

# HomeActivity: Recognizing home activities using sensor data

Stephen Lee, Patrick Pegus and Dong Chen  
ML Final Project

December 19, 2015

## 1 Abstract:

In this project, we use sensors deployed in homes to identify activities. The actuation of sensors provides a unique set of feature representation that allows us to identify 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 [?] 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

Home occupant activities recognition problem has been studying for several years. In [?], Tapia et al. deployed low-cost sensors to monitoring occupant activities at home. As reported, using these sensors information, they can detect events in toilet and bathroom in real residential places with accuracy ranging from 25 89%. In [?], Patterson et al. collected object data using a glove outfitted with a Radio Frequency Identification (RFID) antenna. Then, they used three models, including training HMMs separately, coupling HMMs into a one large system and using execution history of those objects, to recognize object-interaction based activity in a more realistic setting. And they found that utilizing object history could gains the best accuracy-81.2%. In [?], Kasteren et al. designed easy-to-install wireless sensor network (WSN) and a low-cost but accrete annotation method to automatically recognize activities. Raw, Changepoint, and Last methods are employed to represent sensor information. And utilizing HMM and CRF models for activities recognizing are studied as well. In [?], Duong et al. argue that instead of only using duration pattern, it would be more beneficial to combine inherent hierarchical structures. A switching hidden semi-Markov model is used to empirically compare performance with traditional models. Interestingly, the experiments are tested on a tracking missing and activities overlapping dataset. In [?], Sebastian et al. point out the Hierarchical Hidden Markov Model can capture the natural hierarchy information presentation in home activities when generating models. As shown, they are able to learn two simple activity sequences and capture the hierarchical structure represented in the data. In [?], Oliver et al. use a Layered Hidden Markov Models to infer user activity status from online stream data (e.g. video, audio, keyboard and mouse clicks). Using this cascade of HMMs, they can do sensing, training and predicting at different office data granularity. In [?], Subramanya et al. design a hierarchical model in order to monitoring the motion status and context status for a real person. As shown in the work, breaking a big complex activity into smaller sub-activities to build models is very useful for recognizing person activities. Our project follows closely the approaches presented by Kasteren in work [?] others. We describe the work undertaken

and highlight our findings, issues and limitations that are still required to be overcome in this report.

## 4 Approach

### 4.1 Feature Representation

We describe the feature representation we use for classification purpose:

1. Raw: This uses the binary sensors readings directly. For each time step  $t$ , it takes the value of 1 or 0 depending on whether the sensor is actuated.
2. Change: The feature contains the sensor value as 1 only when there is a change in the state.
3. Last: The feature continues to have the sensor value as 1 until another sensor changes its state.
4. Last + Change: This feature combines the Last+Change representations.

### 4.2 Models

We explore and implement the following techniques to recognize activities:

#### 4.2.1 Naive Bayes

Naive Bayes model is a simple probabilistic model that assumes independence between the features. Under the conditional independence assumption, and using the Bayes rule, the conditional probability can be represented as :

$$P(y_t \mid x_{t1}, \dots, x_{tn}) \propto P(y) \prod_{i=1}^n P(x_{ti} \mid y_t) \quad (1)$$

where  $y_t$  is the activity label, and  $x_t$  is the vector of sensor readings at time  $t$ . The model can then be used to classify the labels as follows,

$$\hat{y}_t = \arg \max_{y_t} P(y) \prod_{i=1}^n P(x_{ti} \mid y_t), \quad (2)$$

In our approach we assume that the distribution of  $P(x_{ti} \mid y_t)$  is modeled by independent bernoulli distribution, given by

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

#### 4.2.2 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  $p(y_t | y_{t-1})$  i.e. the probability of going from one state to another; and the emission probability  $p(x_t | y_t)$  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:

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

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  $P(y | x)$ , to predict  $y$  from  $x$ . Thus, the parameters are learnt by maximizing the following conditional probability distribution  $P(y | x)$ ,

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

where  $K$  is the number of feature functions used to parameterize the distribution,  $\lambda_k$  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:

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

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

### 4.3.1 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:

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

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

### 4.3.2 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

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

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

$$g(x, y) = w^T f(x, y) \quad (8)$$

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.

## 5 Experimental Evaluation

### 5.1 Datasets

Table 1, shows the summary of the dataset we use for our evaluation. We evaluate on 3 homes, each home having sensors deployed .

We evaluate our models on three datasets provided by [?]. Figure 1 displays the floor plans and sensor locations of the three residences

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.

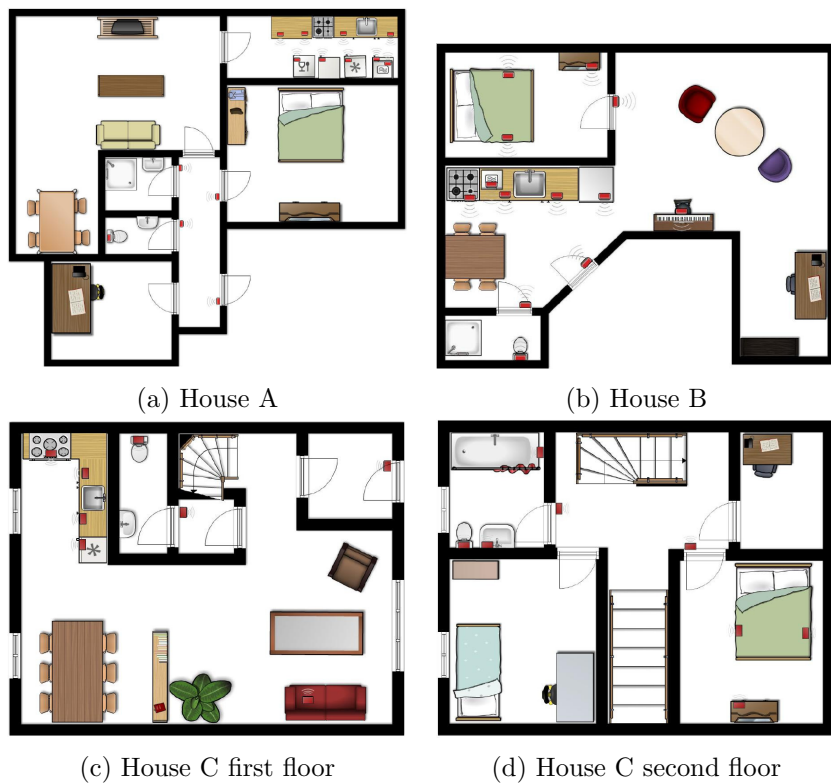


Figure 1: Home floor plans and sensor locations.

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

Table 1: Dataset recording details

## 5.2 Comparative Analysis Metrics

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

$$F - Measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (12)$$

## 5.3 Discussion

# 6 Conclusions and Lessons Learned

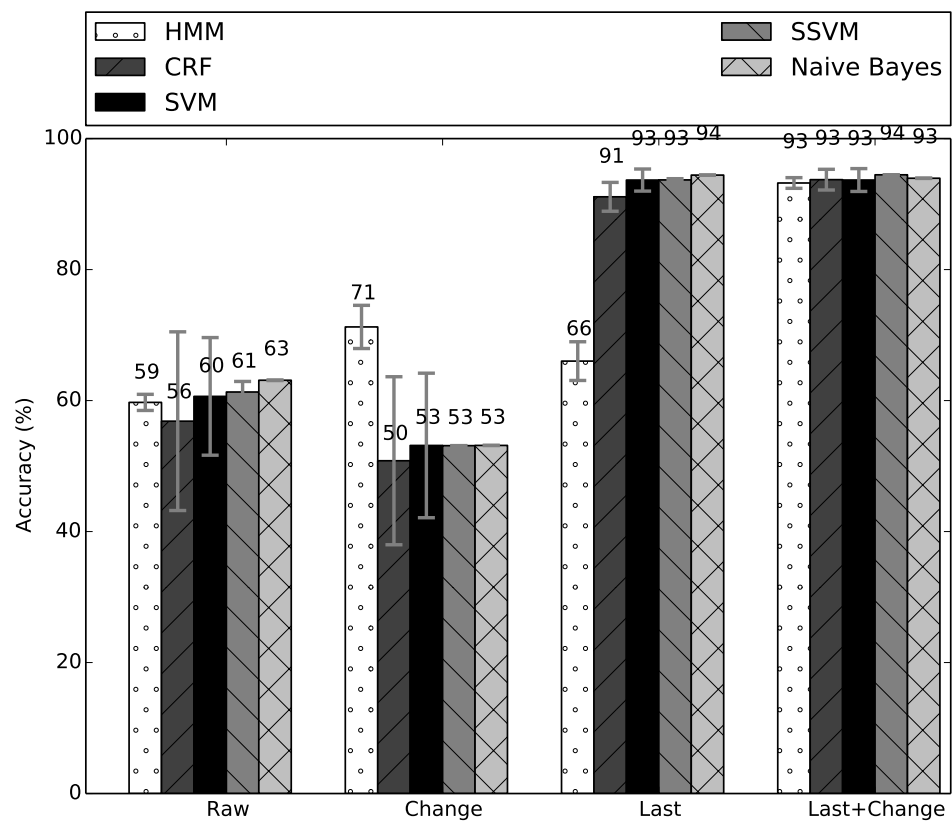


Figure 2: House A



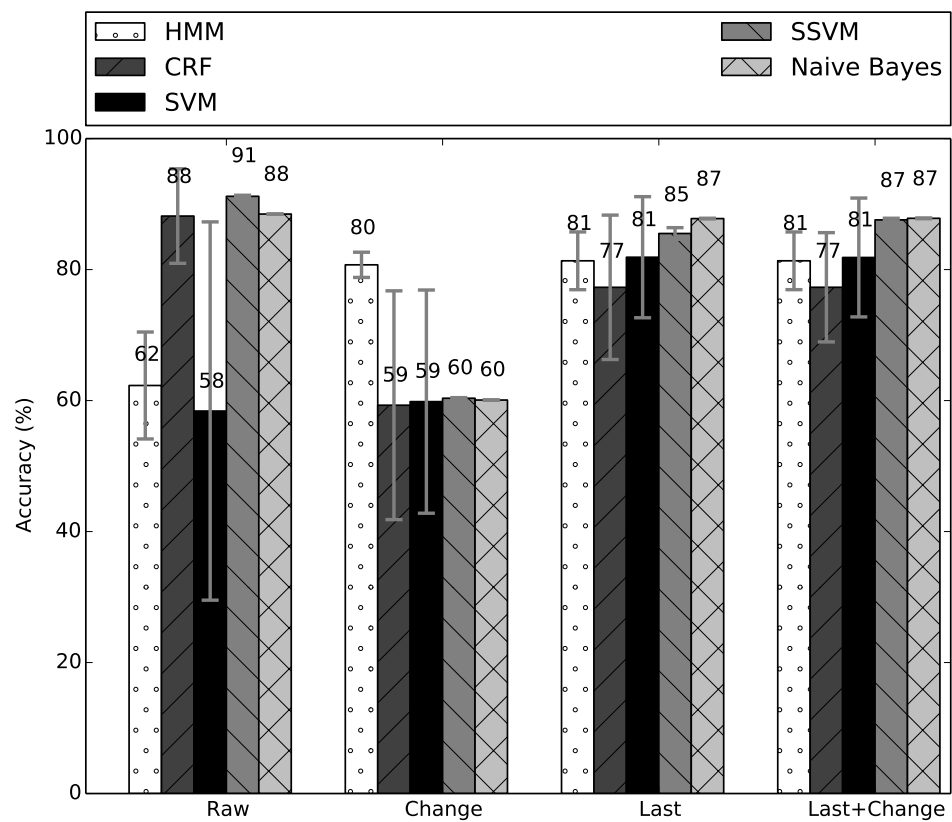


Figure 3: House B

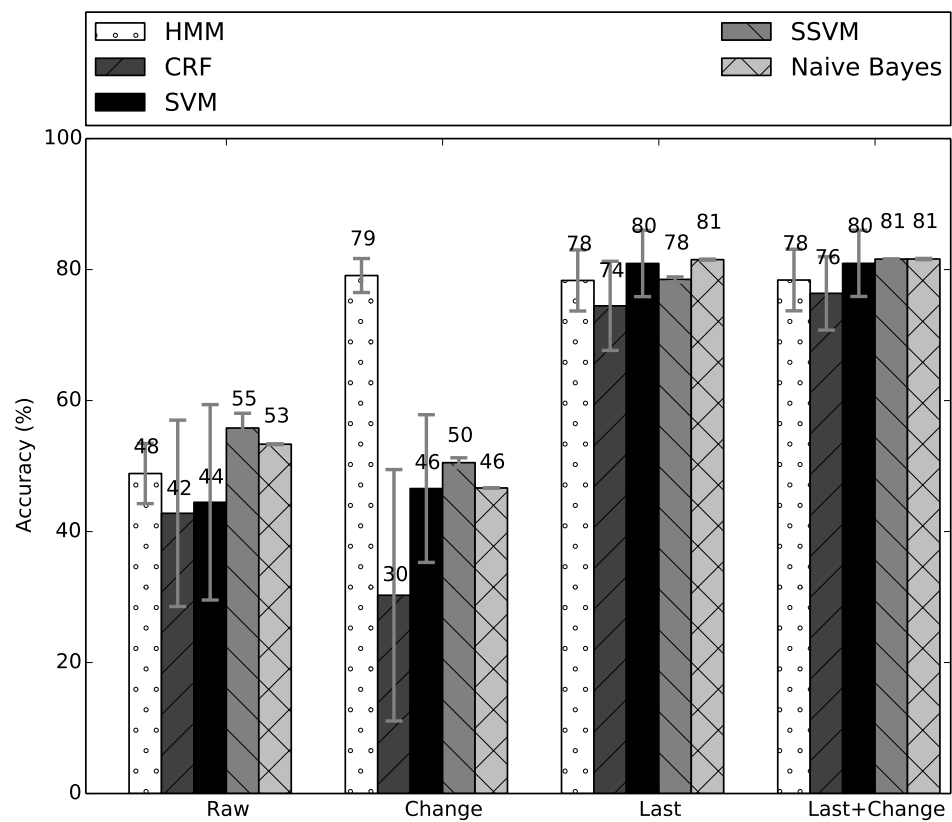


Figure 4: House C