HomeActivity: Recognizing home activities using sensor data

Stephen Lee, Patrick Pagus and Dong Chen
ML Final Project

December 19, 2015

1 Abstract:

In this project, we user sensors deployed in homes to identify activities. The actuation of sensors provides a unique set of feature representation that allows us to identity the type of activity being undertaken by the user. The goal of our project is to use these sensor information to infer activities under different methodologies and analyze their performance. We use datasets from three different houses that are annotated manually to get the 'ground truth'. To capture the underlying model, we use different feature representations from the sensors. We present our results and discuss our findings.

2 Introduction

According to a recent study, it is estimated that almost 25+ billion Internet of Things (IoT) devices will be deployed across homes. Moreover, around 50 trillion GBs of data will be collected using these sensors. The information collected can be used to enrich the interaction between users and the physical devices. For example, NEST thermostat, an intelligent programmable thermostat, learns the occupancy of humans in a home using sensors to automatically turn on/off heating or air conditioning and save energy costs. These sensor information can be used to learn other information such as monitoring the activities of a person (e.g. elderly home) and being able to detect anomalies in behavior over time. In this project, we focus on recognizing the activities of a person using the sensor information available. i.e. We study different techniques to identify when the person is sleeping, eating etc.

Activity recognition has been a widely studied area and provides a set of challenges in itself. First, ambiguity exists if there are more than one activity happening concurrently at any given time, such as watching TV and eating a snack. Also, this muddles the distinction between the start and end of an activity. Second, activation of a sensor may represent doing a similar activity. For example, opening the fridge could represent 'getting a snack' activity or 'cooking' activity'. Third, the information could be noisy e.g. with misfiring sensors

or a mistakes made by humans such as unintentionally opening a cupboard or entering a room. Fourth, recognizing activities has class imbalance as certain activity labels tend to be longer than others. For example, lying on bed or couch, staying idle, or 'not at home' could easily represent the dominant classes in these datasets.

Clearly, activity recognition is a challenging problem. Thus, we chose a dataset wherein the homes was occupied by a single user. We construct features using the binary sensors installed in these homes that we will describe later in section. In fact, we construct multiple feature representations to study the problem and follow closely the approaches studied by Kasteren [3] among others. In the following section, we will be describing our approach, and discuss its performance and lessons learned from undertaking this project.

3 Related Works

The probabilistic models discussed in our project represent the state of the art models used in activity recognition. Tapia et al. used the naive Bayes model in combination with the raw feature representation on two real world datasets recorded using a wireless sensor network [1]. HMMs were used in work by Patterson et al. and were applied to data obtained from a wearable RFID reader in a house where many objects are equipped with RFID tags [2]. In work by van Kasteren et al. the performance of HMMs and CRFs in activity recognition was compared on a realworld dataset recorded using a wireless sensor network [3]. Duong et al. compared the performance of HSMMs and HMMs in activity recognition using a laboratory setup in which four cameras captured the location of a person in a kitchen setup [4]. One type of model that we have not included in our comparison are hierarchical models. They have been successfully applied to activity recognition from video data [5], in an office environment using cameras, microphones and keyboard input [6] and on data obtained from a wearable sensing system [7]. The related works show part of the models in their work, respectively. It is hard to compare across different works. Instead, we implemented a set of models and ran the experiments on the same real word dataset. The evaluation results could be used as the baseline for future research.

4 Approach

We explore and implement the following techniques to recognize activities:

4.1 Naive Bayes

Naive Bayes model is a simple probabilistic model that assumes independence between every pair of features. The factorization of the joint probability over the datapoints as

follows,

$$p(y_{1:T}, x_{1:T}) = \prod_{t=1}^{T} p(x_t \mid y_t) p(y_t) \tag{1}$$

we apply the naive Bayes assumption, which means we model each sensor reading separately, requiring only N parameters for each activity. The observation distribution therefore factorizes as

$$p(x_t \mid y_t = i) = \prod_{n=1}^{N} p(x_t^n \mid y_t = i) \tag{2}$$

where each sensor observation is modeled as an independent Bernoulli distribution, given by

$$p(x_t^n \mid y_t = i) = {\mu_{ni}}^{{x_t}^n} (1 - \mu_{ni})^{1 - {x_t}^n} \tag{3}$$

4.2 Hidden Markov Models (HMM)

HMM is a generative probabilistic model that models the joint distribution of both the observed and the latent states. It is commonly used in recognizing temporal patterns such as handwriting and speech recognition, where the future state is dependent on the previous state. In our project, the latent variable($\mathbf{y}$) is the activity performed by the user, and the observed variable($\mathbf{x}$) is the vector of sensor readings.

At each time step t, the latent variable y_t depends only on the previous hidden variable y_{t-1} and variables before $t - 1$have no influence on it (Markov Property). The observed variable x_t, at time t, depends only on the latent variable y_t. The parameters of the HMM model are the transition probabilities $p(y_t \mid y_{t-1})$ i.e. the probability of going from one state to another; and the emission probability $p(x_t \mid y_t)$ i.e. the probability that the state y_t would generate observation x_t. In order to learn the parameters of the distribution, one can maximize the joint probability $P(x, y)$ of the observed and latent sequences in the training data. The joint probability can be factorized as follows:

$$P(x, y) = \prod_{t=1}^{T} p(y_t \mid y_{t-1}) p(x_t \mid y_t), p(y_1) = p(y_1 \mid y_0) \tag{4}$$

Since the data is discrete in nature, frequency counting can be used to learn the parameters. We use Viterbi algorithm to infer the sequence of activity labels from the observed sensor reading sequences.

4.3 Conditional Random Fields (CRF)

We use linear-chain CRF to represent the latent and observed variables as it closely resembles the HMM in terms of structure. Since, CRF is a discriminative probabilistic model, we can model the conditional probability distribution $P(y \mid x)$, to predict y from x. Thus,

the parameters are learnt by maximizing the following conditional probability distribution $P(y \mid x)$,

$$p(y \mid x) = \frac{1}{Z(x)} exp\left[\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t)\right] \tag{5}$$

where K is the number of feature functions used to parameterize the distribution, λ_k is the weight parameter and $f_k(y_t, y_{t-1}, x_t)$ is a feature function. The product of the parameters and the feature function $\lambda_k f_k(y_t, y_{t-1}, x_t)$ is called the energy function, while the exponential representation is the potential function. The partition function *Z(x)* is a normalization constant that sums over all the potential functions and is given as follows:

$$Z(x) = \sum_{y} exp\left[\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t)\right] \tag{6}$$

In our approach, we use the BFGS method to learn the parameters of the model.

4.4 Support Vector Machines (SVM)

SVM is a discriminative classifier that builds a hyperplane that can be used for classification or regression purposes. It constructs this hyperplane such that the distance between the hyperplane and the nearest point is maximized. In particular, the problem can be seen as minimizing a loss function that can be represented as following:

$$\min_{w,\beta} L(w) = \frac{1}{2}||w||^2$$
$$\text{subject to } y_i(w^T x_i + \beta) \geq 1 \ \forall i,$$

where x_i are the training examples, y_i represent the labels, w represents the weight vector and β is the bias. In our approach, we use a Linear SVM for classification.

4.5 Structured Support Vector Machines (SSVM)

Structured SVM, as the name suggests, makes use of the structure of the output space for classification purposes. It is generally used for classification where the goal is to predict a sequence as compared to a single label in SVM. The loss function is represented by

$$y^* = \arg\max_{y \in Y} g(x, y) \tag{7}$$

where x is the input, Y is the set of all possible output and g is the loss function given by,

$$g(x, y) = w^T f(x, y) \tag{8}$$

Here, the f is feature function, and we use the linear chain CRF model to represent the feature function i.e. the linear combination of the feature potential(nodes) and the transition potential (edges). The parameters of $g(x, y)$ is learnt by minimizing a loss. We use the libraries provided in *pystruct* for classifying the input variables.

4.6 Experimental Setup

We use http://scikit-learn.org packages to present the models.

4.7 Feather Representation

Raw: the raw sensor representation uses the sensor data directly as it was received from the sensors. It gives a 1 when the sensor is firing and a 0 otherwise

Change: The change point representation indicates when a sensor event takes place. That is, it indicates when a sensor changes value. More formally, it gives a 1 when a sensor changes state (i.e. goes from zero to one or vice versa) and a 0 otherwise.

Last: The last-fired sensor representation indicates which sensor fired last. The sen- sor that changed state last continues to give 1 and changes to 0 when another sensor changes state

5 fold cross validation

Divide the dataset into smaller subsequences of 2 hours

5 Experimental Evaluation

5.1 Datasets

The Kasteren dataset is recording a 26-year-old man. He lives alone in a three-room apartment where 14 state-change sensors were installed. Locations of sensors include doors, cup- boards, refrigerator and a toilet flush sensor. Sensors were left unattended, collecting data for 28 days in the apartment. This resulted in 2120 sensor events and 245 activity instances.

As shown in the table, for Kasteren dataset we have three houses with 14, 23, 21 sensors, respectively.

The tableis an example of referenced LaTeXelements.

5.2 Comparative Analysis Metrics

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

Type	*House A*	*House B*	*House C*
Age	26	28	27
Gender	M	M	M
Setting	Apartment	Apartment	House
Room	3	2	6
Duration(days)	25	14	19
Sensors	14	23	21
Activities	10	13	16
Annotation	Bluetooth	Diary	Bluetooh

Table 1: Dataset recording details

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$F - Measure = \frac{2 \cdot Precision \cdot Recall}{Precison + Recall} \tag{12}$$

The Matthews correlation coefficient is used in machine learning as a measure of the quality of binary (two-class) classifications, introduced by biochemist Brian W. Matthews in 1975. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient between the observed and predicted binary classifications; it returns a value between 1 and +1. A coefficient of +1 represents a perfect prediction, 0 no better than random prediction and 1 indicates total disagreement between prediction and observation.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{13}$$

5.3 Experiments

6 Conclusions and Lessons Learned

References

[1] E. M. Tapia, S. S. Intille, and K. Larson. *Activity recognition in the home using simple and ubiquitous sensors.* In Pervasive Computing, Second International Conference, PERVASIVE 2004, pp. 158175, Vienna, Austria (April, 2004).

[2] D. J. Patterson, D. Fox, H. A. Kautz, and M. Philipose. Fine-grained activity recognition by aggregating abstract object usage. In ISWC,

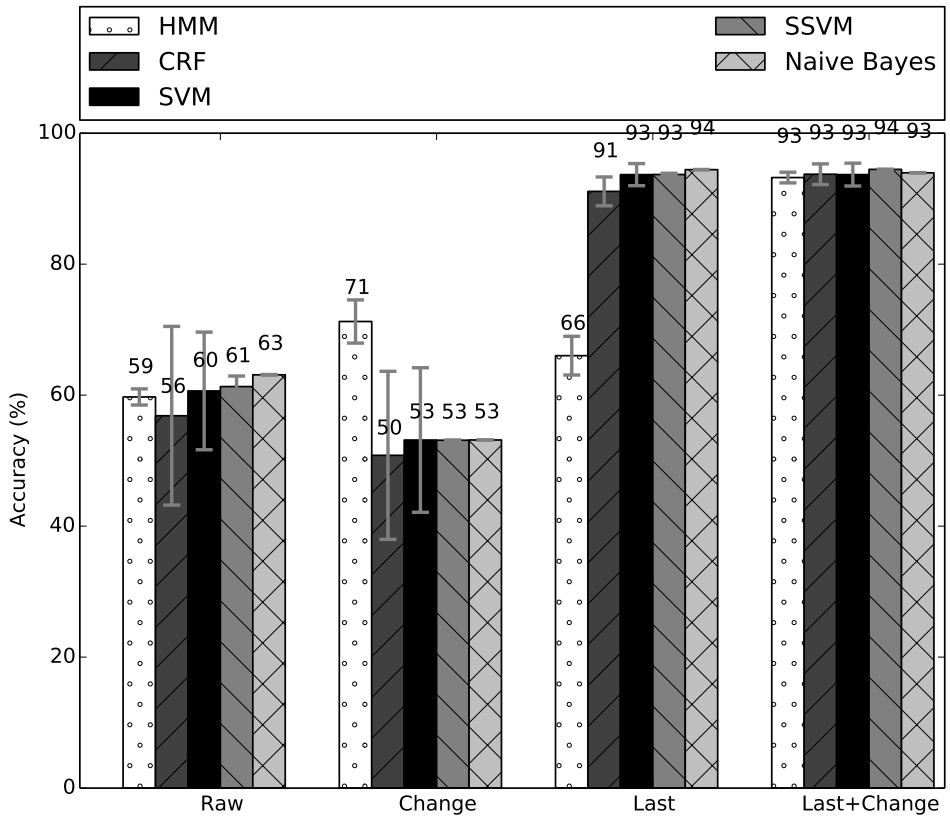

Figure 1: House A

pp. 4451. IEEE Computer Society, (2005). ISBN 0-7695-2419-2. URL http://doi.ieeecomputersociety.org/10.1109/ISWC.2005.22.

[3] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kro se. Accurate activity recognition in a home setting. In UbiComp 08: Proceedings of the 10th international conference on Ubiquitous computing, pp. 19, New York, NY, USA, (2008). ACM. ISBN 978-1-60558-136-1. doi: http: //doi.acm.org/10.1145/1409635.1409637.

[4] T. Duong, D. Phung, H. Bui, and S. Venkatesh, Efficient duration and hierarchical modeling for human activity recognition, Artif. Intell. 173(7-8), 830856, (2009). ISSN 0004-3702. doi: http://dx.doi.org/10.1016/j.artint.2008.12.005.

[5] S. Luhr, H. H. Bui, S. Venkatesh, and G. A. West, Recognition of human activity through hierarchical stochastic learning, percom. 00, 416, (2003). doi: http://doi.ieeecomputersociety. org/10.1109/PERCOM.2003.1192766.

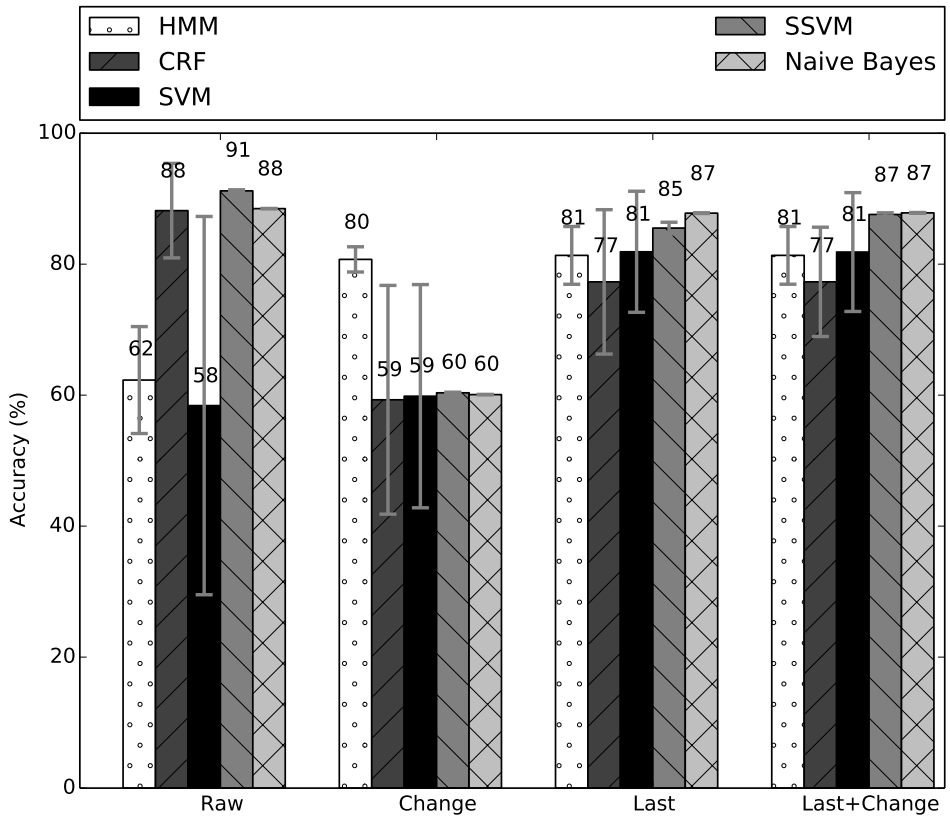

Figure 2: House B

[6] N. Oliver, A. Garg, and E. Horvitz, Layered representations for learning and inferring office activity from multiple sensory channels, Comput. Vis. Image Underst. 96(2), 163180, (2004). ISSN 1077-3142. doi: http://dx.doi.org/10.1016/j.cviu.2004.02.004.

[7] A. Subramanya, A. Raj, J. Bilmes, and D. Fox. Hierarchical models for activity recognition. In IEEE Multimedia Signal Processing (MMSP) Conference, Victoria, CA (October, 2006).

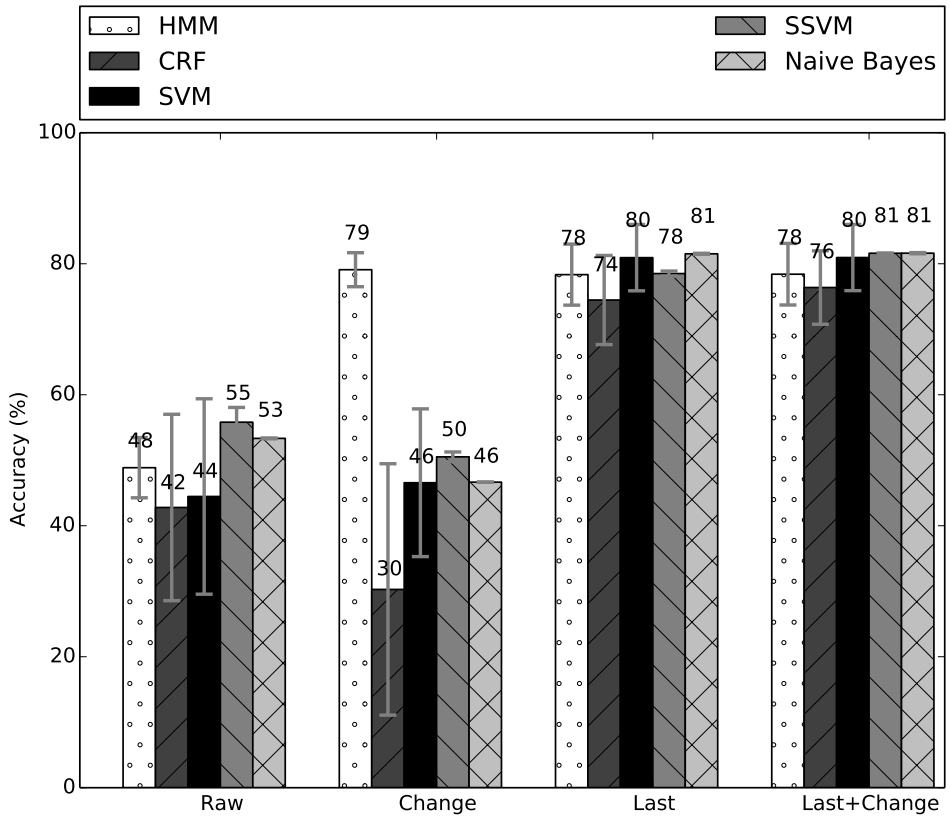

Figure 3: House C