World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:15, No:11, 2021

Smartphone-Based Human Activity Recognition by Machine Learning Methods

Yanting Cao, Kazumitsu Nawata

***Abstract*—**As smartphones are continually upgrading, their software and hardware are getting smarter, so the smartphone-based human activity recognition will be described more refined, complex and detailed. In this context, we analyzed a set of experimental data, obtained by observing and measuring 30 volunteers with six activities of daily living (ADL). Due to the large sample size, especially a 561-feature vector with time and frequency domain variables, cleaning these intractable features and training a proper model become extremely challenging. After a series of feature selection and parameters adjustments, a well-performed SVM classifier has been trained.

***Keywords*—**Smart sensors, human activity recognition, artificial intelligence, SVM.

I. Introduction

HAR, short for Human Activity Recognition, is a broad research area involving the recognition of specific movements or actions of people based on sensor data.

These movements are usually specific activities performed indoors, such as walking, talking, standing and sitting. They may also be more specific activities, such as certain types of activities performed on a factory workshop or in a kitchen. Sensor data can be recorded remotely, such as video, radar or other wireless methods [1]. Alternatively, data can be recorded directly on a portable device, for example, by carrying customized hardware or a smartphone with the accelerometer and gyroscope.

Historically, sensor data for activity recognition has been challenging and time-consuming to collect, and may require customized hardware. The widespread use of smartphones for fitness and health monitoring, as well as other personal tracking devices, now makes sensor data from these devices more readily available, so research efforts targeting human activity recognition have been conducted primarily on data from these hardware phones.

The problem is to predict the class of activities based on sensor data, usually one-dimensional or multi-dimensional. In general, this problem can be modeled as a uni-variate or multi-variate time series classification task. This is indeed a challenging task because there is no obvious or direct way to correlate recorded sensor data with specific human activities, and the same activity can be performed significantly differently by different subjects, resulting in significant differences in the recorded sensor data. The aim is to record the sensor data and corresponding activity of a specific subject, fit a model based on this data, and then generalize the model to classify the activity of the sensor data of a new unseen subject.

In this paper, the experiments have been carried out in a group of 30 people with six activities measured by wearing the smartphone on the waist. The obtained database has been randomly partitioned into two sets, where 5080 data of 15 people have been used for training purposes and 5219 data of the other 15 people as test data.

The main objective is by analyzing the training data with 560 features to classify the activities and improve the prediction accuracy. After looking through these features and relative description, feature selection is used for filtering the useless components and these patterns are used as input of the trained SVM Classifier for the recognition of the activities. Finally, a great result 0.906 comes out on the validation set which is much better than the baseline of SVM 0.868.

II. Data Description and Experimental Setup

The HAR dataset was created through a series of trials [2]. Thirty volunteers aged from 19 to 48 years were selected to take part in this experiment. Each participant was asked to wear a Samsung Galaxy S II smartphone on their wrist and follow the process of activities (Table I). The six required activities of daily living (ADL for short) are: standing, sitting, laying down, walking, walking downstairs and walking upstairs [3].

TABLE I
Process of Activities for HAR Experiment

Number	*Static*	*Time*
0	start	0
1	stand (*)	15
2	sit (**)	15
3	stand (*)	15
4	lay down (*)	15
5	sit (**)	15
6	lay down (**)	15
	Dynamic	
7	Walk (*)	15
8	Walk (**)	15
9	walk downstairs (*)	12
10	walk upstairs (*)	12
11	walk downstairs (**)	12
12	walk upstairs (**)	12
13	walk downstairs (***)	12
14	walk upstairs (***)	12
15	stop	0
	Total	***192***

The process was repeated two times for each subject. The

Yanting Cao and Kazumitsu Nawata are with the Graduate School of Engineering, The University of Tokyo, Japan (e-mails: yanting2016cc@hotmail.com, nawata@tmi.t.u-tokyo.ac.jp).

first time they were asked to fix the smartphone on the left side, and the second time they could place the smartphone any side they wanted. This experiment was conducted in laboratory conditions, but the participants could move freely in order to obtain more naturalistic data.

All the data have been measured by wearing the smartphone on the wrist; then, the obtained database has been randomly partitioned into two sets, where 5080 data of 15 people have been used for training purposes and 5219 data of the other 15 people as test data in the following section. In addition, a vector of 17 features is obtained by calculating variables from the accelerometer signals in the time and frequency domain such as mean, standard deviation, signal magnitude area, entropy, signal-pair correlation, etc.

III. Feature Mapping and Feature Selection

Generally, some standard measures such as mean value, standard deviation, median absolute value, maximum/minimum, signal magnitude area and average sum of the squares would be adopted for feature mapping in HAR experiments [4]. Besides these, more fresh functions used for promoting learning performance: interquartile range, signal entropy, autorregresion coefficients, correlation coefficient, largest frequency component, frequency signal weighted average, frequency signal skewness, frequency signal kurtosis, energy of a frequency interval, angle between two vectors. [5]

After doing the entropy calculations, especially computing their mutual information, a disturbing number of features are not correlative. If cutting off half features then training the model, the precision increased and the value on the validation set became higher. For these reasons, doing a feature selection is a helpful pre-processing method. It also means that there must be quite a lot of redundancy so that the feature selection is necessary and the filters or wrappers have been tried in the feature selection phase.

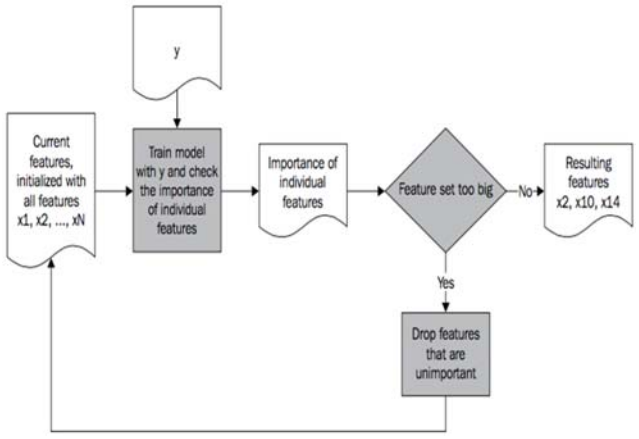

Fig. 1 Process chart of feature selection

Filters have helped immensely to get rid of useless features, but that is about all they can do. After all the filtering, there might still be some features that are independent among themselves and show some degree of dependence with the result variable, but yet they are totally useless from the model's point of view. Therefore, what the wrappers do, as shown in the process chart (Fig. 1), is to ask the model itself to vote on individual features.

In this case, the calculation of feature significance would be shifted to the model training process. Regrettably, but intelligibly, feature significance is determined by ranking value, not by binary. So, we still have to specify where to make the cut, what part of the features we are willing to take, and what part to drop.

RFE, short for recursive feature elimination, would be picked to solve this intractable part. It takes an estimator and the desired number of features to keep as parameters and then trains the estimator with various feature sets as long as it has found a subset of features that is small enough [6]. The RFE instance itself pretends to be like an estimator, thereby, indeed, wrapping the provided estimator.

Given an external estimator that assigns weights to features，for instance, coefficients of a simple unary linear model, RFE is the process of selecting features by recursively considering smaller and smaller sets of features. A more common way to handle it is as follows:

1. train the estimator on the initial feature set and gain the importance of each feature through the “coef_attribute” or “feature_importances_attribute”.
2. remove the least important features from the current feature set. Then repeate this process recursively on the pruned set until the desired number of features is finally reached.

In a nutshell, the main idea of RFE is to iteratively build multiple models repeating the above two steps many times. The order in which the features are eliminated in this process is the ranking of the features. Therefore, this is a greedy algorithm for finding the optimal subset of features and the effectiveness of RFE depends on the model chosen.

For the RFE function, eight main parameters are used in this research:

- estimator: a supervised learning estimator. Its fit method provides information about the importance of the attributes by “coef_attribute” or “feature_importances_ attribute”.
- step: default is 1. An integer indicates the number of features to be eliminated at a time. Less than 1 means that the feature with the lowest weight is removed each time.
- Verbose: default is 0 to control the output verbosity [7].
- n_jobs: control the number of CPU cores utilized in parallel operations. The default is 1, i.e., single-core operation. If set as -1, all cores are enabled for the operation.
- n_features_: the final number of features left through the cross-validation process. Default is half retained.
- support_: the selected status of the selected features. True means selected and False means eliminated.
- ranking_: the ranking of all features in terms of their scores.
- estimator_: the model trained with the remaining features.

By trying cutoff many times, it would be found that the result of 190 features is very stable because features that have been used when requesting smaller feature sets keep on getting selected when letting more features in.

IV. Methodology and Experimental Results

The raw data was downloaded from UCI Machine Learning Repository [8] and all the datasets were implemented by Python software.

Machine Learning methods have been employed which are Logistic Regression (LR), k-nearest neighbors (KNN) and Support Vector Machine (SVM).

A. Logistic Regression (LR) Method

The LR model is represented by the conditional probability P(Y|X) and assumes that this distribution is the logistic distribution. LR learning model is to determine the unknown parameters in this distribution, and the unknown parameter part is denoted by wx+b.

The binomial logistic regression model is with the following conditional probability distribution when:

$$P(Y=0|X)=\frac{1}{1+e^{wx+b}}$$

$$P(Y=1|X)=\frac{1}{1+e^{-(wx+b)}}=\frac{e^{wx+b}}{e^{wx+b}}$$

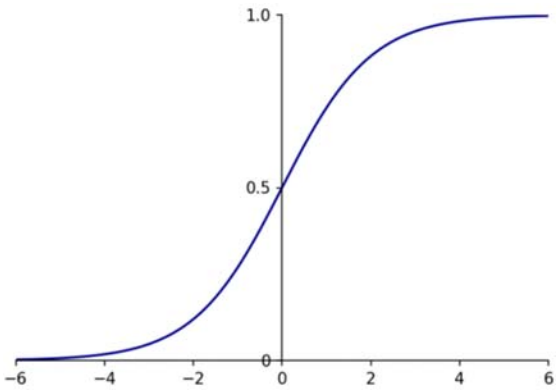

Fig. 2 Sigmoid function curve

Another perspective, the range of wx is the domain of real numbers R. Using the sigmoid function can transform the value of wx into probability because the value domain of sigmoid is also [0,1]. And sigmoid is a monotonically increasing function. The closer the value of the linear function is to positive infinity, the closer the probability value is to 1; the closer the value of the linear function is to negative infinity, the closer the probability value is to 0. The sigmoid function curve is shown in Fig. 2.

B. K-Nearest Neighbors Method

KNN algorithm is a classification algorithm proposed by Cover and Hart in 1967 [9] for applications such as character recognition, text classification and image recognition. The idea of the algorithm is that a sample is most similar to k samples in a data set, and if most of these k samples belong to a certain class, then the sample also belongs to that class.

The key point of the KNN method is how to choose the k-value. If a smaller value of k is chosen, it is equivalent to predicting with training instances in a smaller neighborhood, and the approximation error of learning will be reduced. Only the training instances that are closer to the input instances will work for the prediction results. However, the disadvantage is that the estimation error of learning will increase and the prediction results will be sensitive to the split of instance points in the near neighborhood. If the neighboring instance points happen to be noisy, the prediction will be wrong.

In other words, a decrease in the value of K means that the overall model becomes more complex and less clearly divided, and it is prone to overfitting.

C. SVM Method

Support vector machines is a binary classification model. Its basic model is a linear classifier with maximum interval defined on the feature space, and the maximum interval distinguishes it from other perceptron. SVM also includes kernel tricks, which make it a substantially nonlinear classifier.

The basic idea of SVM learning is to seek for a separating hyperplane that correctly partitions the training data set and has the largest geometric separation.

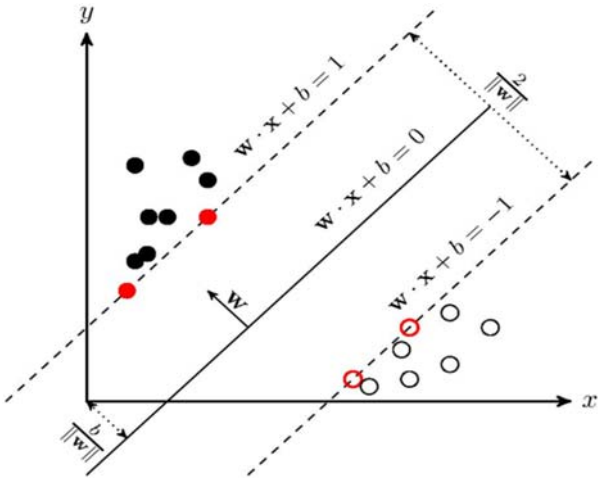

Fig. 3 Hyperplanes of SVM model

In Fig. 3, **w***x+b = **0** is the separating hyperplane. There are infinitely many such hyperplanes for linearly separable data sets, but the geometrically separated hyperplane with the largest interval is unique.

For a nonlinear classification problem like this experiment, it can be transformed into a linear classification problem in some dimensional feature space by a nonlinear transformation to learn a linear SVM in a high-dimensional feature space.

Since in the pairwise problem of linear SVM learning, both the objective function and the classification decision function involve only the inner product of instances and instances [10]. Therefore, instead of explicitly specifying the nonlinear transformation, the inner product is replaced by a kernel function.

D. Experimental Results

However, a hypothesis of "SVM would perform better for classification" came out since related work has showed that SVM had confirmed successful application and performed

better in activity recognition topics [11].

When trying to run with original code, the result was not satisfying even quite disappointing, so adjusting parameters became imperative. Firstly, setting the parameter c located in the real number interval (0,1) and tried as much as possible to compare the different result. Secondly, picking several suggested parameter gamma [12] to compare their difference. Finally, the best result was obtained with following parameters:

- c cost: set the parameter c of C-SVC, epsilon-SVR, and nu-SVR (default 1)
- g gamma: set gamma in kernel function (default 1/num_features)

It is worth mentioning that when selecting 300 features, the result of LR is 0.902 as well as the SVM result is 0.905, so LR could be abandoned in the early stage to save more time, and this proved to be a wise choice.

V. Conclusion and Recommendation

This study further examined the extent of validity of features in this dataset and compared the difference between classification models, including linear regression and SVM classification. Our study finds that among all the 560 features, the top 190 of which is sufficient to train regression models reaching a result of more than 0.90 precision. Our result indicating that the information of this dataset is redundant for a classification task predicting the movement of the persons, and there's a space to balance the trade-off between collecting and preprocessing as much as the raw data for training the classification model with full size, and collecting and preprocessing relatively less information from raw data and train same machine learning models with smaller but effective dataset. At last we got a precision of 0.906 from SVM classification, much better than the baseline of SVM 0.868 [13]. Through feature elimination, the result has shown that the feature selection is appropriate and helpful to solve such a classification problem. What's more for training techniques to complete the task is the parameter selection. It may be considered to further apply the recommendation for which is that if the higher precision required, it had better do some Grid Search (GS). Though GS seems quite naïve [14], its two advantages cannot be ignored: getting global optimum, and easy to optimize with c and gamma independent.

To compare the result of classification result between machine learning models, it found a difference between LR and SVM, 0.902 vs 0.905, were not very significant. Although the SVM model have much more space for further improvement by parameter selection like Grid search, there's no doubt that LR model is also workable for a simple classification task. In machine learning research area, the debate between using complex model for possible performance improvement and using simple model for lower data collection and training cost have been existed for a long time. Thus, these findings provided another evidence that this trade-off exist, simple model like LR can perform as good as complex model SVM without detailed refinement in such simple classification tasks, and researcher and developers may pick what their task need.

References

[1] Wang, J., Chen, Y., Hao, S., Peng, X., and Hu, L., "Deep learning for sensor-based activity recognition: A survey". Pattern Recognition Letters119, 2019, pp3-11.

[2] Anguita, D., Ghio, A., Oneto, L., Parra, X., and Reyes-Ortiz, J. L., "A public domain dataset for human activity recognition using smartphones". In Esann, 2013, Vol. 3, pp3.

[3] Anguita, D., Ghio, A., Oneto, L., Parra, X., and Reyes-Ortiz, J. L., "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine". In International workshop on ambient assisted living, Springer, Berlin, Heidelberg, 2012, pp. 216-223.

[4] Yang, J. Y., Wang, J. S., and Chen, Y. P., "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers". Pattern recognition letters 29(16), 2018, pp2213-2220.

[5] Khan, A. M., Lee, Y. K., Lee, S. Y., and Kim, T. S., "Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis". In 2010 5th international conference on future information technology. IEEE, 2010, pp1-6.

[6] Granitto, P. M., Furlanello, C., Biasioli, F., and Gasperi, F., "Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products". Chemometrics and intelligent laboratory systems, 2006, 83(2), pp83-90.

[7] Guyon, I., Weston, J., Barnhill, S., & Vapnik, V., "Gene selection for cancer classification using support vector machines", Mach. Learn.,2002, 46(1-3), pp.389–422.

[8] Frank, A., and Asuncion, A., "Human Activity Recognition Using Smartphones Data Set". UCI machine learning repository, 2010.

[9] Cover, T., and Hart, P., "Nearest neighbor pattern classification". IEEE transactions on information theory, 1967, 13(1), pp. 21-27.

[10] Zhou Z H., "Ensemble methods: foundations and algorithms[M]". Chapman and Hall/CRC, 2019.

[11] Cortes, C., and Vapnik, V., "Support-vector networks". Machine learning,1995, 20(3), pp273-297.

[12] Mantovani, R. G., Rossi, A. L., Vanschoren, J., Bischl, B., and Carvalho, A. C., "To tune or not to tune: recommending when to adjust SVM hyper-parameters via meta-learning". In 2015 International Joint Conference on Neural Networks (IJCNN), Ieee, 2015, pp1-8.

[13] Subasi, Abdulhamit, et al., "Smartphone-based human activity recognition using bagging and boosting". Procedia Computer Science 163, 2019, pp54-61.

[14] Hesterman, J. Y., Caucci, L., Kupinski, M. A., Barrett, H. H., and Furenlid, L. R., "Maximum-likelihood estimation with a contracting-grid search algorithm". IEEE transactions on nuclear science, 2010, 57(3), pp1077-1084.