DetectAsana: Classification of Body Postures and Movements

Bhavey Wadhwa
2018135
IIIT Delhi
bhavey18135@iiitd.ac.in

Ritik Garg
2018305
IIIT Delhi
ritik18305@iiitd.ac.in

Rohit Makkar
2018087
IIIT Delhi
rohit18087@iiitd.ac.in

Abstract

With the increasing population and lesser life expectancy, the world demands healthcare systems for Ambient Assisted Living(AAL). Ambient Assisted Living(AAL) is an emerging field that aims at exploiting Information and Communication Technology(ICT) in healthcare systems to render personalized, adaptive, and anticipatory requirements. Human Activity Recognition(HAR) can be used to provide information about a person's daily routine to develop an e-health system to support the care of the elderly, chronically ill, and people with special needs. We aim to use unimodal wearable sensor data for the classification of body postures and movements. But the question remains 'Why' and 'Which' algorithm is the best for the particular dataset, as there is no universally accepted algorithm for the same. Here, we will try to answer these 'W's' by comparing performance of various algorithms for the "PUC-Rio" dataset.

1. Introduction

To compare performance of various classification algorithms to classify body postures and movements using the data collected from accelerometers mounted at different body parts. (PUC-Rio Dataset)

In today's world, jobs are becoming more of a sitting work, with increased screen-time. This leads to an obesity prone world. Our work in the field of Human Activity Recognition could be deployed in making products for a better routine. Furthermore, the products hence made can help people correct their body postures and reduce other problems like back pain, fix hunchback problems etc.

2. Literature Survey

The task of posture detection majorly has three sections:

Data Collection: The data is obtained on different subjects using devices placed on body parts like arms, legs, foot, waist, wrist etc. [2]. The devices are specially designed so as not to lose much information while providing the comfort of wearing it [3]. Apart from the raw features, the features such as pitch, norm, roll etc. are calculated from the raw features [2].

In [1], PAMAP2 dataset, which includes gyroscope and magnetometer readings are also present along with accelerometer's data, is used.

Preprocessing: The datasets used were already processed. Although, there were class imbalances that we have taken into account by using undersampling and oversampling the data and assigning the class weights to get better performance.

Performance: Various Machine Learning algorithms were used for classification including Naive Bayes, Decision Trees, and Random Forest [1] [2] [3]. Despite the modifications(sampling) in the datasets, tree based algorithms of Decision Trees and Random Forest ensemble tend to give the best performance on all the datasets.

3. Dataset and Preprocessing

Name: "PUC-Rio" [2]

The data was collected on eight hours activities of four Subjects. Four Accelerometers were positioned on four different body parts (waist, left-thigh, right-ankle, right-arm), each giving reading along 3 axis(x-axis, y-axis, z-axis). The Data with 165,633 samples is classified into five classes namely sitting-down, standing-up, sitting, standing, walking.

Attributes: Name, Gender, Age, Height, Weight, Body Mass Index (BMI), x_1, y_1, z_1 (accelerometer reading mounted on waits of x,y,z axis), x_2, y_2, z_2 (accelerometer reading mounted on left-thigh of x,y,z axis), x_3, y_3, z_3 (accelerometer reading mounted on right-ankle of x,y,z axis), x_4, y_4, z_4 (accelerometer reading mounted on right-arm of x,y,z axis), Class.

The dataset obtained was already preprocessed on the raw features The dataset consists of 11,827 samples of "Sitting Down", 12,414 samples of "Standing Up", 43,390 samples of "Walking", 47,370 samples of "Standing" and 50,361 samples of "Sitting" which is clearly not balanced which can be seen in fig1.

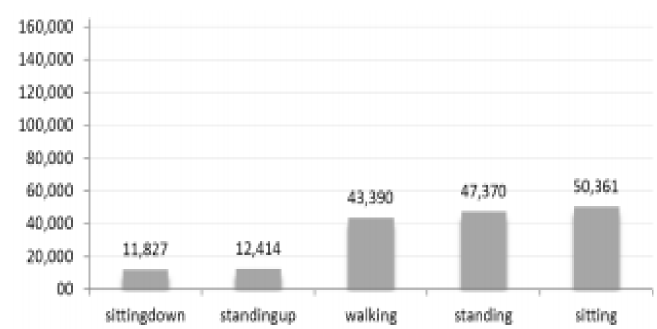

Figure 1. Class Imbalance, Source:[2]

4. Methodology

Feature Extraction: For all the 4 accelerometers, norm, pitch and roll were extracted as:

		Logistic Regression		SGD Classifier		Decision Trees		Random Forest	
	Class	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
Raw Data	**1**	0.992	0.976	0.995	0.966	0.998	0.999	1.000	1.000
	2	0.865	0.735	0.973	0.726	0.986	0.988	0.997	0.997
	3	0.722	0.806	0.682	0.969	0.981	0.982	0.998	0.996
	4	0.557	0.696	0.506	0.742	0.949	0.944	0.993	0.989
	5	0.604	0.744	0.632	0.691	0.932	0.929	0.979	0.992
	Macro	0.748	0.791	0.758	0.819	0.969	0.968	0.994	0.995
	Weighted	0.837	0.825	0.880	0.845	0.982	0.982	0.997	0.997
ADA-SYN (Oversampling)	**1**	0.984	0.941	0.989	0.953	0.998	0.998	1.000	1.000
	2	0.779	0.604	0.843	0.666	0.987	0.988	0.999	1.000
	3	0.612	0.664	0.596	0.893	0.985	0.979	1.000	0.998
	4	0.632	0.636	0.784	0.504	0.984	0.988	0.999	1.000
	5	0.465	0.621	0.317	0.668	0.983	0.983	0.999	0.999
	Macro	0.694	0.693	0.706	0.737	0.987	0.987	0.999	0.999
	Weighted	0.715	0.694	0.772	0.706	0.987	0.987	0.999	0.999
Class Weights	**1**	0.983	0.985	0.994	0.962	0.998	0.998	1.000	1.000
	2	0.820	0.770	0.878	0.804	0.981	0.987	0.998	0.997
	3	0.652	0.768	0.760	0.842	0.982	0.976	0.999	0.995
	4	0.687	0.581	0.270	0.795	0.954	0.951	0.992	0.988
	5	0.687	0.618	0.735	0.496	0.938	0.938	0.977	0.993
	Macro	0.766	0.744	0.727	0.780	0.971	0.970	0.993	0.995
	Weighted	0.810	0.807	0.856	0.828	0.982	0.982	0.997	0.997
Monte Carlo (Undersampling)	**1**	0.693	0.765	0.623	0.715	0.968	0.965	0.992	0.993
	2	0.707	0.760	0.658	0.744	0.976	0.972	0.996	0.994
	3	0.653	0.748	0.595	0.758	0.967	0.976	0.995	0.995
	4	0.821	0.651	0.753	0.656	0.980	0.977	0.995	0.995
	5	0.978	0.951	0.966	0.957	0.996	0.997	0.999	0.999
	Macro	0.770	0.775	0.719	0.766	0.977	0.977	0.995	0.995
	Weighted	0.780	0.770	0.790	0.719	0.977	0.977	0.995	0.995

Table 1. Precision and Recall for different approaches

- **Norm:** $\sqrt{x^2 + y^2 + z^2}$
- **Pitch:** atan(y/$\sqrt{x^2 + z^2}$)
- **Roll:** atan(-x/z)

Twelve more features(norm1, pitch1, roll1, ..., norm4, pitch 4, roll4) were extracted.

All Machine Learning Algorithms work best when there is an evenly distributed class. To deal with the class imbalance problem, we applied the following strategies including undersampling, oversampling and class weights.

- **SMOTE(Synthetic Minority Overlapping Technique):** It is an oversampling technique which ,instead of just duplicating data of minority class, uses KNN(K-Nearest Neighbour) to create synthetic data.
  - **ADASYN(Adaptive Synthetic Sampling):** It is a variation of SMOTE where data is synthesised based upon the original data density. The basic principle it works upon is that data is synthesised more where its density is lesser.
- **Class Weights:** The weights assigned to the class are inversely proportional to the number of data samples in the class. This is done to highly penalize the wrong prediction of the data samples of classes with less data.
- **Monte Carlo Sampling for Undersampling:** Here, all the classes were resampled and the number of samples in each class were equal to the minimum number of samples among all the classes. Let this minimum number of samples be 'n' (here, 11827).

The number of iterations is the number of times random samples of size n are chosen from each class

Algorithm:

- Let the number of iterations be 'm'.
- For each iteration:
  - For each class, randomly pick n data samples with replacement.
  - Run the classification algorithm.
  - Calculate precision and recall class-wise.
- Return the average results for each class.

We used the following classification algorithms to classify the body postures and movements using all the features.

- Logistic Regression
- SGD Classifier
- Decision Tree Classifier
- Random Forest

5. Results and Analysis:

The evaluation metrics for all sampling techniques against the machine learning algorithms are compiled in Table 1 and 2.

On analyzing results on the raw data, the precision and the recall in classes 4 and 5 were significantly less compared to the other

	Logistic Regression	SGD Classifier	Decision Tree	Random Forest
Raw Data	0.825	0.845	0.982	0.997
ADA-SYN	0.694	0.706	0.987	0.999
Class Weights	0.807	0.828	0.982	0.997
Monte Carlo	0.770	0.719	0.977	0.995

Table 2. Accuracy for different approaches

three classes because of the lesser number of training samples.
We oversampled the entire dataset and split the data into the train to test ratio of 70:30 using ADA-SYN, Random Forest Ensemble Classifier worked the best on both train and test data. The problem didn't seem to resolve for Logistic Regression and SGDClassifier. Accuracy for both these algorithms dropped significantly.
Further, we trained the data by assigning weights(inversely proportional to the number of samples in the class) to classes and obtained the results. This method solved the problem to some extent. The precision for class 4 and the recall for class 5 in SGDClassifier were quite low.
Further, on undersampling the data using the Monte-Carlo technique, the results obtained were satisfactory for all algorithms. Among all the algorithms, Random Forest Ensemble Classifier gave the best results in terms of precision, recall and accuracy.

6. Conclusion

This comparative analysis included :

- **Preprocessing of data:** Apart from the 16 features obtained from the accelerometer, 12 other features were calculated i.e. norm, pitch, and roll for each accelerometer. The data was then normalised using Z-score Normalisation. The major data imbalance in the dataset was handled after trying undersampling, oversampling, and changing class weights.
- **Models used:** The Prediction models used for this task were: Logistics Regression, SGD Classifier, Decision Tree Classifier, and Random Forest.
- **Evaluation of Performance:** To test the model's performance on unseen data, K-fold cross-validation was used. The performance was analyzed using the Confusion matrix, Accuracy, Precision, and F1-score. Random forest and Decision Trees showed great performance over undersampled and oversampled data with Random Forest's performance slightly better.
- **Learning from the project:** Handling class imbalance effectively using various under-sampling and oversampling techniques. Understanding the working of different machine learning algorithms and their implementation. Data visualising and feature reduction.
- **Work-Left:** Understanding and analysing the working of Support Vector Machine (SVM) and Naive Bayes on our dataset. Also, trying some feature selection techniques.
- **Individual Contribution:**
  - Bhavey Wadhwa: Implementing the under-sampling and oversampling techniques, analysing the results and maintaining the progress report.
  - Rohit Makkar: Studying different undersampling and oversampling techniques, analysing the results and maintaining the progress report.
  - Ritik Garg: Studying different machine learning algorithms, compiling the results and preparing the progress report.

References

[1] J Mary Hanna Priyadharshini, S Kavitha, and B Bharathi. Classification and analysis of human activities. In *2017 International Conference on Communication and Signal Processing (ICCSP)*, pages 1207–1211. IEEE, 2017. 1

[2] Wallace Ugulino, Débora Cardador, Katia Vega, Eduardo Velloso, Ruy Milidiú, and Hugo Fuks. Wearable computing: Accelerometers' data classification of body postures and movements. In *Brazilian Symposium on Artificial Intelligence*, pages 52–61. Springer, 2012. 1

[3] Ting Zhang, Wenlong Tang, and Edward S Sazonov. Classification of posture and activities by using decision trees. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4353–4356. IEEE, 2012. 1