GEO5017 A2
Urban Objects Classification from AirBorne LiDAR Data

1 Introduction

We live in a 3 Dimensional (3D) world that is composed of urban entities, such as buildings, lampposts, mailboxes, and vegetation. During the last decades, there is an ever-increasing demand, both by academia and industry, for analyzing and interpreting 3D urban geoinformation. A variety of research works have been devoted to the problem of 3D scene understanding. Successful interpretation of urban scenes allows for various modern applications such as urban modeling, autonomous driving, and environment monitoring.

Understanding the urban environment from 3D airborne LiDAR data has been extensively studied in recent years. LiDAR point clouds are vastly available in many countries. Additionally, compared to 2D imagery, point clouds provide more accurate 3D measurements with higher spatial resolution. Classification of point clouds is a fundamental task in 3D scene understanding, which aims to assign each point cloud a meaningful semantic label (e.g., building, tree, car). Semantic segmentation is a more complex task, assigning per point a label. In other words, segmentation can be viewed as dense classification on the point level.

In A1, we focused on unsupervised clustering of airborne point clouds. In this assignment, we will experiment with the supervised classification of LiDAR point clouds.

2 The tasks

2.1 Feature engineering

You will need to design a set of features for the classification. In this assignment, we ask you to come up with at least 6 features. You can reuse some or all the features designed in A1. Please refer to the papers introduced in the lectures for designing new features[1].

[1]References
- SUM: A benchmark dataset of Semantic Urban Meshes.
- Eigen-feature analysis of weighted covariance matrices for LiDAR point cloud classification.
- Feature Relevance Assessment for the Semantic Interpretation of 3D Point Cloud Data.

Again, there are no standard features and we encourage you to be creative to design your own feature set.

2.2 Classification

Carry out point cloud classification using the two classifiers introduced in the course, i.e., SVM and RF. In this assignment, we specifically ask you to use the implementation of the classifiers from Scikit-Learn. For the SVM classifier, please try different Kernel functions, select the most promising one (and justify your choice in the report).

Randomly split the dataset into a training set and a test set (i.e., 6:4). Then for each classifier, train, test, and evaluate the result. Finally, compare the performance of the two classifiers.

For the evaluation, use the metrics described in Section 3.1.

2.3 Feature selection

Are all the features equally important for classification? The answer is "No". Select the 3 most discriminative features, run the training/test again, and compare the performance to that obtained with the original feature set (in Section 2.2).

To select good features, it will be helpful to visualize and observe the feature distributions of different classes to get some intuition. You are encouraged to use the feature selection strategies introduced in the lectures.

3 Evaluation and analysis

3.1 Evaluation metrics

To evaluate the performance of a classifier over the test set, the following metrics are commonly used: overall accuracy (OA), mean per-class accuracy (mA), and confusion matrix.

- **Overall accuracy**

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

where C is the total number of classes. n_i is the number of objects correctly classified in class i. N is the number of objects in total. OA measures the overall performance of the classifier.

- **Mean per-class accuracy**

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

where N_i is the total number of objects in the i-th class. mA measures the per category classification accuracy. It helps to relieve the class imbalance issue by averaging the accuracy of all classes.

- **Confusion matrix**. It has a specific table layout that allows the visualization of the performance of a classifier. Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class[2]. You can compute the confusion matrix by yourself or use Scikit-Learn Metrics.

3.2 Analysis

Your report should also include the following analysis:

- How does feature selection affect the performance of the classifiers?
- From the above experiments, choose the model-feature combination that yields the highest performance. Tune its train test split ratio (i.e., gradually enlarge the training set while decreasing the test set). Compute OA for each split and make a plot of the size of the training set and OA. The plotted curve is also known as the "learning curve" in machine learning. How does the curve behave? Provide your insights on the learning curve?

4 Dataset

We will use the same dataset as in A1 (download A1 again if you don't have it), which contains 500 point clouds of urban objects. For each point cloud, its ground truth label is encoded as follows (based on the base names of the point cloud files):

1) 000 - 099: building;
2) 100 - 199: car;
3) 200 - 299: fence;
4) 300 - 399: pole;
5) 400 - 499: tree.

5 Submission (Due: March. 25th)

Please compress all the following into a single archive titled **GEO5017_A2_Group_0X.zip** (where 'X' is your group ID) and submit it to BrightSpace:

- **A report ($\leq$ 5 pages excluding references)**
  - **Introduction**
    Describe the motivation and goal of this assignment. (5%)

[2] https://en.wikipedia.org/wiki/Confusion_matrix

- **Methodology**
  * Describe your features and explain your choice of the features. (10%)
  * Describe your feature selection process. (10%)
- **Experiments and evaluation**
  * Describe your experiments, classification results (10%)
  * Analysis on how feature affects performance. (10%)
  * Visualize the learning curve and provide your analysis of the curve. (10%)
- **Conclusion**
  * What conclusion can we draw from this assignment? (5%)
  * What could be done to achieve a better result?(5%)
- **A short description of who did what**. (5%)

- **Source code**
  - The source code, archived in a 'code' subfolder. The code should be able to build, run, and reproduce your results without any modification. (30%)
  - [optional] Provide a link to the GitHub repository (only if you use GitHub) in the 'Experiment' section of your report. You are encouraged to collaborate with your teammates on GitHub.