Experimenting with Machine Learning on Graphs

As with spatial networks, if your computer has limited amount of memories or just if you want to save time when experimenting, you can create a subgraph of the airport dataset, for instance only with the most important nodes, or only nodes in a region of the world.

1. Training and Validation set
   (a) Let's start by creating a training set for the Airport dataset. A training set is composed of t edges (taken at random) and t pairs of nodes without edges (taken at random). You can use `np.random.choice` for selecting edges. For non-edges, handling a list of all pairs of nodes is rather inefficient, so it can be more efficient to sample random pairs of nodes and to keep those without edges. Select a balanced set. Typically, you can take $t = L/6$
   (b) Do not forget to remove edges from in the training set from the graph.
   (c) Do the same to create a validation set. To simplify, make it balanced too (we will validate with AUROC score).
2. Computing heuristics
   (a) Using existing functions in `networkx`,(`adamic_adar_index`, etc.) compute common heuristics between all (or a sample of) pairs of nodes on the graph.
   (b) Find the 20 node pairs of higher and lower scores, for each heuristic. Are these rankings intuitively a good starting point?
3. Evaluating heuristics efficiency
   (a) We will use function `roc_auc_score` from package `scikit-learn`. It takes as input two lists, representing for each pair of nodes in our **validation set**, 1)the true label (1 for an edge, 0 for non-edges), and 2)A score for this pair of nodes. Create those lists (ensure that the order of node pairs is the same in all lists.)
   (b) Evaluate the quality of each heuristic according to the AUROC score. (Note that we have not used the training set at this point.)
4. Using Machine Learning
   (a) We will use the `sklearn.linear_model.LogisticRegression` function to train our model. To train the model, we will use the following method: `clf = LogisticRegression().fit(X, y)`, as in the example of the documentation. We need to prepare X and y. X represents the *input* and can be provided as a list of list: each of the internal list corresponds to the features of one node pair. y is the list of values to predict. For instance, `X=[[1,3,1],[2,20,10]]`, `y=[1,0]` corresponds to a training set with 2 examples, each having 3 features, the first one being an edge and the second one a non-edge. Prepare X (combining all heuristics) and y from the training set.
   (b) Train the model.
   (c) Use function `LogisticRegression.predict_proba(Xvalidate)` to get a score corresponding to each node pair in the validation set.
   (d) Compute the AUROC score of this new prediction. Compare with using each of the heuristic alone
5. Going further: Spatial Model
   (a) In a previous class, we have built a spatial model of this network. Do you see how we could use such a model for link prediction? Compare results obtained with such a model with those obtained using Machine Learning.

(b) Can we combine the prediction made by heuristics and by the spatial model to improve the prediction?

6. Going further: Node attributes prediction

(a) Hide the country information of 20% of airports. We could imagine that this information was missing in the database. Propose a ML based method to assign a country to those airports and check the results.

(b) Hide the location information of 20% of airports. Propose a method to rediscover their location.