Urban Objects Classification from AirBorne LiDAR Data

GEO5017 A2

Katrin Meschin - 5163889

TU Delft

March 2022

1 Introduction

LIDAR point clouds are increasingly used for modelling urban environment. Through classification, these massive point cloud datasets can be converted into meaningful and useful information. In this study, supervised classification using random forest classifier and support vector machine are used to assign a class 500 set of point clouds, each representing an object from urban environment. The feature design and feature selection process is explained, followed by an overview and a conclusion of the experiment for determining the learning curve. The ultimate goal for this assignment was to find a set of features and a classification technique that would produce the best classification accuracy with the least training data possible.

2 Methodology

The choice for the features was motivated by the outcome of the last assignment in the course and a paper by Weinmann et al., 2013 that discusses the relevance of features for semantic interpretation of 3D point cloud data. First convex hull for each point cloud set was calculated using an open Scipy Spatial version 1.8.0 (Virtanen et al., 2020) library. The area and the volume of the convex hull was saved as two of the features. Convex hull represents the feature's shape well, especially considering that the given data sets are clean without significant outliers that would distort the convex hull. Both area and volume were chosen as for objects like fences, the area would be big but the volume small, therefore these two features can give good indication of the shape of the objects. In addition six point cloud geometric features were added, namely Eigenvalue sum, omnivariance, eigentropy, anisotropy, planarity and linearity, which were also used by Weinmann et al., 2013. These features were calculated using Jakteristics open library (Caron and Messal, 2022).

2.1 Feature selection

Weinmann et al., 2013 concluded on their study on feature relevance for semantic interpretation of 3D point cloud data that better accuracy can be achieved with a few carefully chosen features, instead of including large set of features in order to compensate for lack of knowledge. The aim of feature selection is to select a compact subset of the most relevant feature for classification and discard redundant and irrelevant information. In this study, a method from Scikit Learn was used to select three most important features by SelectKBest function. The score function was chosen to be F-value as it had also been used by Weinmann et al., 2013. F-value is given when running an ANOVA test and it highlights the *significance* of each feature. The feature selection was done in the training phase and the selected features were used for classifying the testing data set.

From the eight features computed for each point cloud, the SelectKBest function chose three which were the sum of Eigenvalue, planarity and linearity. It is notable, that linearity was one of the features left as it was noted by Weinmann et al., 2013 to have little effect on their classification results.

3 Experiment and evaluation

3.1 Experiments and classification results

Two classifiers were used - random forest classifier (rfc) and the support vector machine (svm) classifier from SciKit learn library Pedregosa et al., 2011. The accuracy of these classifiers were estimated by calculating both the overall accuracy (OA) and the mean per class accuracy (mA).

$$OA = \frac{1}{N}\sum_{i=1}^{C} n_i \tag{1}$$

$$mA = \frac{1}{C}\sum_{i=1}^{C} \frac{n_i}{N_i} \tag{2}$$

where C is the total number of classes, n_i is the number of objects correctly classified in i-th class, N is the total number of objects while N_i is the total number of objects in i-th class. However, since in this study every class has the same number of objects, OA is almost always equal to mA.

For SVM classifier, kernel had to be chosen. For this, five trial runs were made with each of the four possible kernels provided by Scikit Learn and the overall accuracy was assessed. These tests were made when already three features were selected for the program. The kernel with the highest mean overall accuracy from the tests was chosen, therefore polynomial kernel was used for this study.

Kernel function	Test 1	Test 3	Test 3	Test 4	Test 5	Mean
Linear	0.88	0.705	0.695	0.615	0.625	0.704
Polynomial	0.925	0.89	0.905	0.845	0.88	0.889
rbf	0.775	0.775	0.73	0.815	0.775	0.774
Sigmoid	0.175	0.18	0.175	0.22	0.345	0.219

3.2 Visualisation and analysis of learning curve

Once three of the features were chosen for the classification and the polynomial kernel for the svm-classifier, tests were made to see how decreasing the training sample affects the accuracy of the results. The ratio of the training data was decreased from 60% to 10% whereas each time the program was ran three times. The results are shown in Figure 1. The highest overall

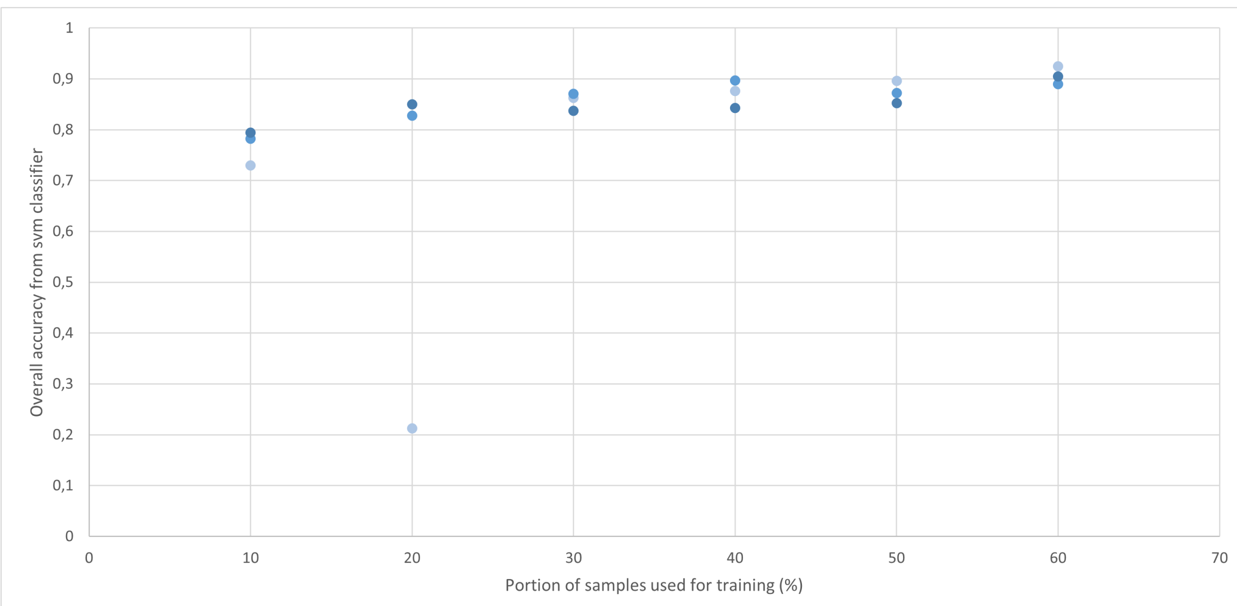

Figure 1: Accuracy of the svm classifier as the training data was decreased when only 3 best features are used

accuracy was recorded when 60% of the input data was used for training and 40% was used for testing, in which case the overall accuracy of the svm classifier was around 90%. When the training data was decreased, the accuracy declined slowly. However, even with only 30% of data used for training and 70% for testing, the accuracy was still around 85%. However, it should be noted that when the training data is small, the results are unpredictable which is evident in the test with 20% training data when one of the test only had 20% of accuracy while the two other tests with the same conditions yielded over 80% of accuracy. This is due to the program choosing the training data at random and with small training set, the probability of training data not being spread out across all classes is higher.

In comparison, figure 2 shows the learning curve in case all features are used. it is clear that using all of the eight features yields significantly lower accuracy for the classification. Even using more training data does not improve the accuracy when all the designed features are used.

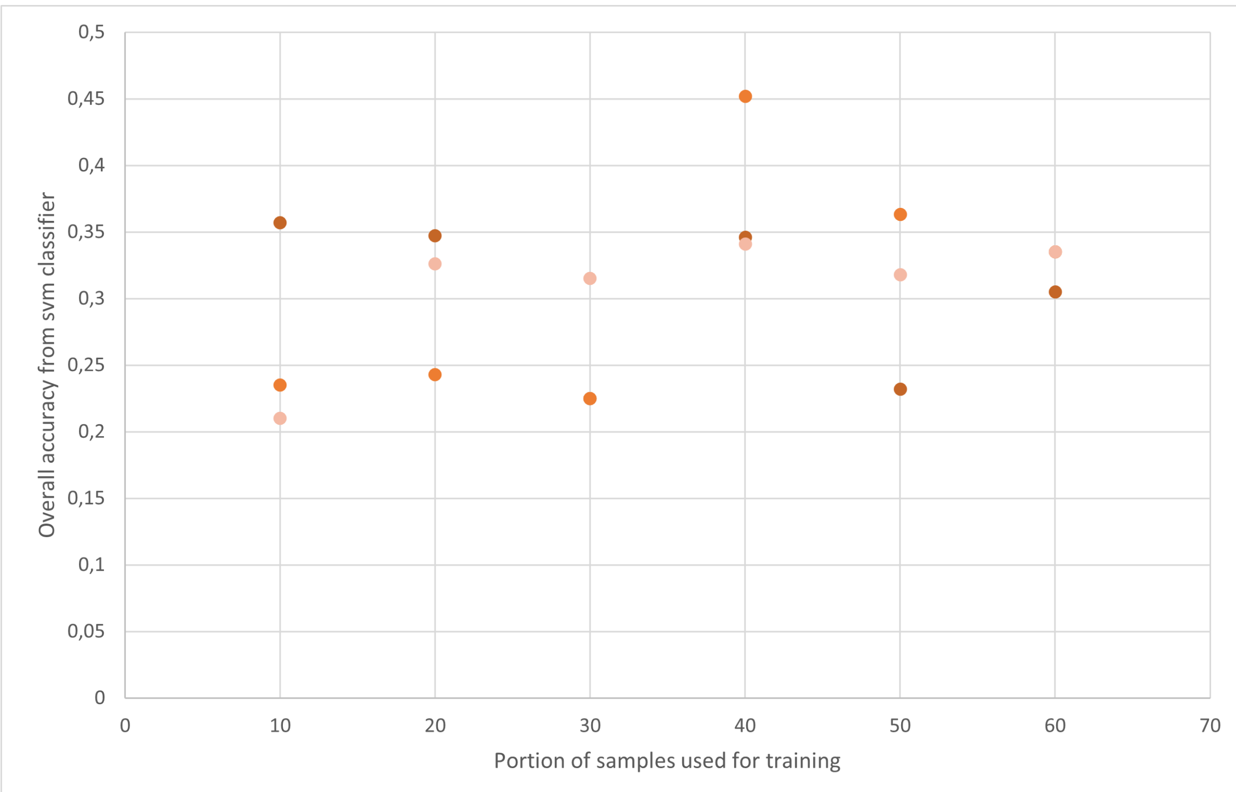

Figure 2: Accuracy of the svm classifier as the training data was decreased when all features are used

4 Conclusion

This study used eight features for 500 point clouds, namely the area and volume of the convex hull, Eigenvalue sum, omnivariance, eigentropy, anisotropy, planarity and linearity. These features were used to train a svm and a random forest classifiers, whereas polynomial kernel was used for the svm classifier as it yiedled the best accuracy. The importance of the features were analysed using Scikit Learn's SelectKBest function together with F-value score function to get three most meaninful features. The accuracy of the classifiers were assessed by calculating the overall and mean per class accuracy as well as confusion matrix. The learning curve was then analysed by performing series of tests where the training data set was decreased. The outcome of this study shows that with the three chosen features - sum of Eigenvalue, planarity and linearity, the overall accuracy remained above 80% even when only 30% of the samples were used for training. Once the training data was decreased, there were inconsistencies in the accuracy yielded. However, figures 1 and 2 provided in this study show clearly that only picking good and most important classifiers for the classification can significantly improve the results. In this study, when all the features were used for the study, the accuracy did not reach above 40% whereas when only 3 most important features were used, the accuracy reached over 80%.

This report and the code is entirely my work.

References

Caron D. and Messal M. (2022). *Jakteristics*. https://github.com/jakarto3d/jakteristics.

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

Virtanen P., Gommers R., Oliphant T. E., Haberland M., Reddy T., Cournapeau D., Burovski E., Peterson P., Weckesser W., Bright J., van der Walt S. J., Brett M., Wilson J., Millman K. J., Mayorov N., Nelson A. R. J., Jones E., Kern R., Larson E., Carey C. J., Polat İ., Feng Y., Moore E. W., VanderPlas J., Laxalde D., Perktold J., Cimrman R., Henriksen I., Quintero E. A., Harris C. R., Archibald A. M., Ribeiro A. H., Pedregosa F., van Mulbregt P., and SciPy 1.0 Contributors (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". *Nature Methods* 17, pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

Weinmann M., Jutzi B., and Mallet C. (2013). "Feature relevance assessment for the semantic interpretation of 3D point cloud data".