Human Activity Recognition With Smartphones

Jeremy Sapienza, Simone Fiorellino, Stefano D'Arrigo

December 27, 2020

1 Introduction

Nowadays everyone has a smartphone in his pocket and carries it everywhere he goes during the day journey. Smartphones collect each time a huge amount of data through their embedded sensors. High tech industries are involved into the development of accurate wearable devices to improve every day life, monitoring the activities of their customers, from walking in a park to laying on the bed.
Hence, we reckoned this topic valuable and in our project we focused on the classification task of human activities, making use of accelerometer and gyroscope records. Our attention was particularly concentrated on the training of two classifiers, GDA and SVM, and on the research of a convenient data representation. In this report we summarize the methods we applied and the choices we made. Finally, in the last section we present our findings and give some further analyses.

2 Related work

For this work we were inspired by the paper [9]. Their research focused on developing a classifier that operates on accelerometer and gyroscope data from mobile phones. As explained in [1], they recorded the stream of data of 30 subjects during different types of activities: walking, walking up stairs, walking down stairs, sitting, standing and laying (Figure 1).
The dataset have two types of data:

- "Raw data": raw gyroscope and accelerometer readings.
- "Preprocessed data": vectors of 561 features that represent 2.56s of time. These features are the result of different functions, some examples are: triaxial average, maximum and minimum acceleration, angular velocity (over the given interval) or Fourier transform.

They used for this part of work the preprocessed data.
To approach this problem they used three different models:

- Naives Bayes
- GDA
- GDA + Hidden Markov Model

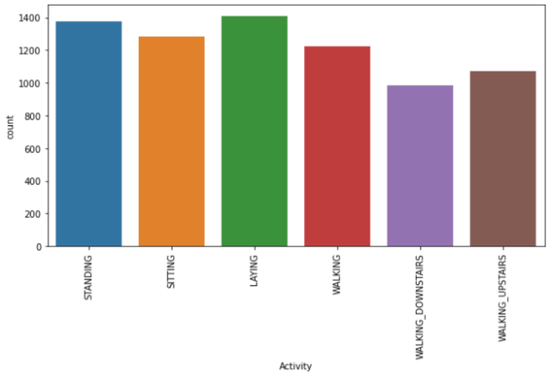


Figure 1: Number of samples of each class. As we can see, the dataset is almost balanced.

Algorithm	**Accuracy**
Naive Bayes	80%
GDA	96%
GDA + HMM	98%

Table 1: Results of the analysis of paper [9].

Reffering to Table 1, the GDA accuracy is much higher than the Naive Bayes model because this last model makes the assumption that the features are independent of each other. But, in this context, having the features more correlated together, the accuracy with the Naive Bayes model is pretty low. Another important aspect of this paper is the dimensionality reduction. They used the PCA in order to decrease the computational complexity of the model and improve the accuracy.

3 Methods

3.1 Data preprocessing

We retained 60% of the data for the models' training; the remaining part was equally splitted into evaluation and test sets.

3.2 Models

As a first step, we implemented the Gaussian Discriminant Analysis classifier and trained it on the data; the choice of this model was mainly conditioned by [9]. For the implementation, we faced out with the problem of multiclass classification, as in [7, 12]. With the purpose of improving the performance, we took into account a step of feature selection and dimensionality reduction:

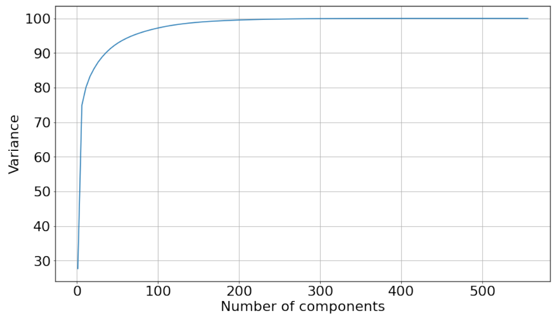


Figure 2: Feature variance with respect to the number of components after the application of the SVD method. The plot confirms that many features are not very discriminant: almost 100% of the variance is retained with just 200 components.

indeed, the high number of derived features forced us to focus on the relationship between the features of the original dataset, the variance of them and the predicted classes. So, we considered two different strategies:

- feature selection with the Analysis of Variance model (ANOVA);
- dimensionality reduction with the Singular Value Decomposition method (SVD) (Figure 2).

The second step was training the Support Vector Machine classifier on the data, following the same approach just described. After reviewing [5, 2, 8, 3] and having tried to implement it as in [11, 6], we decided to use the implementation given by the library "Scikit Learn" [10, 4]. In this case, it was essential finding a good configuration of feature space and model hyperparameters. Since this research would have taken too much to be conducted on each possible combination of hyperparameters, we chose a set of them which we reckoned reasonable. The results of the training and fine tuning step will be presented and commented on in the next section.

4 Experimental results

4.1 GDA results

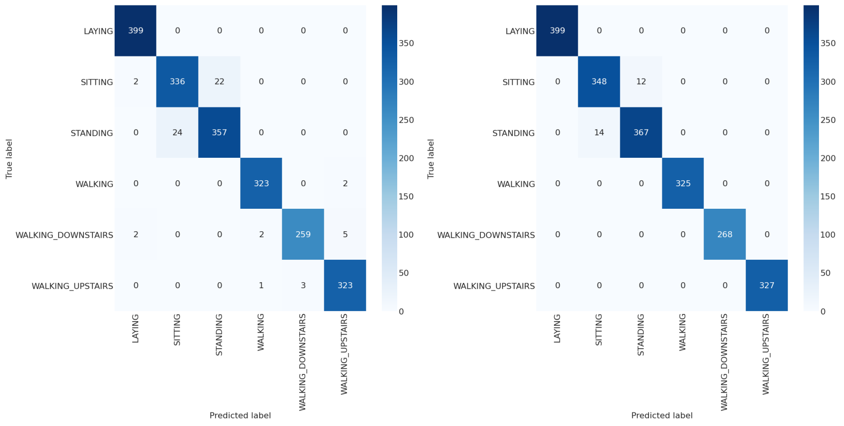
Initially the model led to an accuracy of 94% on the whole feature space. Instead, by applying the dimensionality reduction we improved the score, obtaining an accuracy of 97%. In this model, *sitting* and *standing* are the classes with lowest performance, as we can see in the confusion matrix (Figure 3a).

4.2 SVM results

As we can see in the confusion matrix (Figure 3b), SVM got a better performance than the GDA model, 99% of accuracy. The results we obtained are compliant to our expectations, indeed:

- The GDA carried out the same performance of the reference paper [9].

- Support Vector Machine led to the best results exceeding those of references.
- We improved the performance of our model through the dimensionality reduction and the choice of good hyperparameters.



(a) **(b)**

Figure 3: Confusion matrices of GDA (fig.3a) and SVM (fig.3b).

5 Conclusion and future work

The results we got were quite impressive, cause we managed to reach a very high preformance score. As a future work, it would be worth handling the raw data, in order to elaborate the data without preprocess them.

References

[1] A public domain dataset for human activity recognition using smartphones. In *21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, ESANN 2013., Bruges, Belgium, 24-26 April 2013.

[2] C. C. Aggarwal. *Data mining: the textbook*. Springer, 2015.

[3] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[4] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[5] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[6] A. Kowalczyk. *Support Vectors machines Succintly.* Syncfusion, 2017.

[7] G. Lemaitre. Multiclass classification strategies. https://github.com/scikit-learn/scikit-learn/blob/master/sklearn/multiclass.py, 2020.

[8] T. Ma, A. Ng, and C. Ré. CS 229, Autumn 2009 The Simplified SMO Algorithm. 2009.

[9] T. D. Matt Brown and L. O'Conor. Activity classification with smartphone data. 2013.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

[12] Tajudeen Kolawole. GDA classifier. https://github.com/bamtak/machine-learning-implemetation-python/blob/master/Multi%20Class%20Gaussian%20Discriminant%20Analyses.ipynb.