Final Report

Vaasa University/Machine Learning ICAT3120

Human activity measurement using a smartphone.

Clustering with Machine Learning

Author: Markku Pulli

14.4.2021

==========

Abstract

Goal of the project is to separate input data to six different clusters according to human activity. Two different unsupervised algorithms are tested. Different scenarios we tested. Using the sample data without any dimensionality reduction**[1]** (DR) algorithm like feature extraction or -selection. Feature extraction called Linear Discriminant Analysis (LDA) was used to reduce dimensionality. After multiple rounds of parameter optimization on each algorithm only LDA shows best separation outcome and performance from all of these. Focus on this report is to compare non-dimensionality algorithm and LDA impact on performance. DBSCAN algorithm is used for clustering on both cases. DBSCAN parameter tuning is very time-consuming task. Using Python coding range of different parameters were implemented, and results analyzed. DBSCAN in the best case can create five clusters with silhouette coefficient**[2]** 0.805. As data follows normal distribution Gaussian Mixture Model (GMM) is another clustering algorithm what was tested. DBSCAN vs. GMM Model accuracy is 80.6% vs. 82.6%.

Introduction

Dataset – The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.

Python 3.8 and Jupyter notebook was used to complete all the ML related coding. Code is available separately. Main objective was to compare ML algorithms code efficiency was not the highest priority.

Data Observations

Data shape is 7532x562. Total 561 features and the activity label. Sample count is 13-fold comparing to dimension. Feature extraction is recommended to improve model fit and prediction efficiency. DBSCAN performs better with low dimensionality. It will also reduce the computation time. Less data as “duplicated” data is processed with LDA.

Activity	Description	Samples	share %
1	Walking	1226	16.7%
2	Walking upstairs	1073	14.6%
3	Walking downstairs	986	13.4%
4	Sitting	1286	17.5%
5	Standing	1374	18.7%
6	Laying	1407	19.1%
Total		7352	100.0%

Table 1. Activity distribution.

According to data source documentation the data is already scaled and normalized. This is also verified. As seen from Figure 1. it is visible the data is overlapping between activities and data density is mostly high. Outlier count is low considering total data volume.

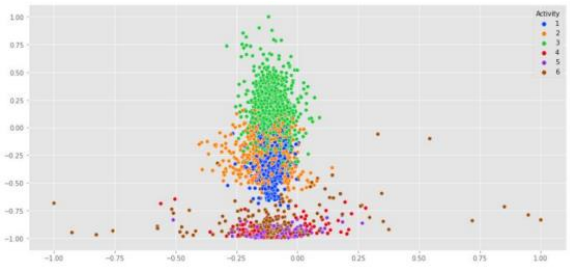


Figure 1. Sample distribution by activity.

LDA component count selection was completed with help of variance ratio. Variance target cover was set to 99%. After iteration it was found four components is adequate count. Variance of each component 1,2,3,4 = [0.73276802, 0.17514607, 0.05460691, 0.02842483] => 99.1%. New data di mension is only 0.7% of the original 561 features with almost no information loss.

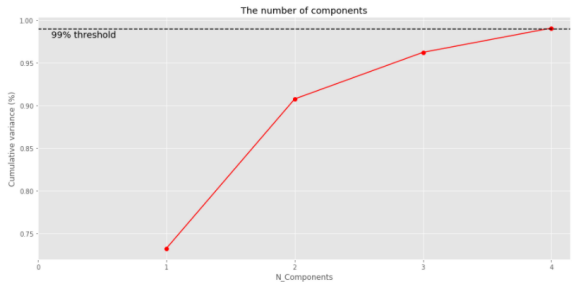


Figure 2. Sample variance by component count.

Transformed data is shown in figure 3. Separatio n between clusters is much more visible. Cluster s 4 and 5 still have overlap. Activities 4 & 5 are 'si tting' and 'standing'. Both are mostly still, minim um movement activities so sensor reading is in s ame range. Data separation is more challenging. Other clusters are well separated.

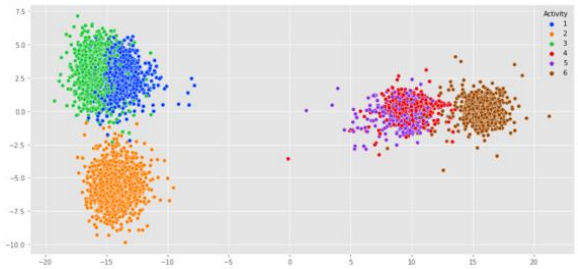


Figure 3.

Link to the dataset: https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

Methodology and Modeling

It was set as an input requirement to use DBSCAN algorithm for clustering the data. DBSCAN is density-based algorithm. DBSCAN works on arbitrary-shaped distributions. Dense areas must be connected. Dimension reduction like LDA is required with high dimension data. Algorithm does not assign outliers to clusters. Number of clusters is known from training data. DBSCAN has two parameters to tune. Distance to the nearest neighbors and number of neighbors for initial cluster formation. Same methodology is used around core points until no new points are reachable. Figure 4.

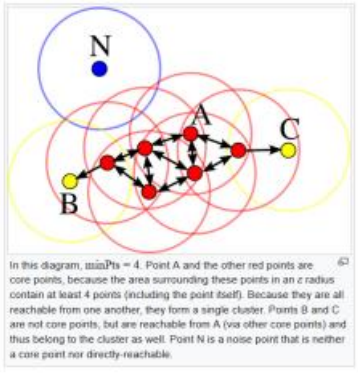


Figure 4. [Source: Wikipedia]

$$N(p) = \{q \in D \mid dist(p, q) \leq \epsilon\}$$

D = dataset, N = ϵ-neighborhood, p = core object, q = density reachable object, ϵ = circle with radius ϵ

Nearest neighbor algorithm is used to find initial set of DBSCAN parameters, see figure 5. Two

parameters are eps and min_samples. Eps is the distance between samples and min_sample is minimum required number of samples to create a cluster.

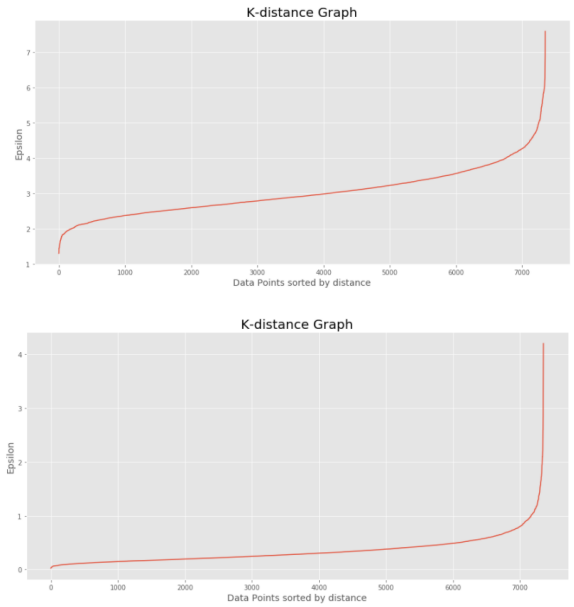


Figure 5.

Initial set of parameters was set to eps=[0.9…1.8] step=0.1 and min_samples=[4…16] step=2. Table 3 shows the cluster count DBSCAN has created for each parameter pair. We know from the dataset correct cluster count is six. Without LDA cluster count varies from 2 to 50. This is indication of very unstable environment. When LDA is used the cluster count is from 5 to 12, much smaller variance. DBSCAN is based on distance metric called Euclidean distance. Below formula is for n-dimensional space.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_i - q_i)^2 + \cdots + (p_n - q_n)^2}.$$

Data with high dimensionality it is difficult and more complex to find appropriate ϵ value to start with.

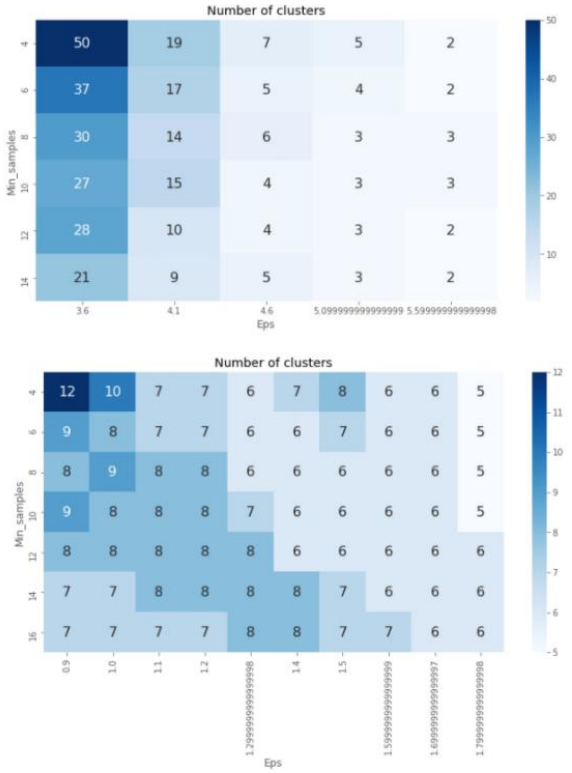


Table 2. Cluster count for each parameter pair.

Table 3 includes silhouette coefficient score for same. Range is {-1…1}. Score close to ‘1’ is most optimal. Top table without DR has highest score 0.45. This is with three separated clusters. With LDA score goes up to 0.8 while cluster count is 6. This proofs the power dimensionality reduction.

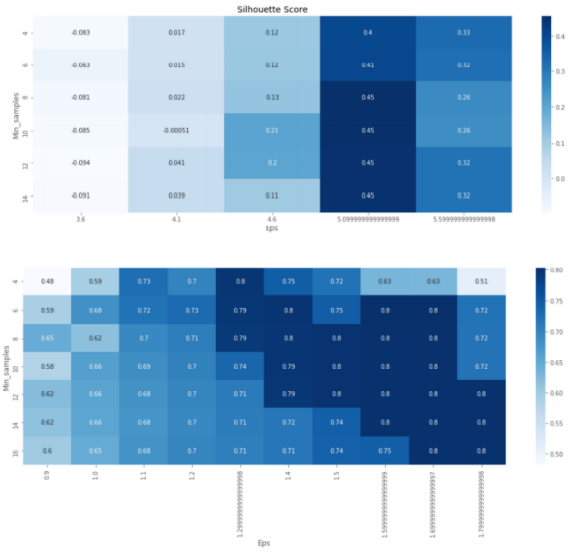


Table 3. Silhouette coefficient for each parameter set.

From the table 3 following parameter sets are selected. Highest Silhouette coefficient score will produce best results.

Non-DR: eps=5.1 and min_samples=10 is chosen, silhouette coefficient score is 0.45.

With LDA: eps=1.6 and min_samples=10 is chosen, silhouette coefficient score is 0.8.

As seen non-DR case highest silhouette coefficient score can separate only 3 clusters and the score=0.45 is still extremely low. We can make conclusion non-DR case is not optimal to use in this situation.

Computation time was also measured for both cases. Parameter tuning computation time with dimensionality reduction is only 6% of the non-DR. Model fit time is less than 3%.

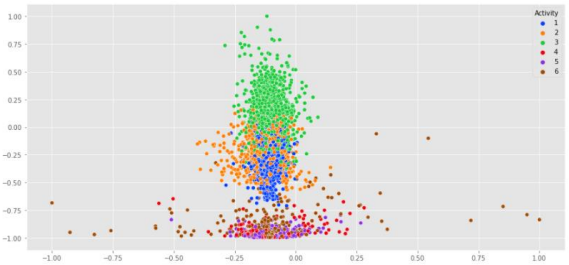
DBSCAN Computation times	DBSCAN Parameter tuning	Final DBSCAN fit [s]
Without dimensionality reduction	4145s, 30 sets	138
LDA	226s, 70 sets	3.2

Table 4. Computation time.

It was also noticed if data is scaled after LDA the computation time is about 1/3 of the previous effort. But the clustering silhouette coefficient was 2% lower with scaled data. This time no scaling performed after LDA. Original dataset was already scaled.

DBSCAN Training Results

Below graphs have results for both cases. These are plots with highest silhouette coefficient score. Even when performing more parameter changes on non-DR results are not improving. With LDA results are significantly better.



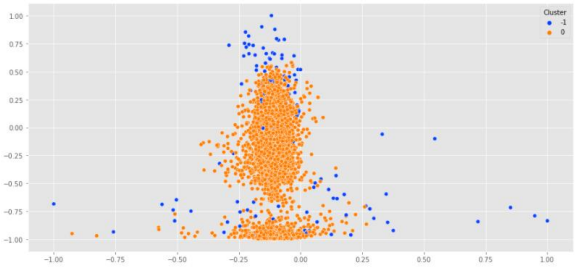


Figure 6.1: DBSCAN only, no DR. Top original clustering and bottom clustering using algorithm.

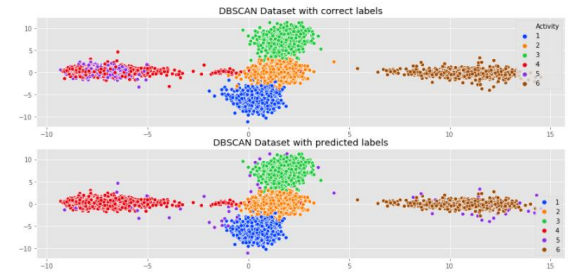


Figure 6.2: DBSCAN + LDA. Top original clustering and bottom clustering using algorithm.

KPI	DBSCAN Only	DBSCAN + LDA
Estimated number of clusters:	3	5
Estimated number of noise points:	499	81
Homogeneity:	0.383	0.85
Completeness:	0.749	0.962
V-measure:	0.507	0.903
Adjusted Rand Index:	0.325	0.787
Adjusted Mutual Information:	0.507	0.903
Silhouette Coefficient:	0.197	0.8

Table 5: DBSCAN performance comparison, non-DR vs LDA. See scikit learn clustering metrics for details.

When comparing result between two exercises it is clearly seen when sample data feature count is high, dimensionality reduction algorithm has high impact on model fit performance. Computation time is fraction of the non-DR case and accuracy is much better. This is especially important when implementing algorithm in a production environment.

Test Data Verification, LDA+DBSCAN

Using the test data same model performance is seen as during the training. Cluster separation is clear.

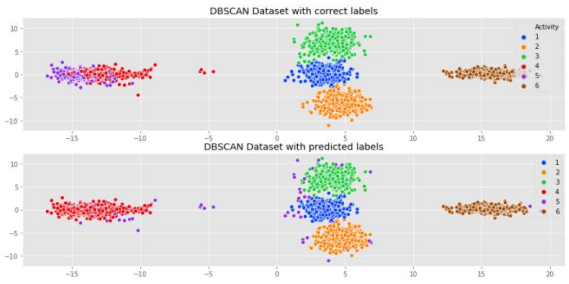


Figure 7. Test data clustering. Top original clustering and bottom clustering using algorithm.

Estimated number of clusters: 5
Estimated number of noise points: 61
Homogeneity: 0.849
Completeness: 0.940
V-measure: 0.892
Adjusted Rand Index: 0.794
Adjusted Mutual Information: 0.892
Silhouette Coefficient: 0.825

Table 6. Test set performance table.

As seen on table 6 test silhouette coefficient is higher than training coefficient. The model performance is verified and approved.

Gaussian Mixture Model

Gaussian Mixture model (GMM) is implemented to compare results between two different clustering algorithms. According to scikit learn 'A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters.' First let us have a look how is the distribution of the data. First three features are selected. These include all three dimensions of sensor data; X, Y and Z.

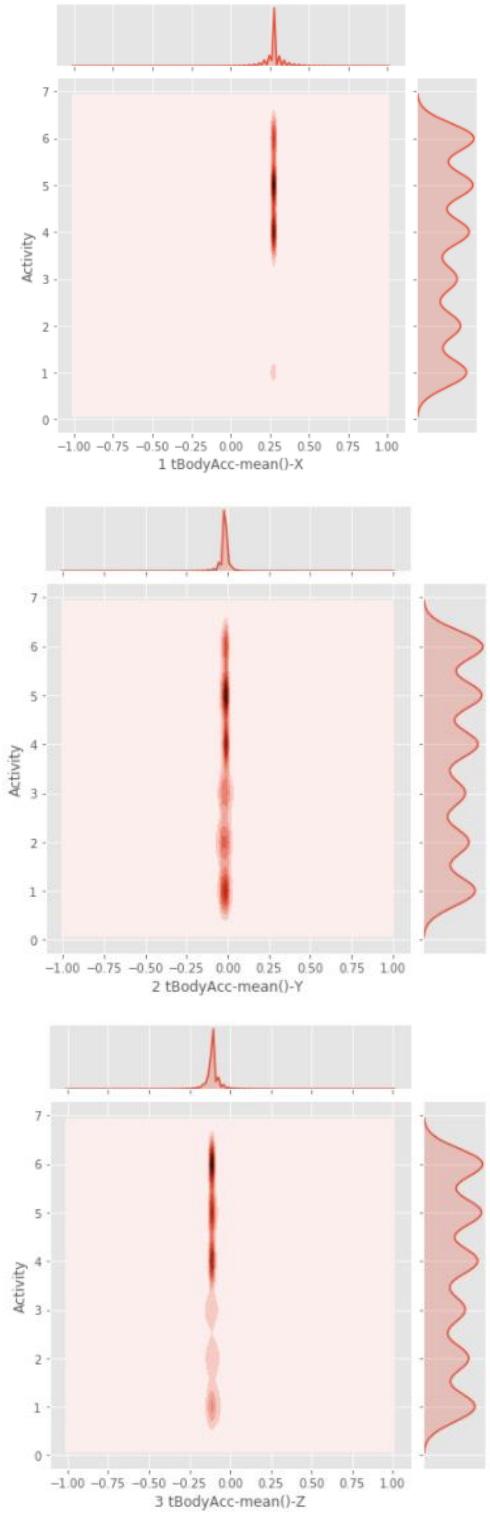


Figure 8. Activity sensor data distribution.

As seen from Figure 8. graphs data follows well gaussian distribution so GMM clustering is expected to perform well. Using BIC, Bayesian

information criterion[3], score we can find the optimal cluster count.

$$\mathrm{BIC} = k\ln(n) - 2\ln(\widehat{L})$$

$\hat{L}$ = the maximized value of the likelihood function of the model M. $\hat{L} = p(x \mid \hat{\theta}, M)$, where θ are the parameter values that maximize the likelihood function.

x = the observed data.

n = the number of data points in x, the number of observations, or equivalently, the sample size.

k = the number of parameters estimated by the model. For example, in multiple linear regression, the estimated parameters are the intercept, the q slope parameters, and the constant variance of the errors; thus, k = q + 2.

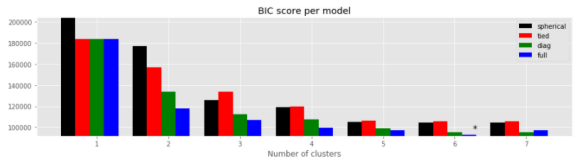


Figure 9. BIC and covariance type selection.

We get six and this matches the dataset cluster count. Best suited covariance type is 'full', each component has its own general covariance matrix.

GMM_log-likelihood Score

Log-likelihood is natural logarithm of expectation Maximization likelihood function. Higher log-likelihood value corresponds to the better model fit. Target is to maximize it. With log likelihood log values are added to total instead of multiplying. This will prevent multiplying numbers close to zero and handling very small numbers. Log likelihood is mostly useful when comparing different model performance as there no real meaning expect weight of evidence.

$$F(\theta) = \sum_{i=1}^{n} \ln f_i(y_i \mid \theta)$$

Log likelihood algorithm.

Results and Conclusions

Accuracy of the algorithms can be found in table 7 below.

Item	Test Accuracy [%]
DBSCAN	80.5
GMM	82.6

Table 7. Model test accuracy

Accuracy of DBSCAN and GMM models is in same range. Both models are struggling to separate sitting (4) and standing (5) activities. GMM is performance is marginally better.

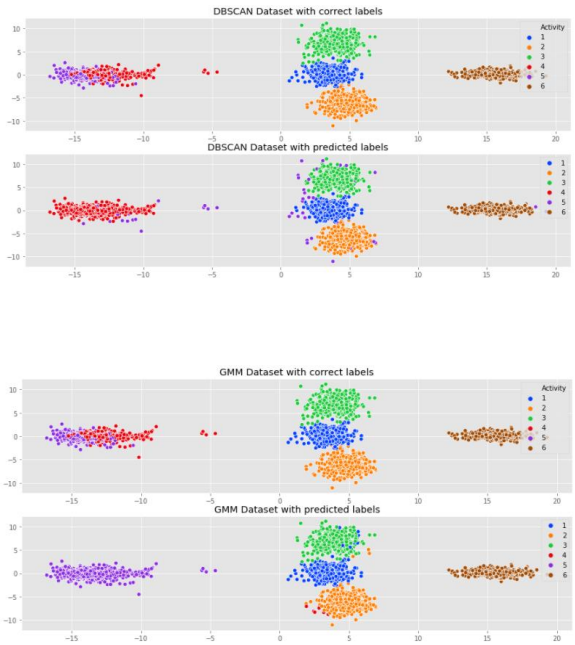


Figure 10. Clustering results of two algorithms.

Data extraction played vital role as without it model accuracy is poor and computation time is 10-fold.

Recommendations and Improvements

One possible approach improve results is breaking down the different measurement equipment readings. Currently all accelerometer and gyroscope measurements are extracted to four components without any detail analysis. Analyzing data further it is possible there is hidden information available to separate clusters

more efficiently and improve model accuracy. Also there are other clustering algorithms available like K-means.

References

[1] Dimensionality reduction, or dimension reduction, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension. Working in high-dimensional spaces can be undesirable for many reasons; raw data are often sparse as a consequence of the curse of dimensionality, and analyzing the data is usually computationally intractable. Dimensionality reduction is common in fields that deal with large numbers of observations and/or large numbers of variables, such as signal processing, speech recognition, neuroinformatics, and bioinformatics. [Wikipedia].

[2] The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient for a sample is (b - a) / max(a, b). To clarify, b is the distance between a sample and the nearest cluster that the sample is not a part of. Note that Silhouette Coefficient is only defined if number of labels is 2 <= n_labels <= n_samples - 1. The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar. [scikit- learn].

[3] In statistics, the Bayesian information criterion (BIC) or Schwarz information criterion (also SIC, SBC, SBIC) is a criterion for model selection among a finite set of models; the model with the lowest BIC is preferred. It is based, in part, on the likelihood function and it is closely related to the Akaike information criterion (AIC). When fitting models, it is possible to increase the likelihood by adding parameters, but doing so may result in overfitting. Both BIC and AIC attempt to resolve this problem by introducing a penalty term for the number of parameters in the model; the penalty term is larger in BIC than in AIC. [Wikipedia].