Selected Topics in Machine Learning Optimization, Computer Vision and Networks: Lab - Machine Learning with Graphs

Arlei Silva

June, 2022

We will use Google Colab for this activity[1]. Each activity has the URL to a Colab notebook with instructions and hints. Copy the Colab and edit it.

You can choose the activity that interests you the most or work on multiple activities if desired.

These is a list of key libraries used:

- networkx: `https://networkx.org`
- numpy/scipy: `https://numpy.org`
- matplotlib: `https://matplotlib.org`
- pytorch: `https://pytorch.org`
- pytorch-geometric: `https://pytorch-geometric.readthedocs.io/en/latest/`

1) Network Science

The goal is to apply network science metrics to characterize real graphs. You can access the google colab project through this link: `shorturl.at/auJS4`. Run the available code to load the libraries and dataset[2]. The tasks are the following:

a) Compute the number of nodes in the network.

b) Compute the number of edges in the network.

c) Plot the degree distribution of the network.

[1] `https://colab.research.google.com`

[2] `https://paperswithcode.com/dataset/cora`

d) Compute the average shortest path for the largest connected component in the network.

e) Compute the diameter for the largest connected component in the network.

Extra) Repeat the steps above for a network of your choice. You can use a dataset provided by networkx.

2) Spectral Graph Theory

The goal is to apply spectral graph theory for graph clustering. You can access the google colab project through this link: `shorturl.at/cpyF9`. Run the available code to load the libraries and dataset[3]. The tasks are the following:

a) Compute the number of nodes in the network.

b) Compute the number of edges in the network.

c) Draw the network using the attribute 'value' to color the nodes (red for 0 and blue for 1).

d) Compute the eigenvalues and eigenvectors of the Laplacian of the network.

e) Draw the second smallest eigenvector of the Laplacian. Does it reveal the community structure?

f) Apply spectral clustering to the network with 2 clusters. Draw the cluster assignments. Is the result good?

g) Apply spectral clustering to the network with 3 clusters. Draw the cluster assignments. Is the result better?

Extra) Evaluate the quality of the cluster assignments compared to the ground-truth assignments.

3) Graph Neural Networks

The goal is to apply GNNs for link prediction. Edge probabilities will be computed using inner products of embeddings:

$$p(u, v) = \langle \mathbf{z}_u, \mathbf{z}_v \rangle$$

[3] `http://www-personal.umich.edu/~mejn/netdata/`

You can access the google colab project through this link: `shorturl.at/eDKT9`. Run the available code to load the libraries and dataset[4]. The GCN example discussed during the lecture is given as an example. The tasks are the following:

a) Change the training algorithm to link prediction using the same GCN as the encoder (encoder=None).

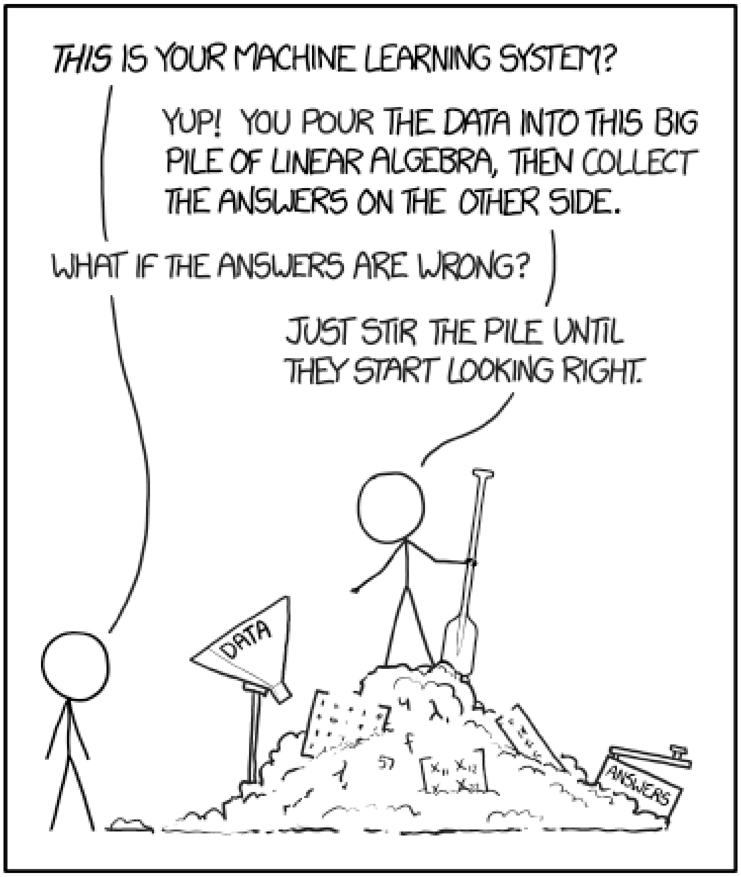
b) Reduce the dimensionality of the embedding to 2.

c) Visualize the embeddings with node labels. Are the embeddings better than earlier?

Extra 1) Change training algorithm to use mini-batches.

Extra 2) Evaluate the quality of the link prediction of your solution.

[4] `https://paperswithcode.com/dataset/cora`


THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

The pile gets soaked with data and starts to get mushy over time, so it's technically recurrent. Source: `https://xkcd.com/1838/`