Pattern Recognition - Assignment 1

Group 30

Reetta Nätkin
Meri Mäkelä
Arttu Linden
Tatu Mäkinen

February 2019

For the presented results we use cross validation with *GroupShuffleSplit (n_splits=30, test_size=0.2, random_state=0)* as our evaluation metric, and *RandomForestClassifier (n_estimators=100)* as our standard classifier, *i.e.*, if no other classifier is specified.

1 The problem

In this work we aimed to classify floor surfaces based on IMU sensor data attached to a robot.

The data contained:

- 9 floor types to classify:
  1. carpet
  2. concrete
  3. fine concrete
  4. hard tiles
  5. hard tiles, large space
  6. soft pvc
  7. soft tiles
  8. tiled
  9. wood
- 10 sensor channels:
  - orientation: X, Y, Z, W
  - angular velocity: X, Y, Z
  - linear acceleration X, Y, Z

- 36 data groups, as raw data was from 36 different recordings
- 1703 training samples and corresponding labels

The data contained 128 temporally consequent samples each from the 10 sensor channels - Input data dimensions were thus **1703x10x128**. Example of data contained in one sample is provided in figure 1.

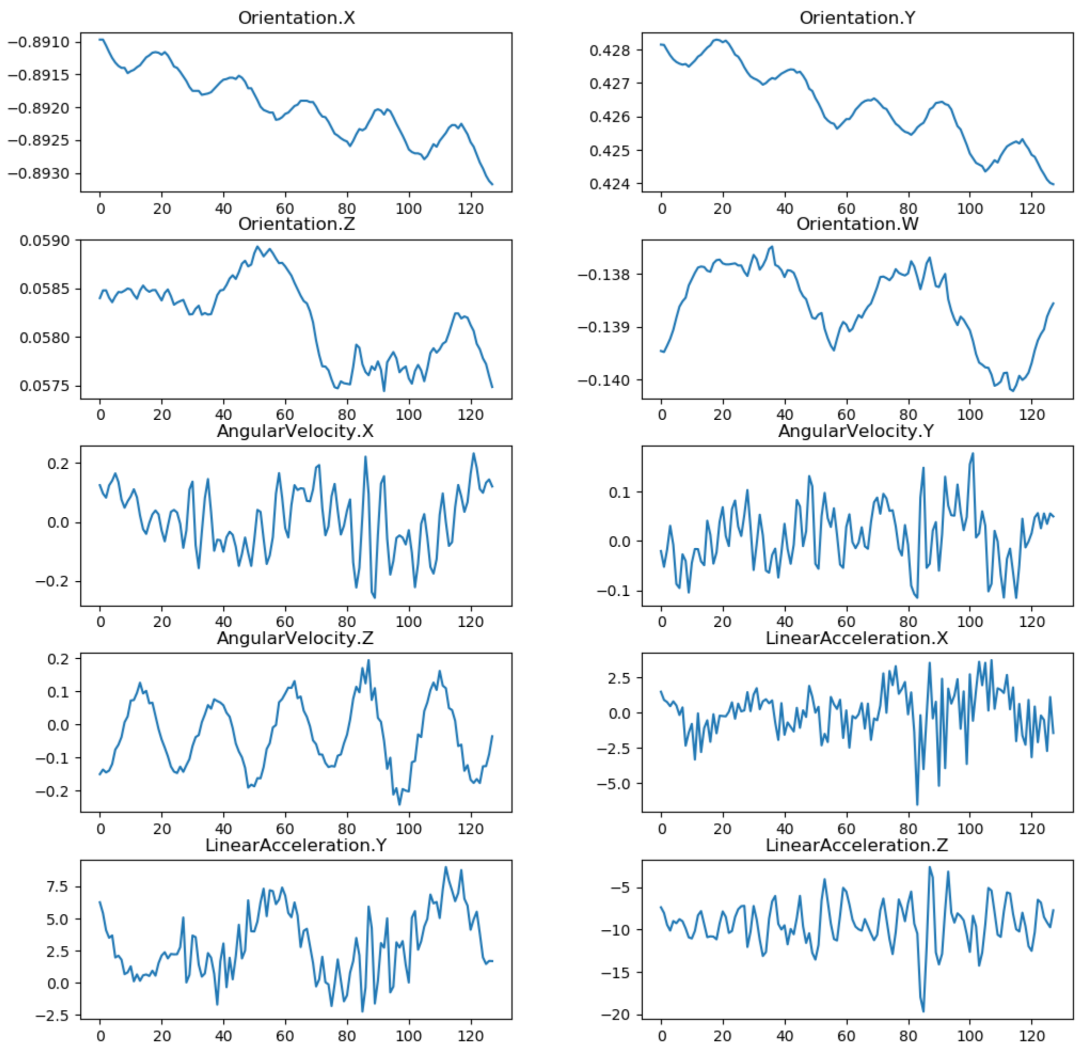

Figure 1: Example of the raw data.

2 Classifiers

We compared the readily available classifiers in scikit-learn in their raw form using mean and standard deviation of the channels as the feature. In addition to the basic classifiers, we tested XGBoost classifier as it has previously given good results. The results are presented in table 2.

Accuracy (mean+std)	Accuracy (raw data)	Classifier
0.4322	0.3450	Random Forest Classifier (100 estimators)
0.4220	0.3502	Extra Trees Classifier (100 estimators)
0.4115	0.3464	XGB Classifier (100 estimators)
0.3991	0.3319	Gradient Boosting Classifier (100 estimators)
0.3908	0.3197	SVC (rbf)
0.3862	0.2102	SVC (poly)
0.3807	0.1118	KNeighbors Classifier (5 neighbors)
0.3704	0.1844	SVC (linear)
0.3641	0.2265	MLP Classifier
0.3405	0.1632	Linear Discriminant Analysis
0.3277	0.1809	Logistic Regression
0.3173	0.1695	OneVsRest Classifier with Linear SVC
0.3171	0.1826	Linear SVC
0.2427	0.2658	AdaBoost Classifier (100 estimators)
0.2388	0.1961	SGD Classifier
0.0842	0.0842	SVC (sigmoid)

Table 1: Basic classifier comparison using *mean+std* and *raw data* as features.

Based on this evaluation it seems that of the basic classifiers, ensemble methods work the best.

3 Data preprocessing

When there is a limited amount of data available, data preprocessing and feature selection can be used to compress the information in the data as classifiable properties.

3.1 Sensors

First it should be analyzed if all the sensors provide useful information. We did this by testing all the raw data combinations of the three channel types using random forest classifier. The results are presented in table 3.1.

Accuracy	Combination
0.3459	velocity+acceleration
0.3447	velocity
0.3442	all data
0.3424	orientation+velocity
0.3373	orientation+acceleration
0.3364	acceleration
0.3186	orientation

Table 2: raw data

Accuracy	Combination
0.4510	velocity+acceleration
0.4360	velocity
0.4294	orientation+acceleration
0.4291	all data
0.4111	orientation+velocity
0.3911	acceleration
0.3578	orientation

Table 3: mean + std

Table 4: Sensor combination comparison using *raw data* and *mean + std* as input.

Based on these results, it seems that the order of importance is 1) velocity, 2) acceleration, 3) orientation.

3.2 Channels

To further analyze the importance of individual channels, we measured the change in standard classifier accuracy when one of the channels is left out. Positive change indicates that the left out channel was not important (rather it just provided noise), while negative change indicates that it included useful information. Results are provided in table 3.2.

The results are within the margin of error of the evaluation. We also tested using raw data as features, and this resulted in an arbitrary order of the channels. Nevertheless, it seems that orientation data is still the least significant, and leaving it out might be beneficial.

3.3 Frequency domain

Quick search for the scientific articles related to inertial measurement unit (IMU) surface classification suggests that frequency domain might contain useful information [1]. This seems reasonable, as different surfaces might cause different vibrations. Vibrations should manifest themselves at least in the velocity and acceleration data.

Accuracy	Channel left out
0.4424	orientation Z
0.4368	orientation X
0.4359	orientation Y
0.4308	velocity X
0.4306	orientation W
0.4304	velocity Z
0.4294	velocity Y
0.4291	**all data**
0.4279	acceleration X
0.4263	acceleration Y
0.4229	acceleration Z

Table 5: Effect of leaving out channels using *mean + std* as input features.

In figure 2 is presented an example of the power spectral densities of the data. The densities are scaled to log space to reduce dominance of the high magnitude frequencies.

As expected, orientation data contains very little information in the frequency domain. Velocity and acceleration data seem to have spikes, which might be useful in the classification process.

Results from testing the classifiers for the Fourier transformed data and power spectral densities are presented in table 3.3.

The results are overall higher than for the raw data, which suggests than frequency domain contains useful information. There are some differences for FFT and PSD as features. FFT seems to perform slightly better on average. However, for some classifiers the accuracy with PSD is significantly better. It should also be noted that results for XGB and SVC with polynomial kernel are flawed and correspond to random guess accuracy.

3.4 Principal component analysis

Principal component analysis can be used to reduce the dimensionality of the feature space. Principal components are orthogonal vectors that maximize the variance with regards to data. Variables that correlate with chosen principal components are left out, and thus noise caused by them can be decreased.

We performed an exhaustive search for the number of principal components using our standard classifier. The results are presented in figure 3.

The optimal amount of principal components seems to be around 25 for the FFT transformed data, after which the accuracy slowly decreases.

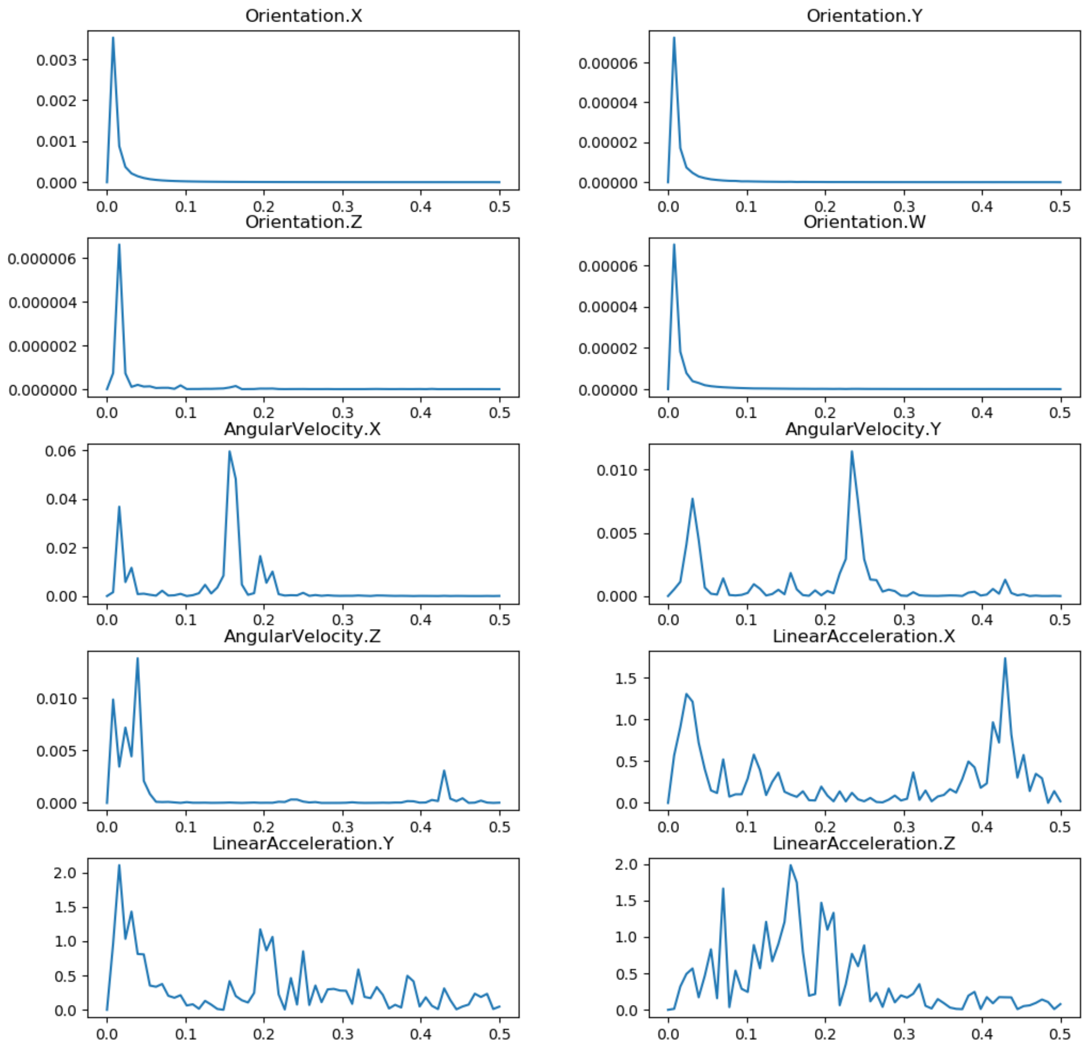

Figure 2: Example of power spectral densities of the data in log scale.

3.5 Features

We tested multiple features calculated from the raw data and the data in the frequence domain. The results are presented in table 3.5.

Based on this results, raw FFT seems to be the best feature to use in classification. Highest magnitude peaks in the frequence domain seemed empirically good for classification. Nevertheless, these results state otherwise.

Accuracy (FFT)	Accuracy (PSD)	Classifier
0.5114	0.5009	MLP Classifier
0.4884	0.4765	Gradient Boosting Classifier (100 estimators)
0.4847	0.1677	SVC (poly)
0.4765	0.4636	Random Forest Classifier (100 estimators)
0.4758	0.4104	KNeighbors Classifier (5 neighbors)
0.4625	0.4318	SVC (rbf)
0.4564	0.4545	Extra Trees Classifier (100 estimators)
0.4552	0.4632	SVC (linear)
0.4168	0.3888	SVC (sigmoid)
0.4154	0.4704	Logistic Regression
0.4149	0.4357	Linear Discriminant Analysis
0.3691	0.4555	OneVsRest Classifier
0.3665	0.4567	Linear SVC
0.3630	0.3636	AdaBoost Classifier (100 estimators)
0.3261	0.3632	SGD Classifier
0.1676	0.1676	XGB Classifier (100 estimators)

Table 6: Basic classifier comparison using FFT and PSD scaled to log space as features.

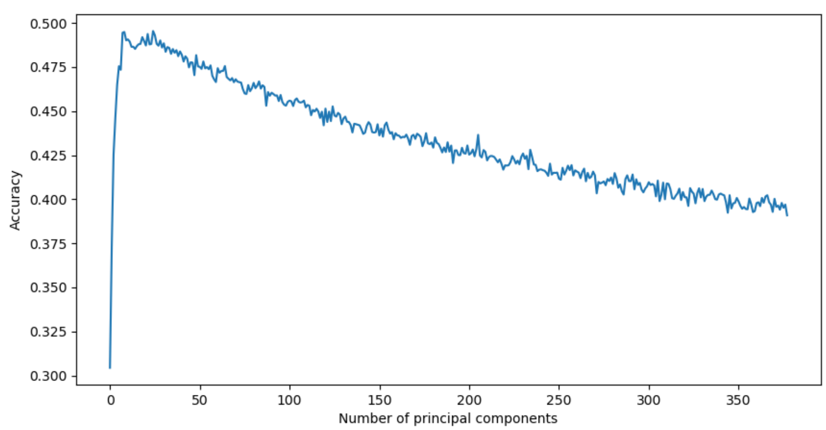

Figure 3: Effect of the number of principal components on accuracy.

Accuracy	Feature
0.4767	FFT (log10)
0.4759	FFT (pow2)
0.4747	FFT
0.4706	FFT (max 2 peaks, std, mean)
0.4634	FFT (max 2 peaks, std, mean) + raw (mean, std)
0.4617	PSD (log10)
0.4613	mean, std, max 3 FFT peaks
0.4590	mean, std from raw and FFT
0.4581	FFT (log10) + raw
0.4580	PSD
0.4538	mean,std from FFT
0.4345	mean, std
0.4264	std
0.3488	mean
0.3480	all
0.2865	max 3 FFT peaks
0.2827	max 2 FFT peaks

Table 7: Feature comparison.

4 Kaggle

We submitted prediction results for the best classifiers based on our analysis. The submissions are shown in table 4.

Despite the work we put into feature selection, we achieved best results in the leader board by using basic standard deviation and mean of the raw data with Random Forest classifier. The results suggest that our cross validation setup did not measure real performance well. The scale and order of the classifiers are different for the real scores and the test scores.

We also experimented using data augmentation, in which we used predicted labels of which the classifier was $> 80\%$ sure in retraining the classifier. This gave us minor 0.4 % increase in accuracy for the XGBoost classifier.

Accuracy	Accuracy (own test)	Classifier	Feature
0.67995	0.43025	Random Forest (100)	mean+std
0.67174	0.50074	XGBoost (200)	FFTlog10, aug
0.67057	0.41514	XGBoost (300)	mean+std, aug
0.66705	0.50074	XGBoost (200)	FFTlog10
0.62485	0.43455	XGBoost (300)	FFT(2peaks+std+mean) + mean+std)
0.54513	0.49423	GradientBoost (300)	FFTlog10, aug, PCA=30
0.42203	0.51672	MLP Classifier	FFTlog10, PCA=30
0.78194	-	1st ranked submission (10.2.2019)	-

Table 8: Public leader board scores for our submissions (aug = augmented).

References

[1] C. Weiss, N. Fechner, M. Stark, and A. Zell. Comparison of different approaches to vibration-based terrain classification. In *EMCR*, 2007.