Human Activity Recognition using Smartphones: an Empirical Evaluation

Author: Dan Wang Student ID: 20513942 Date: April 2016

Abstract

Here we are interested in human activity recognition (HAR) in a smart home system. Such recognition can help the elders and patients with daily tasks, detect abnormality such as falling off the bed and alert their caregivers. Smartphones, as wearable and non-intrusive devices, are used to collect real-time activity data for recognition. With such sensor input, our goal is to classify ongoing action of a person. In this empirical evaluation, we experiment with three machine learning techniques including K-nearest-neighbors (KNN), Support Vector Machine (SVM) and Hidden Markov Model (HMM). We quantitatively compare the three method on a dataset involving six kinds of common actions.

1 Introduction

Human activity recognition (HAR) is to classify a sequence of actions of a person based on sensors' measurements. It has attracted lots of attention for its usefulness in healthcare, security, entertainment and many more fields [9]. In a smart home system, human activities are recognized for context learning and predicting so as to control home devices intelligently, which is believed to reduce the cost of maintaining the environment. With the help of HAR, intelligent agent can be developed to help the elders in their daily tasks [6]. HAR is also a key component of an early warning system, which can identify abnormal / emerging activities, so caregiver can deal with such abnormalities in advance to avoid extremely bad consequences [16].

According to [3], an activity recognition system works by following an activity recognition chain (ARC). It is a sequential process including raw data collection, data processing, data segmentation, feature extraction and selection, training and classification, decision fusion and performance evaluation. However, as the chain itself indicates, each step of the ARC correlates and constraints each other. For example, we need to choose data processing method depending on the sensors used for data collection. Also what machine learning method to use for training and classification usually depends on raw data format, how the data are processed and feature extraction methods.

Due to the rapid advances in smart technologies, many kinds of sensors have been applied in the raw data collection procedure in a smart-home, e.g., special sensors, devices and appliances, wireless networking. When gathering data, wearable devices like smartphones have major advantages in terms of privacy and pervasiveness, compared to sensors like camera [13]. Besides, using accelerometers in smartphones do not need users to wear additional sensor components, so such system should be more practical and compact. Furthermore, smartphone is easy to access, has low-power consumption and reasonable price. Considering these, this project chooses data collected from smartphones.

The motivation for this topic here is as follows. Firstly, the emergence of companies with initiatives to bring smart caring technologies [4] to the elders is interesting. Secondly, this project aims to obtain an insightful understanding on the HAR problem and the selected machine-learning techniques. Here we do empirical evaluation on a public HAR dataset [1] by implementing and comparing several machine learning techniques. The dataset includes six selected activities: *standing*, *sitting*, *laying down*, *walking*, *walking downstairs* and *upstairs*. We choose this dataset because it is publicly available and has already processed and extracted features. Due to time limit, In this project, we will focus on the training and classification procedure in ARC pipeline.

2 Techniques to tackle the problem

2.1 Related works

In the aspect of training and classification, quite a few researchers have done substantial research on HAR. Some survey papers [9, 12] comprehensively summarize existing works in this field. Generally speaking, supervised machine learning techniques used in activity recognition can be broadly divided into template matching or transductive techniques, generative and discriminative approaches [8]. One of the well known template matching is the KNN classifier [10]. Many techniques for distance calculating, such as dynamic time warping (DTW) [11], are developed for using KNN. The Naive bayes classifier [15] is a generative approach and performs quite well when the amount of training data is large. Generative probabilistic graphical models, such as the hidden Markov models (HMM) [7], and discriminative probabilistic graphical models like conditional random fields (CRF) are believed to be good at sequencing data learning. The support vector machine (SVM) has advantages in computational burden and performs good in activity recognition [5]. In the remaining of this section, we summarize some of those interesting works and learn valuable techniques from them.

Zheng et al. [16] proposed a daily tasks clustering algorithm called Growing Self-Organizing Maps (GSOM), which is a self-adative neural network (SANN). The network was initialized with four neurons on a 2-dimensional grid. During the training process, it determined the winning neuron for each training data point and then adapts the weight vector for both the winning neuron and its neighbors. The method is user-friendly since it gathers information from sensors on doors drawers, refrigerators and so on, which can efficiently avoid the invasive and threatening recording caused by cameras. Besides recognizing activity pattern, the algorithm is claimed to be able to identify errors and abnormalities. However, the algorithm parameters, including the learning rate and the number of initial neighbors are achieved by a try-and-error approach.

Uddin et al. [13] combined the guided random forest and the extremely randomized trees for simple and complex human activity recognition. The proposed method is believed to give accurate results, fast computational performance, and minimal parameter settings. Random forest is a supervised ensemble learner, which can learn the impact of each feature towards predicting classes. The idea of guide is a heuristic where the Gini information gain of a feature is weighted by the normalized importance scores derived from normal random forest. This heuristic is mainly designed for reducing the computational burden of feature selection of high dimensional problems. In the extremely randomized trees, each decision tree are built by randomly selecting input variable and split values, resulting an output-independent decision tree. Trees in the forest are aggregated to yield the final output. The authors also claimed that hyper parameters, including the number of trees, number of attributes, and minimum sample size, do influence the performance greatly.

Kim et al. [7] adopted Hidden Markov model ensemble for activity recognition using the UCI Human Activity Recognition dataset, which is collected smart-phone accelerometers. They adopt the HMM because it is suitable for sequential dataset learning and has the capability of spotting sequence. HMM trains the classifier class-independently using a maximum likelihood criterion. This property leads to relatively unsatisfying performance of HMM on dealing with intra-class diversities and inter-class similarities, eg. walking upstairs and walking downstairs. In order to solve this problem, they uses the decision template (DT) based method to integrate the probabilities of multiple HMMs with respect to an observation sequence. This method obtains an accuracy of 83.51% with only two features.

Frank et al. [5] developed a working activity and gait recognition application, which can be installed into a commercial smartphone. The application not only has the ability of recognizing pre-programmed activities (running, jogging, jumping, walking), but also learns the style and gait of a person's movement. Besides, their application does not require special placement of phones on participant's body. In order to realize real-time learning and classification, a computational efficient machine learning algorithm, geometric template matching (GTM), is adopted in their system. In the implementation, the authors also found that GTM is quite robust with respect to the parameters.

2.2 Chosen techniques

In this project, three supervised machine learning algorithms are chosen and implemented, including K-nearest neighbours (KNN), Support Vector Machines (SVM) and Hidden Markov Models (HMM). All these three algorithms are popular techniques for supervised human activity recognition. By implementing these algorithms, a relative deep understanding of how their corresponding parameters influence their performance is established.

Here we give some basic notations. Let $x_n \in \mathcal{R}^d$ be a d-dimensional feature for n^{th} sample. The goal is to predict its label $y_n \in \mathcal{L}$ where $\mathcal{L}$ is the candidate label set. We are given N data points $X = \{x_1, ..., x_N\}$ and their labels $Y = \{y_1, ..., y_N\}$ as training set. Also note that each sample data x_n comes with the time t the data was measured. In other words, the input is a timed sequence of measurements.

KNN [2] is selected as a baseline method because of its simplicity and that it is easy to implement. The hyper parameter K may results in over-fitting while too small ,or under-fitting if too large. Appropriate K depends on the target problem and the number of training data. We experiment with K to yield the best testing error for the UCI human activity recognition dataset.

SVM [14] is selected because of its efficiency and wide applicability. SVM is a well-known algorithm and has been widely applied to many classification and regression fields and performs relatively well. As a kernel based algorithms, SVM reduced the computational complexity by adopting sparse representation, the support vectors. It is also robust to overfitting by maximize the margin, which makes SVM robust to noisy data. The very basic SVM algorithm is designed for 2-class classification. It is interesting to realize multi-class SVMs using lecture-mentioned methods, such as one-against-all, pairwise comparison, and continuous ranking.

HMM [2] is suitable for modeling sequential data such as speech and actions. Such sequential order is neither considered in KNN nor SVM. Usually a person is likely to repeat an activity for a while and switches to another action. For example, the selected dataset shows that a specific activity label repeat many times and switch to another one. This property results in the collected data points (series of time points / time window) correlating with each other. Thus strictly speaking, the training and testing data instances are not independent. HMM can sufficient consider such correlations by representing the state transition process in a simple Markov Chain. By implementing this algorithm we can see whether the correlation between data instances indeed influence the classification accuracy.

3 Evaluation

3.1 Algorithm details

3.1.1 K-nearest neighbors

For classification, KNN only needs a test procedure and predefined K. The classification of each testing data includes two steps.

Step1: calculate the K nearest neighbors $knn(x)$ of the test instance x within the training data X according to some distance measures.

Step2: label instance x using the most frequency label in the K nearest neighbors $knn(x)$.

In this project, we use Euclidean distance for measuring neighbors. To investigate how K value affects the performance, we try various K from 5 to 150. This is our baseline method.

3.1.2 Support Vector Machine

The typical SVM is used for two-class classification, i.e., $y_n \in \{-1, 1\}$. It works by finding a maximum margin between two dataset. The maximum margin is usually thought as a linear separator that maximizes the distance between the closest data points of the two classes, see (1), where $w^T\phi(x)$ is the linear separator for embedding $\phi(x_n)$.

$$\max_w \frac{1}{\|w\|} \min_n y_n w^T \phi(x_n). \tag{1}$$

This problem can be transformed into a convex quadratic optimization problem by fixing the minimal distance as 1, and then minimize $\|w\|$, see (2). Data instances who satisfy $y_n w^T \phi(x_n) = 1$ are active constrains for the optimization problem and are called the *support vectors*.

$$\min_w \frac{1}{2}\|w\|^2, \qquad s.t.\ \ y_n w^T \phi(x_n) \geq 1, \ \ \forall n. \tag{2}$$

The constrained optimization problem can be transformed into an unconstrained optimization problem by adding penalty $a_n \geq 0$ to instances who violate the constraint, see (3). As we can predict, a_n would be bigger than 0 only for the support vectors.

$$\max_{a \geq 0} \min_w L(w, a) \coloneqq \frac{1}{2}\|w\|^2 - \sum_n a_n [y_n w^T \phi(x_n) - 1]. \tag{3}$$

By setting the derivative of $L(w, a)$ with respect to w to 0, we get $w = \sum_n a_n y_n \phi(x_n)$. Substitute w, (4) is obtained where $k(x_n, x_{n'}) = \phi(x_n)^T \phi(x_{n'})$. Due to sparsity of support vectors, finally the problem could be transformed into a sparse optimization problem in $\mathbf{a}$,where many of a_n are zero since the number of support vectors are limited.

$$L(\mathbf{a}) = \sum_n a_n - \frac{1}{2} \sum_n \sum_{n'} a_n a_{n'} y_n y_{n'} k(x_n, x_{n'}) \tag{4}$$

By solving the sparse optimization problem in (5), we can get the optimal vector $\mathbf{a} = [a_n]$.

$$\max_{\mathbf{a}} L(\mathbf{a}), \quad s.t. \sum_n a_n y_n = 0 \quad and \quad a_n \geq 0. \tag{5}$$

When the classifier is trained, the testing for new instance x is achieved by:

$$y = sign \sum_n a_n y_n k(x_n, x) \tag{6}$$

However, (6) only predicts binary label. For multi-class classification to $C \geq 2$ classes, we train C binary one-vs-all classifiers, each for a class of positive and negative samples. For a new instance, we can obtain C classification scores from the C binary classifiers. The final label is the one that corresponds to maximum score (confidence).

3.1.3 Hidden Markov Models

So far, the data instances are classified independently. As mentioned earlier, each sample data x_n is measured sequentially in consecutive time steps. With slightly abuse of notation, we denote x_t the sampled data at time step t. We assume the following Markovian property:

$$p(y_{t+1}|y_t, y_{t-1}, ..., y_1) = p(y_{t+1}|y_t), \ \ \forall\, t. \tag{7}$$

The joint distribution of observations $x_1, ..., x_T$ and hidden variables $y_1, ..., y_T$ is given by:

$$p(y_1, ..., y_T, x_1, ..., x_T) = p(y_1) \prod_{t=2}^{T} p(y_t|y_{t-1}) \prod_{t=1}^{T} p(x_t|y_t). \tag{8}$$

We seek the most likely explanation for a sequential sequence given observations.

$$\arg \max_{y_1 \cdots y_T} p(y_1, ..., y_T|x_1, ..., x_T). \tag{9}$$

The inference for (9) is via dynamic programming and its global optimal can be achieved.

$$\max_{y_1 \cdots y_t} p(y_1 ... y_{t+1}|x_1 ... x_t) \propto \max_{y_t} p(y_{t+1}|y_t) p(x_t|y_t) \max_{y_1 \cdots y_{t-1}} p(y_1 ... y_t|x_1 ... x_{t-1}). \tag{10}$$

The remaining issue is how to learn initial state distribution $p(y_1)$, transition probability $p(y_{t+1}|y_t)$ and emission probability $p(x_t|y_t)$. We estimate the parameters (histograms) $p(y_1)$ and $p(y_{t+1}|y_t)$ by maximum likelihood estimation. Worth mentioning is that in this case $p(y_{t+1}|y_t)$ can be obtained by the frequency os state transition in the training sequences. For emission probability $p(x_t|y_t)$, we tried both parametric Gaussian distribution and non-parametric KNN density estimation. Since we deal with high-dimensional feature (561 dim.), parametric Gaussian or Mixture-of-Gaussians tends to overfit the data and is not reliable. We found KNN density estimation to work well as emission probability $p(x_t|y_t)$.

3.2 Dataset, Results and discussion

We experiment on UCI HAR dataset [1]. There are six activities in the dataset, namely walking, walking upstairs, walking downstairs, sitting, standing and laying. The activities are performed by 30 volunteers whose ages range from 19 to 48 years old. Each participant wore a smartphone and carried out a sequence of actions. Each sequence may have multiple kinds of actions involved. Sequence is sampled with certain time-step and at each time-step, a 561-dim feature vector is recorded. These sensor signals including accelerometer and gyroscope were already pre-processed by applying some filters. In total, sequences of 21 people are treated as the training set and that of the remaining 9 people are used as our testing set.

We count the number of misclassification, divided by the total number of records as the error rate measure. Note that there are multiple records in a single sequence. Fig. 1 shows performance of our KNN classifier with varying value of K.

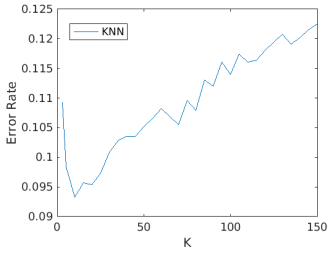

Figure 1: Error rate of KNN classifier for human activity recognition.

KNN achieved best error rate of **9.33%** when using an appropriate K. For SVM, we used a polynomial kernel and played with the order of the polynomials. We found that using order two polynomials worked well and gives an error rate of **3.65%**.

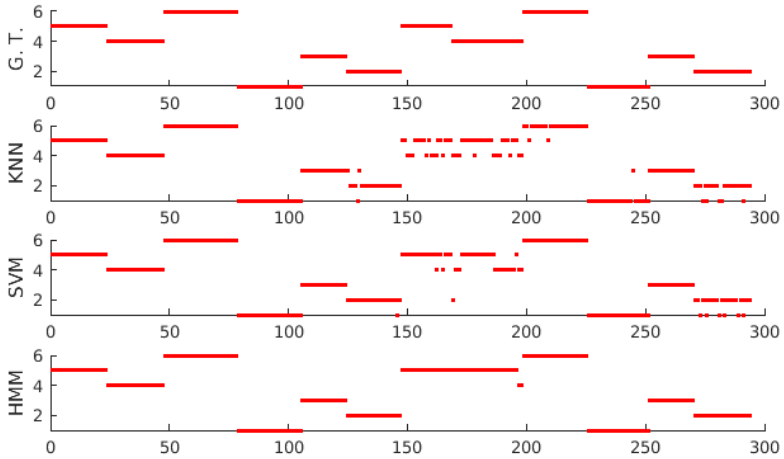

Figure 2: Comparing results of different classifiers on a single sequence. The x-axis denotes time frames. The y-axis denotes predicted classes. From top to bottom: ground truth labels, results of KNN, SVM and HMM.

Our best result is achieved by HMM, which gives error rate of **3.15%**. To see to what extend the Markovian assumption helps our algorithm, we deliberately used uniform translation probability and got much worse result of **9.50%**. In that case we only assign label to achieve the maximum emission probability $p(x_t|y_t)$, which is based on non-parametric KNN density estimation.

Fig. 2 plots the predicted labels and the ground truth label for a sequence of measurements. Our HMM based algorithm gives the most consistent prediction among consecutive time frames and also the best error rate. This confirms that HMM is appropriate for sequential data and enforces structured output.

We also tried a version of HMM with parametric multi-variate Gaussian as the emission probability $p(x_t|y_t)$. This version works pretty undesirable since a single Gaussian or mixture of Gaussians (GMM) overfits the high-dimensional data and didn't generalize well to new data as density estimation. We found non-parametric density estimation to be better.

4 Conclusion

In this project we use smartphones to collect measurements for human activity recognition. Three classifiers including KNN, SVM and HMM are implemented and we found HMM to be the best working algorithm. We manage to achieve recognition accuracy of about **97%** with such HMM method. In the future with much larger dataset we'd like to use recurrent neural network for such sequential data.

Reference

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *ESANN*, 2013.

[2] Christopher M Bishop. Pattern recognition. *Machine Learning*, 2006.

[3] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):33, 2014.

[4] Diane J Cook and Sajal K Das. How smart are our environments? an updated look at the state of the art. *Pervasive and mobile computing*, 3(2):53–73, 2007.

[5] Jordan Frank, Shie Mannor, and Doina Precup. Activity recognition with mobile phones. In *Machine Learning and Knowledge Discovery in Databases*, pages 630–633. Springer, 2011.

[6] Jesse Hoey, Pascal Poupart, Axel von Bertoldi, Tammy Craig, Craig Boutilier, and Alex Mihailidis. Automated handwashing assistance for persons with dementia using video and a partially observable markov decision process. *Computer Vision and Image Understanding*, 114(5):503–519, 2010.

[7] Yong-Joong Kim, Bong-Nam Kang, and Daijin Kim. Hidden markov model ensemble for activity recognition using tri-axis accelerometer. In *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pages 3036–3041. IEEE, 2015.

[8] Narayanan C Krishnan and Diane J Cook. Activity recognition on streaming sensor data. *Pervasive and mobile computing*, 10:138–154, 2014.

[9] Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *Communications Surveys & Tutorials, IEEE*, 15(3):1192–1209, 2013.

[10] Uwe Maurer, Asim Smailagic, Daniel P Siewiorek, and Michael Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, pages 4–pp. IEEE, 2006.

[11] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[12] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga. A survey of online activity recognition using mobile phones. *Sensors*, 15(1):2059–2085, 2015.

[13] Md Taufeeq Uddin and Md Azher Uddiny. Human activity recognition from wearable sensors using extremely randomized trees. In *Electrical Engineering and Information Communication Technology (ICEEICT), 2015 International Conference on*, pages 1–6. IEEE, 2015.

[14] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[15] Harry Zhang. The optimality of naive bayes. *AA*, 1(2):3, 2004.

[16] Huiru Zheng, Haiying Wang, and Norman Black. Human activity detection in smart home environment with self-adaptive neural networks. In *Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on*, pages 1505–1510. IEEE, 2008.