Community Detection

Working with community detection data is a great way to get in touch with new concepts of deep learning and explore the Macro-level behavior of human activity in the city of Beijing.

Observations:
Some of the observations that were made when working with FMT code on community discovery are as follows,

1. The IDEC Implementation results show that the predicted groups of clusters have better similarity visually than the standard K-means algorithm.
2. Unsupervised learning makes it harder to understand and comprehend whether the model really performed well since there are no trained labels to compare against.
3. By using pre-trained MNIST autoencoder weights [2], we are limiting the dimensions of the heatmaps to size 28 * 28 - The purpose of the autoencoder is to find a more compact encoding of the input, and by using a really small input, the quality of the decoded output might be inferior (uncertain about the impact)
    a. Training an autoencoder from scratch for large image sizes ie. 100 * 100, 300 * 300 could be an option.
4. Using an MNIST trained model weights for a completely different input ie. Geolife might not guarantee the most favorable results - since MNIST was trained to look for patterns in handwritten digits and not in roadmaps.
5. By analyzing the cluster output data, it is possible to map the heatmaps back to the real world maps to understand a trajectory, users, and locations.
    a. Understanding trajectory - Relating spatial and temporal aspects
    b. Understand users - To find the travel path of users based on their past
    c. Understand location - Understand the popularity, and crowd gathering at certain locations and at certain times.
6. The IDEC implementation was performed for 10 clusters and hence 10 very unique behavioral patterns were observed within each cluster.
7. The SSIM mean value for the IDEC decoded layer output was 0.45876 whereas the paper by Ferreira et al [2] quoted the SSIM for Geolife - autoencoder as 0.2266. A higher SSIM means that there is more similarity between the users in the clusters

Improvements:
1. Autoencoders are not that efficient when compared to Generative Adversarial Networks in reconstructing an image. Replacing the autoencoder with a GAN model + clustering could be an area to look into.

ClusterGAN Architecture as proposed by Sudipto Mukherjee et al from the University of Washington [1].

2. Training an autoencoder from scratch for large image sizes as mentioned in the observations

3. Certain user data also contains a label.txt file in ‘Data/010/labels.txt’ that shows information such as start time, end time, and transportation mode. Relating the mode of transport with the trajectory could yield some useful information.

4. Since the experiment was performed for each user at a different month, it did not account for granular details like seasonality and time, it gets harder to interpret the information. Hence, the implementation could also be performed at a granular level.

References:

[1] Mukherjee, S., Asnani, H., Lin, E., Kannan, S.: ClusterGAN: Latent space clustering in generative adversarial networks. arXiv:1809.03627 (2019)

[2] Danielle L. Ferreira, Bruno A. A. Nunes, Carlos Alberto V. Campos, and Katia Obraczka. 2020. A Deep Learning Approach for Identifying User Communities Based on Geographical Preferences and Its Applications to Urban and Environmental Planning. ACM Trans. Spatial Algorithms Syst. 1, 1 (August 2020), 24 pages.