# Assignment 2

CS 512: Data Mining Principles (Fall 2022)
Instructor: Hanghang Tong

Release date: Oct. 6th, 2022
Due date: Nov. 15th, 2022

- This assignment will cover the content from slides #4 (Deep Learning), #5 (Outlier Detection), and #6 (Network and Graph Connectivity).
- Feel free to discuss with other members of the class when doing the homework. You should, however, write down your own solution **independently**. ***Very Important Notes*: (1) there is a fine line between collaboration and completing the assignment by yourself and (2) aiding others to cheat would have the same consequence as the cheating itself. Please try to keep the solution brief and clear.**
- Please use Piazza first if you have questions about the homework. Also feel free to send us e-mails and come to office hours.
- The homework is due at 11:59 PM on the due date. We will be using Canvas for collecting assignments. *Please do not hand in any handwritten solution (including scanned solutions on papers or handwritten solutions on tablets), only the typed solution (e.g., Microsoft Word, Latex, etc) will be graded.* The datasets and starting codes for HW2 are in **HW2_source.zip** on Canvas. Contact the TAs if you are having technical difficulties in submitting the assignment. We do **NOT** accept late homework!
- The solution report should be submitted as a **single** pdf file using the name convention `yourNetid_HW2.pdf`. If you use additional source code (Python is recommended) for solving problems, **you are required to submit them** and use the file names to identify the corresponding questions. For instance, '`yourNetid_HW2_problem1.py`' refers to the python source code for Problem 2 for HW 2. Compress all the files (pdf and source code files) into one zip file. Submit the compressed file ONLY.
- For each question, you will NOT get full credits if you only give out a final result. Necessary calculation steps are required. If the result is not an integer, round your result to 2 decimal places.

**Problem 1. Short Questions.** (8 points) Enough justification is needed for every question. If it is a 'True or False' question, you need to clearly state 'True or False' before your justification.

1. (**True or False**) (1 point) The data that cannot be reconstructed by succinct representations is more likely to be an outlier.
2. (**True or False**) (1 point) The diameter of a fully connected graph is 1.
3. (**Short Answer**) (1 point) What is the core idea of proximity-based outlier detection methods?
4. (**Short Answer**) (1 point) Name three superiorities of CNN compared with MLP.
5. (**Short Answer**) (1 point) What is the difference between translational equivalence and translational invariance.
6. (**Short Answer**) (1 point) Explain why dropout can help prevent overfitting.
7. (**Short Answer**) (2 points) Compute the clustering coefficient (Network average clustering coefficient) of the network given in Figure 1. The definition of Network average clustering coefficient can be found at `https://en.wikipedia.org/wiki/Clustering_coefficient`

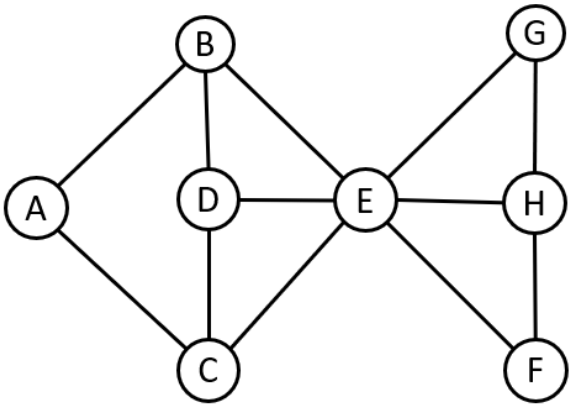


Figure 1: Network

**Problem 2. Distance-Based Outlier Detection** (8 points) A sample $o$ is defined as a $DB(r, \pi)$-outlier if

$$\frac{||o'|dist(o, o') \leq r||}{||D||} \leq \pi, \tag{1}$$

where $||\cdot||$ denotes the cardinality, $D$ is the whole dataset, $r$ is the distance threshold, and $\pi$ is the fraction threshold.

(a) (2 points) Explain with your own words why the above formula can be used to measure the distance-based outliers.

(b) (2 points) Explain with your own words why the above criteria can be rewritten as: if $dist(o, o_k) > r$ then $o$ is an outlier, where $o_k$ is the k-nearest neighbor of $o$ and $k = \lceil \pi ||D|| \rceil$.

(c) (4 points) Given a 1-D dataset as: $\{-4.5, -4, -3, -2.5, 0, 3, 3.5, 4, 4.5, 5\}$, report which nodes are the $DB(2, 0.2)$-outlier and show all the intermediate results.. Here we use the $|\cdot|$ as the distance function.

**Problem 3. Density-Based Outlier Detection** (10 points) Carefully read contents from textbook and lecture notes about "Density-Based Outlier Detection" and answer the following questions.

(a) (2 points) What is the main differences between distance-based outlier detection and density-based outlier detection? Explain with your own words.

(b) (4 points) Given a data sample $o$ from the dataset $D$, $dist_k(o)$ denotes the distance between $o$ and its $k$-nearest neighbor. Then

$$N_k(o) = \{o' | o' \in D, dist(o, o') \leq dist_k(o)\} \tag{2}$$

denotes the $k$-instance of $o$. A smoothed distance measure from $o'$ to $o$ is reachability distance as

$$reachdist_k(o \leftarrow o') = \max\{dist_k(o), dist(o, o')\}. \tag{3}$$

Thus, the **local reachability density** is defined as

$$lrd_k(o) = \frac{||N_k(o)||}{\sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)} \tag{4}$$

Explain the meaning of Eq. 4 with your own words. Try to explain from (i) the meaning of $\frac{\sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)}{||N_k(o)||}$ and (ii) if the sample $o$ is closer to a cluster, will $lrd_k(o)$ be larger or smaller?

(c) (4 points) The **local outlier factor** is defined as

$$LOF_k(o) = \frac{\sum_{o' \in N_k(o)} \frac{lrd_k(o')}{lrd_k(o)}}{||N_k(o)||}. \tag{5}$$

Explain with your own words about (i) for the numerator, why we use $\sum_{o' \in N_k(o)} \frac{lrd_k(o')}{lrd_k(o)}$ but not $\sum_{o' \in N_k(o)} lrd_k(o')$ and (ii) why we need $||N_k(o)||$ as the denominator; if we define $MY\_LOF_k(o) = \sum_{o' \in N_k(o)} \frac{lrd_k(o')}{lrd_k(o)}$, do you think $MY\_LOF_k(i)$ and $MY\_LOF_k(j)$ are comparable?

**Problem 4. Training Neural Networks** (10 points)

(a) (2 points) Identify challenges (a-d) when optimizing a non-convex objective function $E(\theta)$ in Figure 2.

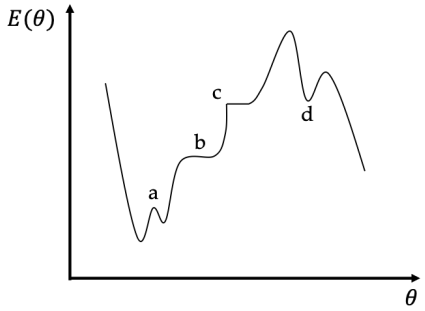


Figure 2: Challenges in optimizing non-convex objective functions

(b) (8 points) Consider a neural network in Figure 3, we denote the input of neuron $i$ as $I_i$, corresponding output as $O_i$, and the weight for link between neuron $i$ and $j$ as $w_{i,j}$. With mean-square error loss $L = \frac{1}{2}(T - O_k)^2$ and sigmoid activation function ($O_k = \frac{1}{1+e^{-I_k}}$).

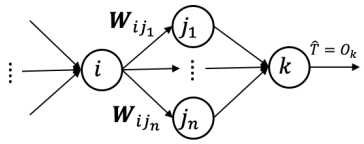


Figure 3: Neural network architecture

1. (2 points) Prove that the error for the output neuron $k$ is $\delta_k = O_k(1 - O_k)(O_k - T)$.
2. (2 points) Prove that the error for the hidden neuron $i$ is $\delta_i = O_i(1 - O_i)\sum_j w_{ij}\delta_j$.
3. (2 points) Now change the loss function to the cross-entropy loss $L = -T \log O - (1 - T) \log(1 - O)$, compute the error for the output neuron $k$.
4. (2 points) Based on the previous questions, briefly explain why cross-entropy loss can avoid the gradient vanishing problem when the output unit is saturated faced by the sigmoid function.

**Problem 5. Neural Network Models** (10 points)

(a) (2 points) Consider the 2D convolution in Figure 4, compute the feature maps of the input by applying two kernels $K_1, K_2$ respectively (stride=1). (Show your results in python list format, e.g., $K_1$ can be represented as $[[1, 0], [0, 1]]$.)

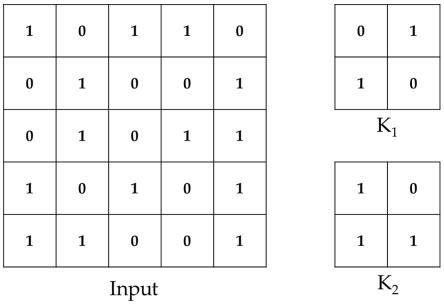


Figure 4: Input data and kernels for 2D convolution

(b) (2 points) Compute the feature map after applying a $2 \times 2$ average pooling layer with stride=2 to the feature maps from (c).

(c) (1 point) Given an input feature with shape $N \times N \times D$ and $L$ kernels with shape $K \times K \times D$, compute the shape of the output feature after convolution with stride=$S$. (Floor your answer in case of the boundary issue.)

(d) (3 points) Consider the RNN in Figure 5, given input data $h_0, x_1, x_2$ and network parameters $\mathbf{W}, \mathbf{U}, \mathbf{V}$, without non-linear activation function and bias, compute $h_1, h_2, \hat{y}_2$.

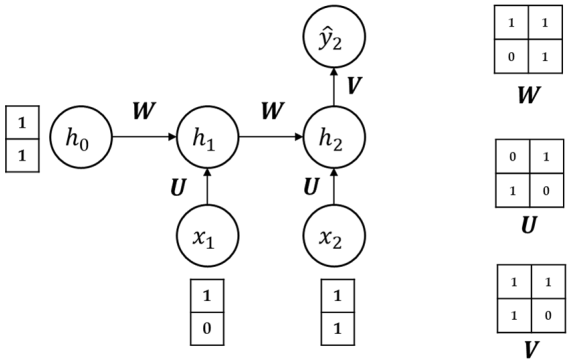


Figure 5: Input data and parameters for RNN

(e) (2 points) Consider the graph in Figure 6, if we apply a two-layer GCN on this graph, draw the message passing tree for node a.

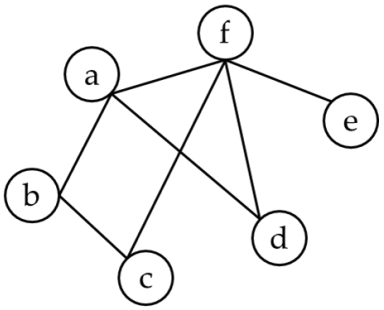


Figure 6: Input graph for GCN

**Problem 6. Implementations of Outlier Detection Algorithms** (10 points) For this problem, the ALOI dataset (in the source file) is used with following steps.

(a) (2 points) Load the dataset, report the number of samples and features.

(b) (4 points) Implement a proximity-based model LOF on the given dataset, report the micro F1 score of the outlier detection performance.

(c) (4 points) Implement a reconstruction-based model autoencoder on the given dataset, report the micro F1 score.

You should only use python standard library, Numpy, scikit-learn, and pyod[1] in your implementation. Your source code is required to submit. Please ensure it is bug-free and print the results clearly.

[1] https://github.com/yzhao062/pyod

**Problem 7. Image Classification with CNN** (12 points) Implement a CNN image classifier on CIFAR dataset using PyTorch. Before implementation, make sure you go through the PyTorch tutorial on CIFAR dataset (`https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html`).

(a) (1 point) Load CIFAR dataset, report number of classes, number of training images, number of test images, and image shape.

(b) (1 point) Report your training loss along the training process, i.e., a figure with the index of training batch as the x-axis and the value of loss function as the y-axis.

(c) (10 points) Submit your model together with the training code. The final score will be given based on the classification accuracy on the test dataset as shown in Figure 7.

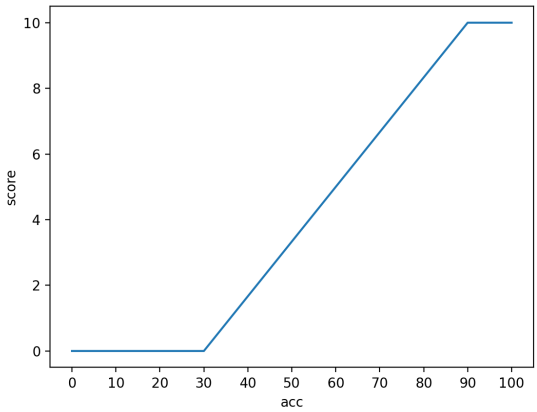


Figure 7: Grading function

You are required to use PyTorch for your implementation. Your source code and trained model are required to submit. Please ensure it is bug-free and print the results clearly.

**Problem 8. Network Connectivity** (15 points)

Reading paper "Node Immunization on Large Graphs: Theory and Algorithms" and answering the following questions.

(a) (3 points) Please explain why the first eigenvalue of the adjacency matrix is a good "Vulnerability" measure of a graph?

(b) (3 points) Please describe the disadvantages of using the first eigenvalue of the adjacency matrix as the "Vulnerability" measure of a graph.

(c) (3 points) Why is the proposed "Shield-value" score a good measure?

(d) (3 points) Briefly analyze the time complexity and space complexity of "NetShield" and explain why.

(e) (3 points) Briefly explain the difference between "NetShield" and "NetShield+". What are the advantages of "NetShield+" compared with "NetShield"?

**Problem 9.** (17 points) **SIS model** For an SIS model on a network, every node has two statuses: sick or healthy. $\beta$ denotes the infection rate that a sick node infects its neighbors in a time step. $\delta$ denotes the recovery rate of a sick node turning into a healthy node in a time step. Wang et al.[2] propose that there will be no epidemic if $\frac{\beta\lambda}{\delta} < 1$ where $\lambda$ is the largest eigenvalue of the adjacency matrix. Let us verify it on the given **social network** dataset (an edge list representing an unweighted and undirected network). From time step $t$ to time step $t+1$ (1) the virus propagates along the network (the infected nodes will keep infected) based on $\beta$ and then (2) all the infected nodes will try to recover based on $\delta$.

(a) (2 points) If a healthy node has $n$ sick neighbors, what is the probability of this healthy node get infected in the next time step?

(b) (2 points) What is the largest eigenvalue of the given network?

(c) (10 points) Assume that at the beginning, all the nodes are infected. If $\beta = 0.01$ and $\delta = 0.05$, draw a line chart whose x-axis denotes time steps, y-axis denotes the number of nodes. One line represents the number of healthy nodes and the other line represents the number of infected nodes. Show the results from 100 time steps. Does this result align with the theory mentioned in our problem?

(d) (3 points) What would the line chart be if $\beta = 0.01$ and $\delta = 0.40$? Does it align with the theory mentioned in our problem?

You should only use python standard library and Numpy. Your source code is required to submit. Please ensure it is bug-free and print the results clearly.

[2]Wang, Yang, et al. "Epidemic spreading in real networks: An eigenvalue viewpoint." 22nd International Symposium on Reliable Distributed Systems, 2003. Proceedings.. IEEE, 2003.