Assignment 7 – Deep Learning for Geospatial Data

Advanced Data Mining

Task 1 [4pkt]

1. Download the dataset from the following link: https://www.kaggle.com/datasets/humansintheloop/semantic-segmentation-of-aerial-imagery
2. Take a first look at the data. Are the classes balanced? Visualize a few interesting examples to better understand the dataset.
3. Build a UNet model in PyTorch. You can use standard layers like `nn.Conv2d`, `nn.MaxPool2d`, and `nn.ConvTranspose2d`. Use Cross Entropy (`nn.CrossEntropyLoss`) as your loss function. Then train the model using the downloaded dataset.
4. After training, visualize some filters from the convolutional layers. Do they seem to detect useful patterns or features?
5. Evaluate the results. Does the model perform better on certain classes? Are there classes that are harder to predict?

Task 2 [5pkt]

Implement the SeqScan clustering algorithm based on the pseudocode on page 16 of the following paper: https://arxiv.org/pdf/1805.02102

Test your algorithm using one of the two options:

- Create a few small sample datasets manually (worth 1 point), or
- Generate your own synthetic dataset, similar to the example shown in Figure 12 of the paper (worth 2 points).

Task 3 [3pkt]

a) **[0.5 points]** What is the time complexity of the approximate algorithm used for trajectory partitioning in the TRACLUS algorithm? Provide a justification for your answer.
b) **[0.5 points]** Give an example where the approximate algorithm for trajectory partitioning fails to find the optimal partitioning.
c) **[1 point]** Let's consider hurricane trajectories. Intuitively, stronger hurricanes should have higher weights when calculating the loss function N_ε . How could you modify a density-based clustering algorithm for line segmentation to take trajectory weights into account?
d) **[1 point]** Is the original distance function used in density-based clustering for line segmentation a proper metric? Use the triangle inequality to provide a counterexample if it is not.

Bonus [5pkt]

Download the dataset from: https://www.kaggle.com/datasets/rhammell/planesnet

Implement a Masked Autoencoder (MAE). You may base your work on one of the available implementations, such as:
https://paperswithcode.com/paper/masked-autoencoders-are-scalable-vision

Test your implementation. How well does your algorithm reconstruct the images? Suggest suitable evaluation metrics to assess the quality of the reconstruction.