Inferring Identity using Accelerometers in Television Remote Controls

Keng-hao Chang∗, Jeffrey Hightower†, and Branislav Kveton‡

∗University of California, Berkeley, California, USA
†Intel Research, Seattle, Washington, USA
‡Intel Research, Santa Clara, California, USA

Abstract. We show that accelerometers embedded in a television remote control can be used to distinguish household members based on the unique way each person wields the remote. This personalization capability can be applied to enhance digital video recorders with show recommendations per family-member instead of per device or as an enabling technology for targeted advertising. Based on five 1-3 week data sets collected from real homes, using 372 features including key press codes, key press timing, and 3-axis acceleration parameters including dominant frequency, energy, mean, and variance, we show household member identification accuracy of 70-92% with a Max-Margin Markov Network (M^3N) classifier.

1 Introduction

Personalizing the television watching experience has become a hot topic as service providers, content creators, and consumer electronics manufacturers all search for ways to expand their user-base, provide exciting and relevant programming, increase the effectiveness of advertising [1], incorporate digital home technologies like interactive TV [2], and distinguish their devices' features and usability. Most personalized capabilities and services are not possible, however, without first knowing who is watching TV. The work presented in this paper addresses this challenge of distinguishing between television watchers in a household.

Our new method of distinguishing TV viewers applies the lightweight biometric of analyzing people's hand motions and button press sequences on remote controls. This method is effective yet simple enough to be invisible and embedded pervasively. Users can simply grasp the remote control as needed and watch TV without any effort to explicitly login or identify themselves. Our system observes people's hand motion in the background and analyzes whether it matches existing signatures.

2 Related Work

There are other research and commercial efforts to develop ways to detect and identify TV viewers. Some existing approaches ask TV viewers to validate their identity explicitly. Digital video recorders such as TiVo^{TM} ask users to enter a user-name and password with an on-screen remote-driven virtual keyboard in order to access some personalized services. Orca Interactive (www.orcainteractive.com) uses a custom remote control device to read users' fingerprints. These logins and cryptographic-grade biometrics have high accuracy (in particular, a low false-positive rate) compared to the sensor-based approaches like the one presented in this paper. They are also quite secure and thus useful for authorizing sensitive transactions like purchases or subscriptions. But logins and intentional actions are cumbersome to perform repeatedly and their prompts almost certainly interfere with natural TV watching behavior. Another approach that is similar to our work uses computer vision for face detection and recognition. Hwang and colleagues' work is a good example of this approach [3]. Similar to our approach, facial recognition can be used to identify people in a "background" fashion, without explicit user input. However, a sense of privacy intrusion can come along with an embedded camera staring at every activity that happens in the livingroom, bedroom, or wherever the TV is positioned.

Outside the television domain, the work most similar to our contribution is that of Hodges and Pollack who showed that users manipulating everyday kitchen appliances (coffee making materials in their experiments) could be distinguished with approximately 77% accuracy based on their patterns of usage [4]. They applied decision trees for their classification, as did we in our initial work before we improved our results using a higher-level sequence information with a Max-Margin Markov Network (M^3N) classifier.

There is also quite a bit of work on combining machine learning with physical sensors to infer what activities someone is engaged in. This activity recognition research is worth mentioning in that it involves similar components to our work—namely, machine learning plus sensors like accelerometers—but it does not specifically focus on determining identity. Three specific examples are Bao and Intille's work using body-worn accelerometers to recognize physical activities [5], Philipose and colleagues' work where participants wear an RFID bracelet to sense which objects they are interacting with to infer their Activity of Daily Living based on web-mined models [6, 7], and Consolvo and colleague's Ubifit persuasive fitness technology [8, 9].

3 Feasibility Study

To consider the feasibility of the entire project, we first conducted a small study to understand how people use remote controls to watch TV. We recruited five of our lab colleagues videotaped them watching TV and channel surfing. We found some interesting patterns in the video recordings, which not only made us feel more comfortable to proceed in the research, but also inspired some of our feature selection approach.

Remote Control Orientation One participant, shown in Fig. 1b, did not hold the remote control horizontally while switching channels. Another participant did not aim the remote control at the TV when switching channels.

(a) (b) (c)

Fig. 1: Snapshots of different hand motion patterns as captured in our plausibility study. In comparison to (a), the participant in (b) holds the remote with different orientation, and the participant in (c) leans his arm on the sofa, which stabilizes his movements.

Physical Support Some participants put their hands on the sofa arm, as shown in Fig. 1c, which stabilized their hands and induced less vibration on the remote control. Some participants usually put their arm on their lap. Another participant regularly left the remote control directly on the coffee table and switched channels without holding it.

Shaking while Surfing One participants tended to shake the remote control in a seemingly unique way while surfing. Specifically, between each button press, he wiggled the remote while deciding whether to switch to the next channel.

Based on these high-level observations, we hypothesized that hand motion pattern might be distinguishable if we look at acceleration features *before*, *during*, and *after* each button press. Features before the press roughly capture the hand motion when the remote control was picked up or held between surf actions; features during the press capture distinctiveness in the orientation of remote as well as the dynamics of actually pressing the buttons; features after a press capture how the remote falls back to the arm, couch, lap, or table. We use these characterizations simply as a principled way to create features from the data stream for use by the machine learning algorithms. As such, it is not important that these descriptions are precise or exactly capture how all people use remotes.

In addition to our motion pattern observations, the ethnographic literature, specifically the work of Langan, revealed that females tend to switch to a planned channel by channel numbers while males tend to surf channels more using channel up and down buttons [10]. Though this work predates innovations like on-screen program guides and digital video recorders, which may alter or nullify some of the potential gender differences, it nonetheless led to our second hypothesis that capturing data about which keys were pressed and in which sequence may be valuable. Again, the veracity of these ethnographic claims is probably not critical since we use them simply as principles to justify including button press and button timing information as features for the machine learning algorithms.

4 Data Collection

We created the hardware and software needed to record the acceleration forces imposed on remote controls and also to capture the button presses. We used this capability to conduct a real-world data collection study in five households.

4.1 Hardware and Software

We wanted to have no dependency on a particular brand of remote or type of TV source (e.g. cable, fiber, satellite, broadcast) so we could collect data on the devices already owned by our participants. Thus, we designed our data collection hardware to easily integrate into a variety of TV setups. The hardware and software components are described below and shown in Fig. 2.

Fig. 2: Data Collection Components: top, a logging laptop; bottom from left to right, a video camera pointed at the room to gather ground truth about who was watching TV, a remote control with our accelerometer module attached, a universal infrared code receiver.

Accelerometer Module Our 3-axis accelerometer module can be attached and wired into the power source of any remote control. In our deployments, we would purchase the same model remote control used by the household and modify it to attach our accelerometer module. At the conclusion of the study we would return the household's original remote. The accelerometer module continuously measures and transmits all the acceleration forces imposed on the remote control. The module hardware is a custom 3-axis accelerometer board connected to a Telos sensor mote [11], which acts as a relay to transmit the data to the logging laptop. The module is enclosed in a custom plastic case. We optimized the module to use as little power as possible and found that on most remote controls it would last 2-3 days while continuously recording data.

Infrared Receiver We use the Tira-2.1 multi-protocol infrared receiver made by HomeElectronics. In our deployments, we placed this receiver by the TV to capture the infrared signals caused by button presses on the remote control. Each infrared code is timestamped and logged by its unique ASCII code string. Infrared remotes will transmit the same signal several times (typically three times for a "normal" button press) to make sure the TV receives the signal, or continuously if the participant keeps pressing a button.

Logging Laptop Computer A laptop computer receives and logs the acceleration and button press data streams. In our deployment it would be placed next to or behind the participants' TV operating with its lid closed. Acceleration data is wirelessly transmitted to the laptop through another paired Telos mote plugged into the laptop's USB port. The infrared code stream is received through a direct USB connection to the Tira infrared receiver. All data is timestamped with a precision of 100ns.

Video Camera The last component is a video camera pointed at the room where the TV is located. The logging laptop automatically starts capturing a video clip whenever it receives an infrared button press and stops encoding the clip after 10 seconds without any additional button presses. We use these video clips in our experiments to hand-label ground truth about who was watching TV. To give participants control over their privacy, before returning the data collection system to the researchers, household members were given a way to access and review the video files in rapid playback to delete any video clips they did not wish to share with us. We omitted this data from our experiments.

4.2 Data Collection Study

We conducted a real-world data collection study in five households in Seattle metropolitan area of the United States. The households were recruited through one of their members working with us as colleagues. Doing completely outside recruiting seemed unnecessary for this study because all members in each household except one were not familiar with the project. Furthermore, we did not see a significant potential for bias even in the one member of each house who is our colleague since manipulating a TV remote is a simple physical activity. Everyone in the households already knew how to use a remote control (except one child who was very young and has not yet learned to use a remote) and there is no additional learning curve added by our technology, therefore there is no potential bias where someone who is technically trained might be able to learn our technology more quickly.

The composition of the five households are different: the first household is a four person family with two parents, a pre-teen, and a teen; the second is a couple; the third is a couple with a child who is too young to use remote controls; the fourth is a house with two graduate student roommates; the fifth is a large house with four graduate student roommates.

We asked each household to simply watch TV as they usually do while having the data collection system installed for one to three weeks. The system collected data 24 hours a day. Since we knew our sensor module mounted on participants' remote controls would last around 3 days when wired into a typical remote, we gave each household several extra sets of batteries and instructions to replace the batteries in their remote "every other day or whenever they thought about it." These informal instructions were sufficient in that we only saw one dropout in the data across all the households due to a battery dying. Even in this case we probably did not miss any data because the participant told us that she immediately replaced the batteries when she realized they were drained, which is not surprising because her remote would not work without fresh batteries since it shared power with our module.

We wanted to collect three weeks of data from each household, but the amount of time we collected data in each house varied between one and three weeks due to participants' summer vacation schedules, limited data collection hardware (we built two complete data collection rigs), and one mother who stopped participating after two weeks when she decided that the family had watched enough television for the summer and needed to spend more time on other pursuits.

5 Experimental Method

By iteratively adding features and analyzing their performance, we settled on a two-level classification technique: **button-press-level classification** and **session-level classification**, the former containing *motion-features* and *button-features* and the latter using motion-features and *inter-button-features*. Features for button-press-level classification occur before, during, and after a single button press. In this classifier, an inference about who is using the remote control is computed with each button press. At the higher level, session-level classification aggregates a sequence of button-press-level classifications and also has additional features that describe the longer sequence of button presses (e.g. the histogram). In this classifier, an inference about who is using the remote control is computed at the end of each session window. We will evaluate classification accuracy at both levels. Our feature extraction routines are implemented in MATLAB.

5.1 Button-Press-Level Classification Features

Classification at the button-press-level makes use of motion-features and button-features. Motion-features are computed from the accelerometer data. Twelve different time-windows (three types with four lengths of each type) are first located in the data stream around the current button press at time t. The three window types are *preceding*, *centered*, and *succeeding* capturing hand motion before, during, and after a button press, respectively. The preceding window has the right end point located at $t - 0.5$ seconds, the midpoint of the centered window is anchored at t seconds, and the start point of the succeeding window is positioned at $t + 0.5$ seconds. The four window lengths are $\{0.5, 1, 2, 4\}$ seconds. The same set of features is computed for each of the 12 windows. The reason for having 12 windows is that the window size and type may influence the value of a extracted feature and, since we do not a priori know the best choice, we generate a variety and and let the classification algorithms decide the utility of the features by assigning them importance weights.

For each of the twelve windows we compute the (1) energy, (2) dominant frequency, (3) magnitude of the fundamental frequency, (4) mean, (5) variance, (6) maximum, (7) minimum, (8) median (9) range, and (10) correlation coefficient. The first nine features are extracted for each of the x, y, and z axes of the accelerometer. Energy, describing the total amount of hand motion, is calculated by the squared sum of the results of Fast Fourier Transform (FFT) with the DC component excluded. The fundamental frequency is defined as the frequency with the highest magnitude from the result of FFT (again, with DC removed), which provides information about shaking. The mean in three axes serves as an indicator of the remote control's orientation. In addition, the correlation coefficient is extracted from each of the x-y, y-z, x-z axis pairs, calculated by $(\Sigma a_i b_i - \overline{a}\overline{b})/((n-1)\sigma_a \sigma_b)$ where a and b are sequences of n measurements with mean $\overline{a}$ and $\overline{b}$ and standard deviation σ_a and σ_b.

Button-features used by the button-press-level classifier include (1) the infrared code of the button press signal, (2) the number of times the code was sequentially transmitted, (3) the approximate duration of the key press, and (4) a time-of-day to let the classification algorithms take into account habits of when particular people in a household watch TV in a day. In Section 4, we mentioned that a button signal may repeat if a participant keeps pressing the button. Therefore, we merge multiple sequential button presses into one classification step to create the the second and third features, which serve as an approximation of the button press duration.

5.2 Session-Level Classification Features

Patterns or frequency in a series of motions and button presses may also benefit user identification. On first glance, however, there is a "chicken and egg" problem: we want to extract features from consecutive button presses to identify a person, but we do not know whether a given sequence of button presses were made by the same person. Fortunately, a little domain knowledge gives us an effective heuristic: if there is continuous acceleration imposed on a remote control, then the same person is holding the remote during this period of time. While no heuristic is ever completely correct, this one turns out to be both reasonable and effective in practice. In examining the video clips we captured to hand-code the ground truth, we never saw someone operate the remote and then hand it directly to someone else to operate. Therefore we look for periods where the remote control is stationary, specifically where the energy of acceleration drops to near 0 for for more than s seconds, and label this point as a session boundary and possible transition between users. Based on video analysis, we set $s = 8$ and the found that the heuristic effectively segmented the data such that over 98% of the sessions were indeed occupied by the same person. Using these heuristic-derived session boundaries, we can now calculate the following features session-level features:

Motion-Features The same set of motion-features described in Section 5.1 are also calculated over the entire session. This captures hand motion in a "macro" view, spanning across multiple button presses.

Inter-Button-Features We generate features about all the button presses in a session including: (1) the number of button presses, the (2) mean and (3) variance of the intervals in between button presses, and (4) a histogram (count of appearance) of button presses in the session. The first three features indicate the frequency of pressing button behaviors, and the histogram shows the habit which buttons are used more often than the others.

6 Results

We use WEKA [12], a popular suite of machine learning software from the University of Waikato, to test the performance of several machine learning methods including Naive Bayes Classifier, C4.5 Decision Tree, Random Forest, and Linear Support Vector Machine (SVM). Our evaluations use ten-fold cross validation. To bring more realism to our results, cross validation is done over large contiguous blocks of time. For example, dividing the data at the individual button-press level to evaluate button-press-level classification would artificially boost accuracy, so we instead divide 5 days worth of data into 10 half-day blocks.

6.1 Button-Press-Level Classification

Table 1 shows the results for identifying users at the granularity of single button presses. We found few accuracy differences between the various machine learning algorithms on these data sets so we report Random Forest results for all data sets. Table 1 also shows statistics about each data set including the number of participants in the household, the total number of button presses recorded, the distribution among participants, and the baseline. Since some members of a household will watch more TV than the others, the baseline is the accuracy that would result if an oracle knew which person in the household pressed the most buttons on the remote control and always reported them as the answer. This baseline oracle would achieve decent accuracy, but a poor F-measure. In general, our classification accuracy is 12% better than the baseline (a 35% relative improvement) with classification at button-press granularity.

Table 1: Accuracy of button-press-level identity prediction using a Random Forest learning algorithm.

	Statistics				Results	
Household	#Participants	#Presses	Distribution	Baseline	F-Measure	Accuracy
1	4	8756	0.63/0.33/ 0.04/0.01	62.8%	0.80/0.52/ 0.22/0.87	70.96%
2	2	695	0.82/0.18	81.6%	0.93/0.55	87.48%
3	3	122	0.25/0.75/ N/A	74.6%	0.68/0.89/ N/A	83.61%
4	2	834	0.72/0.28	72.4%	0.90/0.65	84.53%
5	4	1240	0.36/0.15/ 0.07/0.42	42.0%	0.69/0.39/ 0.74/0.70	66.45%

6.2 Session-Level Classification

The accuracy of the button press-level result was encouraging, so we believed that accuracy could be improved further by incorporating temporal relationships between button presses. Our first attempt was to simply smooth over sequential button-press-level estimates to try and exploit the theory that two button press events that happen closely in time are more likely to be made by the same person. This simple smoothing approach did not boost accuracy, however, because when the classifier made a mistake, the confidence measure for the wrong decision was still high making filtering or majority voting work poorly. Instead we adopted a more principled way to improve accuracy using the previously described session-level features.

At the session level, we trained two machine learning classifiers: Linear SVMs and Max-Margin Markov Networks (M^3Ns) [13]. The features consist of the session-level features described in Section 5.2. The SVMs predict using a linear combination of features. The M^3Ns extend the SVMs such that they capture the temporal relationship between consecutive sessions. More specifically, for M^3Ns we model the fact that the same user usually uses the remote control several session in a row. Note that at the time we report these results WEKA does not support the M^3N graphical model so we implemented this algorithm ourselves. As shown in Table 2, this more complex approach leads to better results than SVMs—Linear SVMs are on average 11% better than the baseline (a 30% relative improvement) and the M^3Ns are on average 17% better (a 46% relative improvement), which is 6% better than the SVMs. Accuracy for the third household does not show a large improvement because the training set is extremely small relative to the other 4 households. Insufficient training data always impacts accuracy in any supervised machine learning. Even with a tiny training set, the accuracy is still no worse than the baseline oracle and improves slightly using the M^3N graphical model.

Though it is tempting to do so, the values in Table 1 and Table 2 are not directly comparable because partitioning button presses into sessions changes

Table 2: Accuracy of session-level identity prediction. Household three's accuracy does not show a large improvement due to insufficient session training data.

	Statistics			Accuracy	
Household	#Participants	#Sessions	Baseline	SVM	M^3N
1	4	458	53.9%	61.79%	69.87%
2	2	124	76.6%	90.32%	91.94%
3	3	28*	75.0%	75.00%	78.57%
4	2	90	65.6%	81.11%	88.89%
5	4	340	44.1%	61.78%	72.06%

the nature of the problem—specifically, the "baseline oracle" in the session case knows which person in the household watches the most total TV instead of which person pressed the most buttons on the remote. With this difference in mind, we can conclude that temporal modeling and session-level classification does indeed offer an improvement over button-level classification because the former shows a 17% improvement over its baseline while the latter has an improvement of only 12% over its baseline (the relative improvement is even greater at 46% versus 35%). Session-level classification is also more realistic because it more closely matches the ways people actually use and share remote controls.

6.3 Feature Evaluation

The Linear SVMs trained in section 6.2 provide a way to evaluate the importance of features because an SVM assigns importance weights to its features for class prediction. The prediction is made by weighted linear combination of features, i.e. $y = \arg\max_c \sum_k w_{ck} x_k + b_c$. We can think of the weight w_{ck} as a vote assigned to a particular feature x_k. The feature values themselves are normalized in their variance to a value between 0 and 1.

The first analysis is to look at how many features are actually important to the classification. The classifiers were given 372 different features as input, but, as the rank-order weight plot in Figure 3 shows, only about 10-20 features have high weight after feature selection. These features contain most of the classification power for that particular participant.

The rank-order weight analysis in Figure 3, however, does not reveal the features' variance, i.e. whether the set of highly weighted features is the same or different across participants. To illuminate this issue, Tables 3, 4, 5, and 6 break down the ten most indicative features for each participant for households 1, 2, 4 and 5, respectively (household 3 is excluded excluded due to its insufficient data). Each feature has in parentheses its level (<u>S</u>ession or <u>B</u>utton), followed by a dash, followed by its feature type (Hand <u>M</u>otion or <u>B</u>utton Press Feature). For category B-M, the feature is abbreviated as feature_windowType_windowLength_axis, e.g. fundamental frequency extracted in center window of length 2 in y-axis as fund-

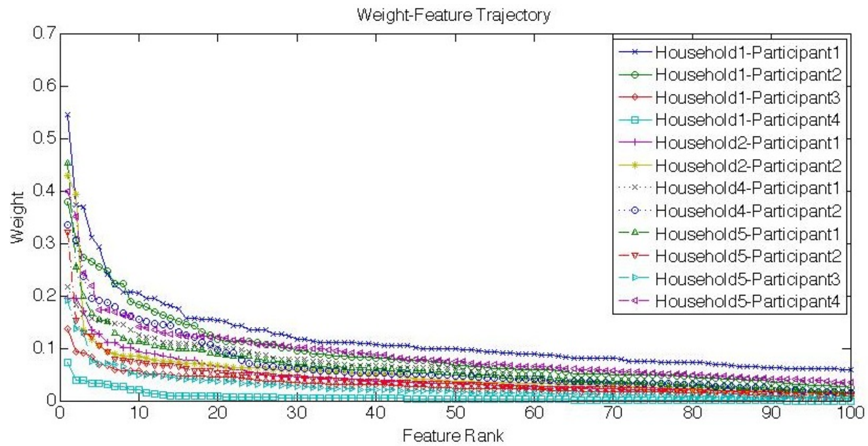

Fig. 3: Weights of the Linear SVM's features for different participants plotted in rank order shows that, although there are 372 features input to the classification algorithms, a small set of them are selected because they contain most of the classification power for a particular person.

Freq_center_2_y. Similarly, for category S-M, the feature is abbreviated as feature_axis, e.g. maximum in the session window for the y-axis is max_y. In addition, buttons and their codes are hashed into integers to be uniquely identified. Finally, there is a special category B/S-B because, for classification at session granularity, aggregating individual button presses in a session actually generates a button press histogram that spans features in both session and button level.

Looking across households we can see that the highly weighted features are definitely not identical but there are some similarities and frequently occurring features. For the button-features, the code of the button press signal appears the most, which indicates the habit of pressing certain button is a good indicator of who is using the remote. For example, one TiVo^{TM} user in the household may avoid commercials with the skip-forward-30-seconds button while another always presses the fast-forward arrow. Unsurprisingly, the session-level inter-button-press histogram feature also shows up, indicating that the count and sequence of button presses is also a good discriminator of users. The frequency of pressing buttons are also distinguishing in some cases. Motion-features at both the button-press-level and session-level are selected. In particular, the fundamental frequency, magnitude, and energy are reported several times, meaning the shaking remote behavior is a distinctive pattern in some cases. The maximum, minimum, and mean features are listed, showing the orientation of a remote control can be somewhat indicative. In addition, windowing acceleration with different types and lengths also helps. In general this analysis gives us reassurance about the results since the selected features seem to match our intuitive ideas about which features would be useful. Although it does not seem to be the case that a particular subset of features is universally useful for all households,

Table 3: Top features for predicting participant identify in Household 1

Participant 1			Participant 2		
Category	Feature	Weight	Category	Feature	Weight
(B/S-B)	button code/histogram 16	0.545	(B/S-B)	button code/histogram 25	0.379
(B/S-B)	button code/histogram 15	0.370	(S-M)	correlation_xy	0.307
(B/S-B)	button code/histogram 26	0.370	(B/S-B)	button code/histogram 18	0.274
(B-M)	fundFreq_center_2_y	0.312	(B-M)	fundFreq_center_0.5_z	0.266
(B-M)	fundFreq_center_1_y	0.294	(B/S-B)	button code/histogram 24	0.255
(B-M)	range_suceeding_1_z	0.242	(B/S-B)	button code/histogram 22	0.248
(B-M)	fundFreq_center_2_x	0.221	(B-M)	energy_center_4_y	0.224
(B/S-B)	button code/histogram 6	0.207	(B-M)	magnitudeFundFreq_center_4_y	0.224
(B-M)	magnitudeFundFreq_center_4_z	0.207	(B-M)	correlation_center_1_xz	0.190
(S-M)	correlation_xz	0.205	(B-M)	energy_center_2_y	0.183
Participant 3			Participant 4		
Category	Feature	Weight	Category	Feature	Weight
(B-M)	fundFreq_center_4_z	0.138	(B-M)	correlation_center_0.5_xy	0.074
(B/S-B)	button code/histogram 5	0.094	(B/S-B)	button code/histogram 18	0.040
(B-M)	max_preceding_4_y	0.091	(S-M)	fundFreq_z	0.040
(B-M)	correlation_succeeding_1_xz	0.083	(B/S-B)	button code/histogram 22	0.034
(B/S-B)	button code/histogram 27	0.070	(B/S-B)	button code/histogram 6	0.033
(B-M)	max_preceding_2_y	0.067	(S-M)	max_y	0.031
(B/S-B)	button code/histogram 13	0.059	(B/S-B)	button code/histogram 5	0.028
(B-M)	magnitudeFundFreq_center_2_z	0.058	(B/S-B)	button code/histogram 28	0.027
(B-M)	correlation_succeeding_2_xz	0.057	(B/S-B)	button code/histogram 7	0.022
(B-M)	fundFreq_center_0.5_y	0.057	(S-M)	range_y	0.022

Table 4: Top features for predicting participant identify in Household 2

Participant 1			Participant 2		
Category	Feature	Weight	Category	Feature	Weight
(B/S-B)	button code/histogram 123	0.196	(B/S-B)	button code/histogram 143	0.430
(S-M)	correlation_xy	0.195	(B/S-B)	button code/histogram 142	0.393
(B/S-B)	button code/histogram 115	0.172	(B-M)	correlation_succeeding_1_xy	0.136
(S-B)	number of presses	0.136	(B-M)	fundFreq_center_y_2	0.117
(B/S-B)	button code/histogram 119	0.128	(S-M)	min_x	0.106
(B/S-B)	button code/histogram 121	0.111	(S-M)	energy_x	0.093
(S-M)	correlation_yz	0.111	(B-M)	range_succeeding_2_y	0.089
(S-M)	range_x	0.102	(B-M)	max_succeeding_4_x	0.085
(B-M)	correlation_succeeding_2_xz	0.102	(B-M)	max_succeeding_2_x	0.085
(B-M)	correlation_center_1_xy	0.093	(B-M)	var_succeeding_2_y	0.085

Table 5: Top features for predicting participant identify in Household 4

Participant 1			Participant 2		
Category	Feature	Weight	Category	Feature	Weight
(B-M)	fundFreq_center_0.5_x	0.217	(B-M)	fundFreq_center_0.5_z	0.335
(B-M)	energy_center_4_z	0.184	(B/S-B)	button code/histogram 118	0.306
(B/S-B)	button code/histogram 181	0.165	(B/S-B)	button code/histogram 174	0.237
(B-M)	energy_center_0.5_x	0.157	(B/S-B)	button code/histogram 212	0.195
(S-M)	correlation_xz	0.153	(B-M)	magnitudeFundFreq_center_2_z	0.192
(B-M)	energy_center_0.5_z	0.151	(B-M)	energy_center_0.5_y	0.188
(S-M)	var_x	0.147	(B-M)	energy_center_2_y	0.179
(S-M)	energy_z	0.145	(S-M)	min_z	0.168
(S-M)	range_z	0.136	(B-M)	magnitudeFundFreq_center_1_z	0.162
(S-B)	number of presses	0.125	(B-M)	energy_center_1_y	0.156

Table 6: Top features for predicting participant identify in Household 5

Participant 1			Participant 2		
Category	Feature	Weight	Category	Feature	Weight
(B/S-B)	button code/histogram 226	0.453	(B/S-B)	button code/histogram 308	0.321
(S-M)	correlation_xz	0.255	(B/S-B)	button code/histogram 227	0.153
(S-B)	number of presses	0.200	(B-M)	magnitudeFundFreq_center_2_y	0.129
(B-M)	var_succeeding_4_x	0.167	(S-M)	correlation	0.123
(S-M)	variation_x	0.155	(B/S-B)	button code/histogram 159	0.105
(B-M)	range_succeeding_4_x	0.151	(B-M)	range_center_0.5_x	0.094
(S-M)	range_x	0.130	(B-M)	range_center_0.5_y	0.080
(B-M)	fundFreq_center_1_z	0.120	(S-M)	correlation_xy	0.078
(B/S-B)	button code/histogram 148	0.114	(S-M)	med_z	0.076
(S-M)	energy_z	0.113	(B-M)	range_center_0.5_z	0.073
Participant 3			Participant 4		
Category	Feature	Weight	Category	Feature	Weight
(B/S-B)	button code/histogram 308	0.191	(B/S-B)	button code/histogram 152	0.399
(B-B)	button press duration	0.137	(B/S-B)	button code/histogram 151	0.352
(B-B)	button signal repetition	0.132	(S-M)	energy_y	0.244
(B-M)	magnitudeFundFreq_center_1_z	0.076	(S-M)	max_z	0.219
(S-B)	number of presses	0.072	(B-M)	var_preceeding_1_y	0.174
(B-M)	range_center_0.5_x	0.071	(B-M)	energy_center_4_x	0.173
(B-M)	magnitudeFundFreq_center_1_y	0.069	(S-M)	var_z	0.169
(B-M)	magnitudeFundFreq_center_4_z	0.065	(B-M)	fundFreq_center_1_x	0.164
(B/S-B)	button code/histogram 227	0.053	(S-M)	range_z	0.161
(B-M)	range_center_2_x	0.050	(S-M)	mean_z	0.142

from the results shown we can conclude that households members do have sufficiently different behavior combinations such that machine learning methods are able to find a unique feature subset and infer identity.

6.4 Button-Features versus Motion-Features

Tables 3, 4, 5, and 6 reveal a similar number of highly weighted button-press and hand-motion features, which raises additional interesting questions: Do hand-motion or button-press features contribute more to the overall accuracy? Are button-press features alone sufficient to identify users? How does one type of feature complement the other? To answer these questions we ran the experiments again with only hand motion features and with only button press features. The results are shown in Tables 7 and 8.

Table 7: Accuracy comparison with subsets of features of button-press-level identity prediction.

Household	Accuracy with all features	Accuracy with only motion-features	Accuracy with only button-features
1	70.96%	69.82% (↓)	77.58% (↑)
2	87.48%	85.90% (↓)	96.26% (↑)
3	83.61%	96.72% (↑)	87.70% (↑)
4	84.53%	85.61% (↑)	83.21% (↓)
5	66.45%	64.84% (↓)	74.11% (↑)

Table 8: Accuracy comparison with subsets of features of session-level identity prediction.

Household	Accuracy with only all features		Accuracy with only motion-features		Accuracy with only button-features	
	SVM	M3N	SVM	M3N	SVM	M3N
1	61.79%	69.87%	58.52%(↓)	60.70%(↓)	57.86%(↓)	57.21%(↓)
2	90.32%	91.94%	86.29%(↓)	87.90%(↓)	90.32%(-)	92.74%(↑)
3	75.00%	78.57%	78.57%(↑)	78.57%(-)	75.00%(-)	78.57%(-)
4	81.11%	88.89%	77.78%(↓)	81.11%(↓)	63.33%(↓)	63.33%(↓)
5	61.18%	72.06%	54.12%(↓)	57.94%(↓)	43.24%(↓)	44.41%(↓)

Table 7 suggests that for button-press-level classification, button-features alone work as well or better than if motion-features are included. In fact, motion-features seem to drag down the overall accuracy when combined with button-features. Using motion-features alone performs similarly to using all features.

If this were the end of the story, the conclusion would be to abandon the accelerometer hardware and simply use a button press logger as the sole input to the classifier. However, Table 8 shows that there is a different trend in session-level classification—we may not want to give up the accelerometer quite yet. At the session level, only by using both types of features can the system achieve top accuracy. In fact, using button-features alone results in the lowest accuracy, sometimes by a significant margin (except for household 2 where it merely holds even).

Why is this trend different from the one in Table 7? We made a hypothesis that in button-press-level classification there might be more momentary deviation in motion-features. Hence, they worsen the overall accuracy when combined with button-features. In contrast, in session-level classification motion-features are calculated in the span over several button presses, which smoothes out the momentarily noise happened in the button-press level. The motion-features are therefore more informative and contribute the overall accuracy in session-level classification. We can also look at this from another perspective. The session-level motion-features may reveal patterns of moving the remote while pressing a series of buttons, which is less prone to having high variance.

7 Future Work

A potential confounding factor in our results, though not one we believe to be significant, is that our accelerometer module did change the shape of participants' remote controls. Though we always attached the sensor module in a place where it did not interfere with any of the normal hand positions, it may still have changed our participants behavior in some way. Toward this end it would be beneficial in future studies to shrink the acceleration module significantly and embed it inside the void space in the remote's plastic so the overall form of the remote is unchanged.

Though we are pleased with how well acceleration and button presses seem able to identify users, there are many other features, sensors, and sources of information that we would like to add to try and improve the accuracy. For example, hand shape, detected using pressure sensors or capacitive field sensors, may be a very good indicator of who is using the remote. Program guide information is another potential source of input as different people in a household may be attracted to different TV shows or categories of TV show. To test these new sensors and ideas, we plan to follow this work with a longer study of least 8 families over more than 1 month. In this new deployment we will also evaluate an application that makes use of this new personalization capability, specifically a digital video recorder that can recommend TV programs to each household member instead of the control case where recommendations are provided based on the behavior of the entire household.

Television users are probably not willing to go through a training phase where they repeatedly tell the system who is using the remote with each button press. Therefore, our approach would be much more practical if we could apply semi-supervised machine learning to the problem. Specifically, if we could automatically cluster sequences of similar button presses and sessions we could reduce the burden to the point where the user must only be infrequently prompted to verify their identity to provide a training label for the machine learning. The prompts would gradually decrease as the models improved. Even better would be a completely unsupervised learning technique where, given the number of users in a household, the system clusters and learns the models completely on its own, perhaps learning the users' names out-of-band by observing a login name (e.g. when the user is making an online purchase) that correlates with a particular cluster. Semi- and un-supervised learning are only possible if there are sufficient distinguishing features in the data. In the future, we plan to test the feasibility of these approaches on this type of data as well as study the tradeoff between the labeling effort by users and the learning curve of the system.

8 Conclusion

We have built and evaluated the technologies to test the hypothesis that accelerometers embedded in a television remote control can distinguish household members based on the unique way each person wields the remote. Based on real TV watching data collected from five households with 2-4 people of various demographics in each household for 1-3 weeks we achieved user identification accuracy of 70-92% by including both button press-level and session-level features in a Max-Margin Markov Network (M^3N) classifier. We analyzed the feature selection finding that only 10-20 of the 372 features hold most of the distinguishing power for a given household and participant, but the actual features vary somewhat by participant. We also found that button press features alone (without motion data) work well in a simple classifier that triggers with each button press, but to get the greater accuracy from inferring over longer sessions, both button press and motion features are desirable.

Though more accuracy is always better, we believe our results are already sufficient to enable useful TV personalization applications such as improved targeted advertising and digital video recorders that provide program recommendations per user instead of per device. Additional sensors, such pressure sensors or capacitive field sensor to detect users' hand shapes, may boost accuracy even further and a semi-supervised learning approach would make the system more deployable. Ultimately, combining our approach with an existing heavyweight mechanism such as login-password or secure biometrics could result in a complete TV personalization system that is natural and invisible for everyday personalization enhancements while supporting infrequent but authentication-critical situations like financial transactions.

References

1. von Rimscha, M.B., Rademacher, P., Thomas, N., Siegert, G.: The future of TV commercials. In: Annual Meeting of the International Communication Association, San Francisco, CA (May 23 2007)

2. Lu, K.Y.: Interaction design principles for interactive television. Master's thesis, Georgia Institute of Technology (May 2005)
3. Hwang, M.C., Ha, L.T., Kim, S.K., Ko, S.J.: Real-time person identification system for intelligent digital TV. In: Consumer Electronics, ICCE 2007, Las Vegas, NV, Las Vegas, NV (2007) 1–2
4. Hodges, M.R., Pollack, M.E.: An object-use fingerprint: The use of electronic sensors for human identification. In: Proceedings of the Nineth International Conference on Ubiquitous Computing (Ubicomp 2007). Lecture Notes in Computer Science, Springer-Verlag (September 2007) 289–303
5. Bao, L., Intille, S.S.: Activity recognition from user-annotated acceleration data. In: Proceedings of the Second International Conference on Pervasive Computing (Pervasive). (2004) 1–17
6. Patterson, D.J., Fox, D., Kautz, H., Philipose, M.: Fine-grained activity recognition by aggregating abstract object usage. In: ISWC '05: Proceedings of the Ninth IEEE International Symposium on Wearable Computers, Washington, DC, USA, IEEE Computer Society (2005) 44–51
7. Philipose, M., Fishkin, K.P., Perkowitz, M., Patterson, D.J., Fox, D., Kautz, H., Hahnel, D.: Inferring activities from interactions with objects. IEEE Pervasive Computing **3**(4) (2004) 50–57
8. Consolvo, S., McDonald, D.W., Toscos, T., Chen, M.Y., Froehlich, J., Harrison, B., Klasnja, P., LaMarca, A., LeGrand, L., Libby, R., Smith, I., Landay, J.A.: Activity sensing in the wild: a field trial of ubifit garden. In: CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2008) 1797–1806
9. Consolvo, S., Klasnja, P., McDonald, D.W., Avrahami, D., Froehlich, J., LeGrand, L., Libby, R., Mosher, K., Landay, J.A.: Flowers or a robot army?: encouraging awareness & activity with personal, mobile displays. In: UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing, New York, NY, USA, ACM (2008) 54–63
10. Langan, C.R.: A case-study of how people within the same household differ in their use of television. web document: http://www.aber.ac.uk/media/Students/crl9501.html (April 1997)
11. Polastre, J., Szewczyk, R., , Culler, D.: Telos: Enabling ultra-low power wireless research. In: Information Processing in Sensor Networks, 2005. IPSN 2005. (April 2005) 364– 369
12. Holmes, G., Donkin, A., Witten, I.H.: Weka: A machine learning workbench. In: Second Australia and New Zealand Conference on Intelligent Information Systems, Brisbane, Australia (2007)
13. Taskar, B., Guestrin, C., Koller, D.: Max-margin Markov networks. In: Advances in Neural Information Processing Systems 16. (2004)