM layer on top of the static classification stage ensures temporal smoothness and allows us to continuously track the activities.

The rest of the paper is organized as follows: Section 2 provides an overview of the activity recognition system. Section 3 presents the feature selection and discriminative classifier training method. Section 4 describes how the results from the classifiers are combined with HMMs. Section 5 describes our experimental results and the performance of the system, and Section 6 discusses our conclusions and possible future directions

2 Activity Recognition System Overview

The first problem we address is the systematic identification of modalities and features that are well suited for accurate recognition of natural human activities. The second problem we tackle is how these features can be effectively used to develop models that accurately recognize and track various activities. Below we give a brief overview of the different components in our activity recognition system.

Sensing and Feature Extraction

Using a shoulder mounted multi-sensor board (Figure 1(A)), we collect approximately 18,000 samples of data per second. To reduce the dimensionality and to bring out details in the data we compute a total of 651 features; which include linear and mel-scale FFT frequency coefficients, cepstral coefficients, spectral entropy, band pass filter coefficients, integrals, mean and variances. We combine the features from various sensors to produce a 651 dimensional feature vector at 4Hz. However, since we have sensors with different sampling rates, there can be multiple instances of a feature within the 0.25 second window; that operate on different portions of the data. Furthermore, when calculating some features (e.g. the integral features) we incorporate a longer time window that varies from several seconds to as long as a minute. We restrict the time windows to only use data from the past, so that our system functions without any latency.

Feature Selection and Discriminative Activity Models

Earlier work has shown that discriminative methods often outperform generative models in classification tasks [Ng and Jordan, 2002]. Additionally, techniques such as bagging and boosting that combine a set of weak classifiers can further improve accuracy, without over-fitting to the training data [Schapire, 1999]. [Viola and Jones, 2001] have shown that boosting can be used not only as a method for combining classifiers but also as a method for selecting discriminative features. We use their proposed approach to select only a fraction of the total features, and to train very simple ensemble classifiers to recognize a broad set of activities.

Capturing Temporal Regularities

The activities people perform have certain natural regularities and temporal smoothness, e.g. people do not abruptly switch back and forth between walking and driving a car; thus, the recent history can help in predicting the present. Using a sequence of posterior probabilities computed from the instantaneous classifiers, we train Hidden Markov Models (HMMs) that significantly improve the performance and smoothness of our recognition system. By incorporating the static classification results we overcome the weakness of HMMs as effective classifiers [Jaakkola and Haussler, 1999].

3 Selecting the Right Features

Given a rich set of sensor data and features, our classifiers will work best if we select the right features that enable the classifiers to discriminate well between classes, and if we remove features that are not useful or which might even confuse the classifiers. Although it might be possible to hand pick the optimal features for certain activities, this is not a viable solution when the number of activities become large or when the sensor signals are not intuitive. In such scenarios, automatic techniques for finding the right set of features become increasingly important. A practical activity-recognition system will use a minimal number of features and the simplest possible models needed for high accuracy.

3.1 Feature Selection and Activity Classification using Boosted Decision Stumps

In this paper, we assume that people engage in N different type of activities. Given the set of activities $\mathbf{A} = \{A^1, ..., A^N\}$, we also assume that we have a set of training data for each of those activities. Each sample in the training set consists of a feature vector $\mathbf{f} = \{f_1, ..., f_K\}$ extracted from the sensors. For each activity A^i we are

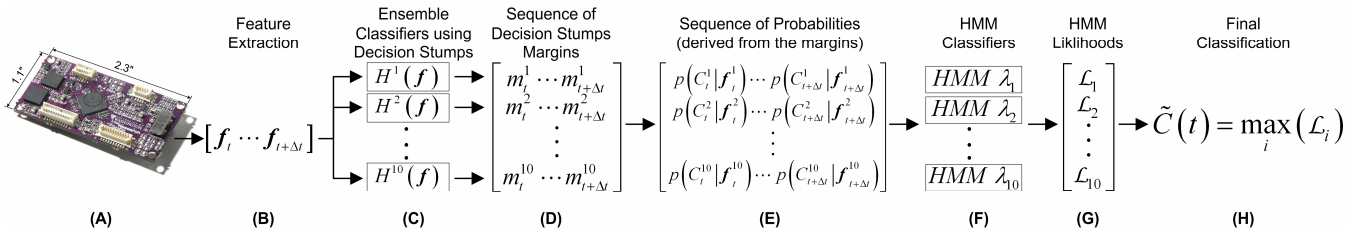

Figure 1: Flow diagram describing the classification system presented in this paper. **(A)** Our sensor board records a sequence of raw sensor recordings **(B)** from which we compute our feature vector. We pick the top fifty features per class, from our feature vector, and **(C)** supply them as inputs to our ensemble of decision stumps classifier. **(D)** Each decision stumps classifier outputs a margin at time t. **(E)** This sequence of margins can then be converted to probabilities by fitting them to a sigmoid. **(F)** The sequence of probabilities is then supplied to ten HMM classifiers **(G)** each of which outputs a likelihood. **(H)** The class with the highest likelihood is the classified class.

interested in finding a ranking of the feature set $\mathbf{R}^i = \{r_1^i, ..., r_K^i\}$ based on their usefulness in recognizing activity A^i. Moreover, we want to find a cut-off point τ^i for the ranked feature set such that adding features beyond τ_i does not significantly improve the accuracy of the classifier C^i, i.e. $\Delta(\text{error}(C^i(f_{r_1}^i, ..., f_{r_N}^i))), \quad \text{error}(C^i(f_{r_1}^i, ..., f_{r_\tau}^i))) \le \varepsilon$. The reason behind estimating τ^i is that, if $\tau^i \ll N$ then we can reduce the computational complexity of our classifiers by not extracting the less useful features. Since our final goal is to have the classifiers run on devices that users carry or wear, the computational costs of the classifiers are critical.

For each activity A^i, we iteratively train an ensemble of weak binary classifiers $H^i = \{h_1^i, ..., h_N^i\}$ (Figure 1(C)) and obtain a ranking $\mathbf{R}^i = \{r_1^i, ..., r_K^i\}$ for the features using the variation of the AdaBoost algorithm proposed by [Viola and Jones, 2001]. The weak classifiers are constrained to use only one feature, and at each iteration *m* of boosting we select the feature and the associated weak learner $h_m^i(f_m)$ that minimizes the training error $\varepsilon_m^i(f_m)$ on the weighted data. The error $\varepsilon_m^i(f_m)$ is used to re-weight the data for the next iteration and to compute the weight α_m^i for $h_m^i(f_m)$. At the end of this process, we have a ranking for the features based on how useful each feature is in discriminating A^i from the other activities A^j $(j \ne i)$, and we also have a set of weak classifiers $h_m^i(f_m)$ and weights for those classifiers α_m^i. The final output is a weighted combination of the weak classifiers, and by estimating the error of C^i as a function of the number of features used, we can also find τ^i for C^i. So, for a given data point, the prediction of C^i is

$$H^i(\mathbf{f}) = \text{sign}\left(\sum_{m=1}^{\tau^i} \alpha_m^i h_m^i(f_m)\right)$$

Each classifier C^i uses the top τ^i features, which is a fraction of the total number of features available, i.e. $\mathbf{f^i} = (f_1^i, ..., f_{\tau^i}^i)$.

We tried two different weak classifiers in our system: (i) a discriminative decision-stump and (ii) a generative naïve Bayes model (conditional probability distributions are modeled using histograms that have 100 bins/dimension) as our weak classifier. In our experiments, the decision stump consistently outperformed the naïve Bayes classifier, so we only use results from the decision-stump based classifier in the later sections. The decision stump finds the optimal threshold θ_m^i for each feature f_m that minimizes the weighted error such that $h_m^i(f_m) = 1$ if $f_m > \theta_m^i$ and $h_m^i(f_m) = -1$ otherwise.

For the boosted static classifiers, the classification margin for a data point can reflect the confidence in that prediction [Schapire *et al.*, 1997]. The margin of an example is the weighted fraction of the weak classifiers votes assigned to the correct class.

$$m^i(\mathbf{f^i}) = \frac{\sum_{m=1}^{\tau^i} \alpha_m^i h_m^i(f_m^i)}{\sum_{t=1}^{\tau^i} \alpha_m^i}$$

$$H^i(\mathbf{f^i}) = \text{sign}(m^i(\mathbf{f^i}))$$

However, constructing classifiers to output a posterior probability is very useful, especially if we want to combine the results of the multiple classifiers later on. One method of computing posterior probability directly is to fit a sigmoid function to the output of the ensemble classifier [Platt, 1999] (Figure 1(E)). In our case, the posterior probabilities are derived as follows -

$$p(C^i \mid \mathbf{f^i}) = \frac{e^{\varphi m^i(\mathbf{f^i})}}{e^{\varphi m^i(\mathbf{f^i})} + 1}, \text{ where } \varphi \text{ is a constant}$$

A static classifier predicts the label for each data point independently. Most of the time this independence assumption is clearly invalid and the prediction of previous data points can help with the current classification. A temporal model that uses the confidence of the predictions of the classifiers C^is instead of the raw features $\mathbf{f^i}$ is likely to have a greater impact on the performance. The ability to recognize activities in continuous time chunks would also allow us to learn how people transition between activities and thereby allow us to learn more about people's behavior and activity patterns. In the next section, we describe how we combine the confidence values of the static classifiers to build time-series models of the activities.

4 Incorporating Prediction History using Hidden Markov Models

HMMs have been successfully used in modeling different types of time-series data, e.g. in speech recognition, gesture

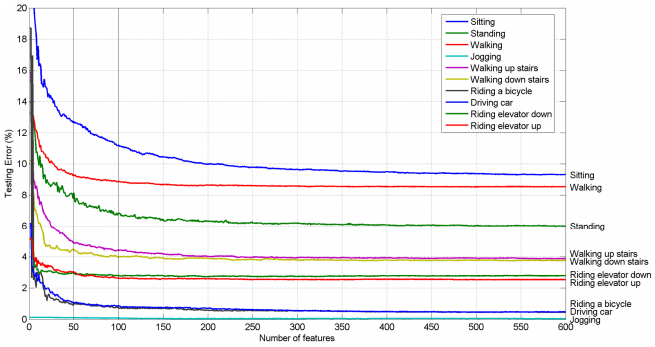

Figure 2: Testing error rates per class as a function of the number of features selected. After 50 features are selected most of the testing errors for the classes have leveled off. The data graphed here is averaged from several smaller feature selection runs.

tracking etc. We use HMMs to capture the temporal dynamics; but instead of directly using the raw features selected in the previous section, we trained our HMMs using the posterior probabilities of the static classifiers. The advantage of using the posterior probabilities is that we can take advantage of the results from the discriminatively trained classifier, as well as reduce the complexity of the HMMs. The earlier work of [Jaakkola and Haussler, 1999] has also shown the benefits of combining generative models into discriminative classifiers by deriving kernel functions from probability models. [Clarkson and Pentland, 1999; Oliver *et al.*, 2002] have used the output of an HMM model as input to another HMM. [Yin *et al.*, 2004] have used the output of static classifiers directly into HMMs for speech reading application; however they do not compute the margin or class posterior probability of these classifiers, which can be more effective than the raw outputs [Platt, 1999].

For each activity, a new HMM model λ_i (Figure 1(F)) is learned using a sequence of examples rather than individual instances. We construct a new input feature space based on the posterior class probabilities, which is

$$\tilde{\mathbf{f}} = \begin{bmatrix} p(C^1 \mid \mathbf{f}^1) \\ \vdots \\ p(C^N \mid \mathbf{f}^N) \end{bmatrix}$$

Given a set of observations $\{\tilde{\mathbf{f}}_1, \tilde{\mathbf{f}}_2, ..., \tilde{\mathbf{f}}_T\}$ for each activity, we learn the parameters of the HMM λ_i using the standard expectation-maximization (EM) method. During testing, we have a continuous sequence *S*, which we use to compute the likelihood value $L_i(t)$ for λ_i at time *t* using a sliding window of duration Δt (Figure 1(G)) –

$$L_i(t) = P(\tilde{\mathbf{f}}_t, \tilde{\mathbf{f}}_{t+1}, ..., \tilde{\mathbf{f}}_{t+\Delta t} \mid \lambda_i, S)$$

The final segmentation and classification is based on the HMM that has the highest likelihood value, i.e. $\tilde{C}_i(t) = \max_i L_i(t)$ (Figure 1(H)).

Accelerometer	37.7%
Ambient Light (IR-Vis)	2.5%
Audio	23.9%
Barometric Pressure	12.9%
Digital Compass	2.1%
Hi-Freq Vis Light	3.3%
IR Light	3.6%
Relative Humidity	4.1%
Temp. from Barometer	3.2%
Temp. from Relative Humidity	3.2%
Visible Light	3.3%

Table 2: The percentage of features from the top 50 that originated from the different sensors, averaged across all activity classes.

Alternatively, we could have trained the various states of a single HMM to recognize the different activity classes and to learn the transition characteristic between activities. We choose not to do that here as our activities are primitive and a single transition statistic is not very meaningful. However, we believe that the output of these HMMs could be used to train a single dynamic model of more complex behavior where the transition statistics would also be more informative.

5 Experiments

To validate our approach, we recorded 12 hours of data consisting of a large number of activities (such as sitting, walking, jogging, riding a bike, driving a car, etc.) using the wearable multi-sensor board. The dataset was collected across multiple days, by two volunteers (who are not the researchers) in various indoor and outdoor locations. The recordings were done in long stretches (one hour on average), where the duration of the activities themselves ranged from seconds (e.g. *entering a building*), to hours (e.g. *driving a car*). The volunteers were asked to go about performing a series of activities naturally without any specific constraints on the order, for example "go to building X and walk around inside". To capture day-to-day variations in the activities we collected multiple instances of the various activities over the course of six weeks. On average we have about an hour of data per activity and around 100 instances per activity.

Feature Selection

For the feature selection stage, we selected 80% of the total data available for each class for training. Based on the training examples we derived a ranking R^i for the features for each activity individually using the boosted decision stump procedure described in section 3.1. Figure 2 shows the testing error as a function of the number of features used for classification. From the results we see that classification error tapers off at around $\tau^i = 50$ features for most classes. If we were to pick more features beyond the top 50 our performance only improves slightly with the testing error only improving <1% for 600 features. The practical advantage of features selection is that we can significantly reduce the computational burden on our resource constrained devices, without drastically affecting the

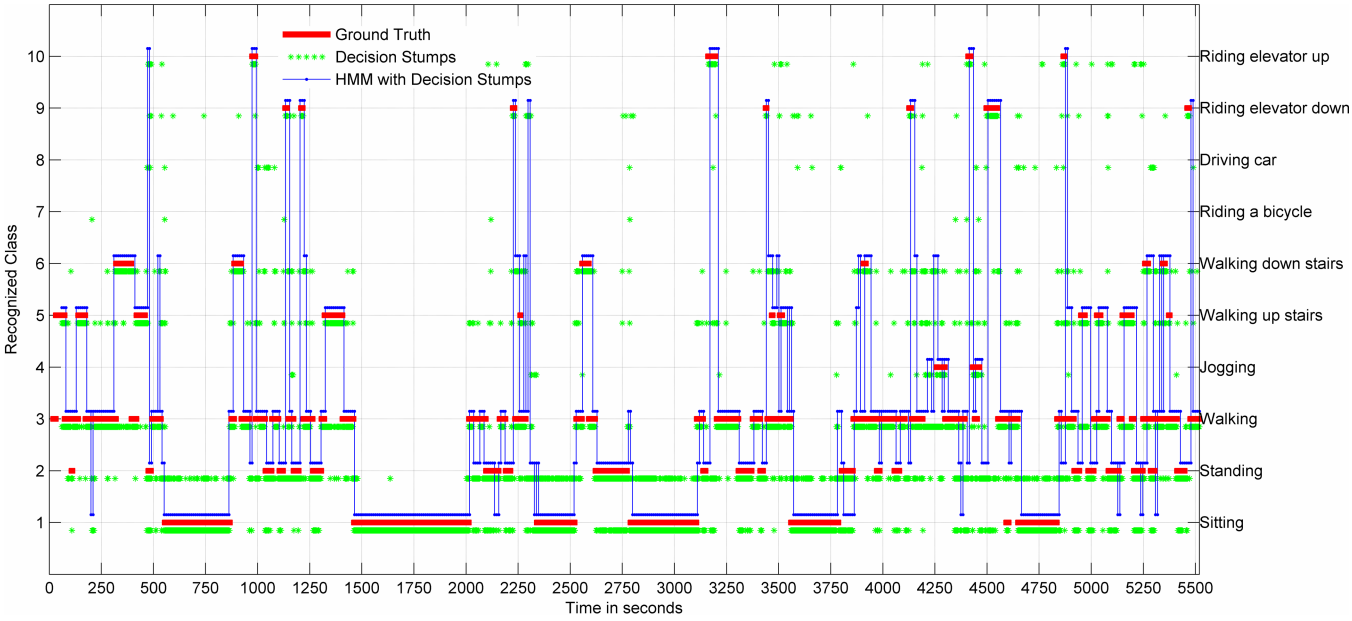

Figure 3: Output of the static decision stumps classifiers (at 4Hz), and the HMM classifiers (trained with output probabilities of the static classifiers) for a continuous 90 minute segment of the data. The results are overlaid on top of the ground truth which was obtained by annotating video recorded from a webcam worn by our volunteers. The video was only used for determining ground truth and not as an additional sensor input.

performance. Moreover, by performing boosting (re-weighting the data and selecting discriminatory features successively based on the error), we perform much better than taking a non-boosted (no re-weighting) approach to selecting the best 50 features. The accuracy ((true positive + true negative) / (total # of examples)) for the boosted features selection was on average ~11% higher than the non-boosted method. Table 2 lists the contribution of the different sensors to the final classifier. The majority of the top 50 features came from the accelerometer, audio and barometric pressure sensors. Barometric pressure data was useful in distinguishing activities that involved floor transitions (e.g. walking up/down stairs, elevator up/down); the sensor is sensitive enough to pick up pressure differences between a single floor.

Static Classification Results

Using the top 50 features we tested the performance of the ensemble classifier for two different weak classifiers – (i) decision stump (discriminative) and (ii) naïve Bayes (generative). The total duration of our test dataset was five and a half hours. The decision stumps outperformed the naïve Bayes classifiers by a large percentage. Table 3 shows the precision (true positive/(true positive + false positive)) and recall (true positive/(true positive + false negative)) numbers for the 10 activities in the dataset using the ensemble of decision stumps. Table 5 lists the average precision and recall numbers for the naïve Bayes as well as decision stump classifiers.

Continuous classification results

Although the decision stumps results from Table 3 are quite good on their own, Figure 3 illustrates the classification errors encountered for a continuous trace. The majority of the trace tends to be correctly classified by the decision stumps; but, with some scattered misclassifications. The addition of the HMM layer on top of the static classifier helps to smooth these classification errors as shown by the line in Figure 3.

The parameters of the HMMs were trained using 20 example scenes (on average 30 minutes of scenes) for each class. Each HMM had two hidden states and Gaussian observation probabilities. Classification was performed using a 15 second sliding window with 5 second overlap. Table 4 shows the sliding window performance results for the HMM, using the posterior probabilities as inputs, tested with concatenated test scenes. The overall accuracy in this case was 95%. It is interesting to note that the points in Figure 3 where the HMM and ground truth differ appear to be somewhat natural and realistic; for example, classifying a region without any ground truth between walking and sitting as standing. In fact, the HMM output reveals some deficiencies in the ground truth. For example, some segments whose ground truth was marked as walking are in fact standing (as determined by post-analysis of the video by the experimenters), and are correctly recognized as standing by the HMM.

To compare the performance to a more standard HMM approach, we trained a new set of HMMs that used the top 50 raw features as inputs rather than the output of the static classifiers. The performance of these HMMs was

Precision

Labeled Activities \ Classified Activity (by Decision Stumps)	Sitting	Standing	Walking	Jogging	Walking up stairs	Walking down stairs	Riding a bicycle	Driving car	Riding elevator down	Riding elevator up
Sitting	90.9%	43.3%	1.1%	0.3%	2.6%	2.7%	7.2%	10.2%	9.0%	5.6%
Standing	7.1%	44.9%	0.3%		0.9%	0.3%	1.8%	0.7%	1.5%	1.5%
Walking	1.2%	8.7%	95.1%	1.3%	21.1%	12.9%	5.4%	1.3%	0.8%	0.7%
Jogging	0.0%		0.1%	98.3%		0.1%				
Walking up stairs	0.0%	0.1%	1.9%		73.6%	0.7%	0.1%	0.0%		0.2%
Walking down stairs		0.0%	1.4%	0.1%	1.0%	83.0%	0.1%		0.5%	
Riding a bicycle	0.1%	0.1%	0.2%				85.3%			
Driving car	0.5%	0.0%	0.0%		0.2%		0.1%	87.7%	0.1%	0.2%
Riding elevator down	0.1%	1.7%			0.1%	0.1%		0.0%	87.5%	0.4%
Riding elevator up	0.1%	1.2%	0.0%		0.5%			0.1%	0.5%	91.4%

Recall

Labeled Activities \ Classified Activity (by Decision Stumps)	Sitting	Standing	Walking	Jogging	Walking up stairs	Walking down stairs	Riding a bicycle	Driving car	Riding elevator down	Riding elevator up
Sitting	86.6%	10.0%	0.8%	0.0%	0.1%	0.1%	0.8%	1.3%	0.2%	0.1%
Standing	38.2%	58.2%	1.3%		0.3%	0.1%	1.0%	0.5%	0.2%	0.2%
Walking	1.6%	2.7%	92.6%	0.0%	1.5%	0.7%	0.7%	0.2%	0.0%	0.0%
Jogging	0.1%		1.8%	97.7%		0.3%				
Walking up stairs	0.1%	0.2%	26.1%		72.8%	0.6%	0.1%	0.1%		0.1%
Walking down stairs		0.1%	22.7%	0.1%	1.2%	75.4%	0.3%		0.3%	
Riding a bicycle	0.8%	0.3%	1.3%				97.6%			
Driving car	4.1%	0.0%	0.1%		0.1%		0.1%	95.6%	0.0%	0.1%
Riding elevator down	3.4%	14.4%			0.2%	0.2%		0.1%	81.3%	0.3%
Riding elevator up	3.2%	10.0%	0.1%		1.0%			0.4%	0.4%	84.8%

Table 3: Precision and recall numbers for the decision stumps classifier. A randomly selected 20% set of data was set aside and used for the test results here. Overall accuracy is 91%.

Precision

Labeled Activities \ Classified Activity (by HMM)	Sitting	Standing	Walking	Jogging	Walking up stairs	Walking down stairs	Riding a bicycle	Driving car	Riding elevator down	Riding elevator up
Sitting	89.8%	38.5%	0.5%				0.4%	33.4%		
Standing	10.1%	50.8%	1.4%							
Walking	0.1%	7.4%	97.7%		5.2%	2.5%				
Jogging				100.0%						
Walking up stairs					94.8%					
Walking down stairs			0.5%			97.5%				
Riding a bicycle		3.3%					99.6%			
Driving car								66.6%		
Riding elevator down									100.0%	
Riding elevator up										100.0%

Recall

Labeled Activities \ Classified Activity (by HMM)	Sitting	Standing	Walking	Jogging	Walking up stairs	Walking down stairs	Riding a bicycle	Driving car	Riding elevator down	Riding elevator up
Sitting	87.5%	3.7%	0.1%				0.1%	8.6%		
Standing	65.6%	32.8%	1.6%							
Walking	0.4%	4.0%	93.8%		1.3%	0.4%				
Jogging				100.0%						
Walking up stairs					100.0%					
Walking down stairs			2.5%			97.5%				
Riding a bicycle		1.7%					98.3%			
Driving car								100.0%		
Riding elevator down									100.0%	
Riding elevator up										100.0%

Table 4: Precision and recall numbers for the HMM classifier, using posterior probabilities as inputs. Overall accuracy is 95%.

	Naïve Bayes	Decision Stumps	HMM with Decision Stumps	HMM with Raw Features
Precision	92%	98%	99%	54%
Recall	45%	84%	91%	40%

Table 5: Overall precision and recall numbers for the various generative and discriminative classifiers, used in evaluating our system.

significantly worse, even worse than the static classifier, demonstrating the importance of discriminative classifiers in distinguishing between activities (see table 5 for a comparison of the overall precision and recall numbers for the various classifiers used in our experiments). We recognize that the modeling of more complex activities may require a generative model of personal behavior. However, we believe that discriminative classifiers that map sensor data into primitive activity classes will reduce a large amount of the sensor noise and allow us to learn complex behaviors more effectively.

6 Conclusion

The problem of recognizing human activities from sensor data presents diverse statistical challenges: different classes of actions need to be actively distinguished from each other; and a model needs to incorporate the fact that people's actions are extended over time. In the present study, we approached these problems by combining the discriminative power of an ensemble of decision stumps with the generative and temporal powers of HMMs.

As our presented results have shown, the combination of discriminative and generative classifiers is more effective than either of the classifiers on their own. Not only does the HMM on-top of the discriminative classifier perform better than the discriminative classifier on its own, but it also produces very smooth and accurate outputs as Figure 3 shows.

Feature selection plays an important role in our system not only in improving the performance of our classifier but also in creating a practical system. Our selection of the best 50 features for our top classes leaves us with 242 features from our original 651. This reduces the number of features necessary by more than 60%, which is a significant computational saving. This savings could be further improved upon by optimizing the calculation of the features to take advantage of the new subset, for example there are efficient techniques to obtain FFT coefficients only for required sub-bands [Goertzel, 1958]. In addition, this subset could be further reduced by allowing the feature selection process to determine a more optimal stopping point.

Thus, our work lays out several directions in which the automatic recognition of human actions can be pursued further. First, multi-modal wearable sensors provide a cheap, lightweight and unobtrusive means of obtaining richly detailed data in unconstrained environments, and over long periods of time. Second, the very richness of such sensor readings, and the mass of data collected, demand that suitable preprocessing and data-reduction techniques be applied. We found that by selecting the most informative features, computational costs could be cut by more than 60%, although still greater savings are probably attainable. Third, the multi-faceted nature of human activities presents opportunities for multiple machine-learning approaches to be combined, with the complementary strengths of different approaches (in this instance, boosted decision stumps and HMMs) meeting different aspects of the computational challenge. Our initial studies, presented here, have yielded high recognition rates, suggesting that this is a fruitful approach. Our future work will focus on incorporating and building on the techniques presented here to recognize more complex behavior patterns (e.g. *cooking*, *cleaning*, etc.) Further development of these ideas will, we hope, lead to activity recognition systems that can move from beyond the research lab out into the real world, offering applications in areas as diverse as smart rooms, ethnography, and health care for both the young and aging population.

Acknowledgments

The authors would like to thank Dieter Fox for the helpful comments and discussions. We also thank the two undergraduate students who collected many hours of data that was used our experiments.

References

[Bao and Intille, 2004] L. Bao and S. Intille. Activity Recognition from User-Annotated Acceleration Data. *Proc. Pervasive*, 1-17, Vienna, Austria, 2004.

[Clarkson and Pentland, 1999] B. Clarkson and A. Pentland. Unsupervised Clustering of Ambulatory Audio and Video. *Proceedings of the International Conference of Acoustics, Speech and Signal Processing*, Phoenix, AZ, 1999.

[Gavrila, 1999] G. Gavrila. The visual analysis of human movement: A survey. *Computer vision and Image Understanding* 75(1), 1999.

[Goertzel, 1958] G. Goertzel. An Algorithm for the Evaluation of Finite Trigonometric Series. *American Mathematical Monthly* 65, 1958.

[Jaakkola and Haussler, 1999] T. Jaakkola and Haussler. Exploiting generative models in discriminative classifiers. *In Advances in Neural Information Processing Systems*, 1999.

[Kern *et al.*, 2003] N. Kern, B. Schiele, *et al.* Multi--Sensor Activity Context Detection for Wearable Computing. *Proc. EUSAI, LNCS*, 220-232, Eindhoven, The Netherlands, 2003.

[Lukowicz *et al.*, 2002] Paul Lukowicz, H. Junker, *et al.* WearNET: A Distributed Multi-sensor System for Context Aware Wearables. *Proceedings of the 4th international conference on Ubiquitous Computing*. Göteborg, Sweden, Springer-Verlag: 361-370, 2002.

[Lukowicz *et al.*, 2004] Paul Lukowicz, Jamie A. Ward, *et al.* Recognizing Workshop Activity Using Body Worn Microphones and Accelerometers. *Pervasive Computing: Proceedings of the 2nd International Conference*, 2004.

[Ng and Jordan, 2002] A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. *In Advances in Neural Information Processing Systems*, 2002.

[Oliver *et al.*, 2002] N. Oliver, E. Horvitz, *et al.* Layered representations for human activity recognition. *Fourth IEEE Int. Conf. on Multimodal Interfaces*, 2002.

[Patterson *et al.*, 2003] D. Patterson, L. Liao, *et al.* Inferring high-level behavior from. *In the Proceedings of the Fifth Annual on Ubiquitous Computing*, Seattle, Washington, USA, 2003.

[Pentland, 1996] A. Pentland. Smart Rooms. *Scientific American* 274(4):68-76, 1996.

[Platt, 1999] J. Platt. Probabilities for SV Machines. *Advances in Large Margin Classifiers*. A. Smola, P. Bartlett, D. Scholkopf and D. Schuurmans, MIT Press, 1999.

[Rubinstein and Hastie, 1997] Y. D. Rubinstein and T. Hastie. Discriminative vs. informative learning. *In the Proceedings of Knowledge Discovery and Data Mining*, 49-53, 1997.

[Schapire, 1999] R. E. Schapire. A brief introduction to boosting. *Sixteenth International Joint Conference on Artificial Intelligence*, 1401-1405, 1999.

[Schapire *et al.*, 1997] Robert E. Schapire, Yoav Freund, *et al.* Boosting the margin: a new explanation for the effectiveness of voting methods, 322--330, 1997.

[Viola and Jones, 2001] Paul Viola and Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. *Computer Vision and Pattern Recognition*, 2001.

[Yin *et al.*, 2004] P. Yin, I. Essa, *et al.* Asymmetrically Boosted HMM for Speech Reading. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, pp II755-761, Washington DC, USA, 2004.