Assignment 1

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

Release date: Aug. 23rd, 2022
Due date: Oct. 6th, 2022

- This assignment will cover the content from slides #1 (Introduction), #2 (Frequent Pattern Mining), #3 (Classification), and #4 (Clustering).
- 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 HW1 are in **HW1_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_HW1.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_HW1_problem1.py`' refers to the python source code for Problem 1 for HW 1. 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**) FP growth is always faster than Apriori.
2. (**True or False**) For sequential pattern mining, both the sequence itself and the items within every event are ordered.
3. (**True or False**) The support vectors of soft margin SVM only contain points lie inside the margin bound.
4. (**True or False**) In most cases, clustering analysis is supervised learning.
5. (**True or False**) Zero-shot learning is also known as unsupervised learning.
6. (**True or False**) Suppose $\alpha = \{\alpha_0, \alpha_1, ..., \alpha_m\}$ is the Minimum DFS Code of of graph a, then add an edge on the rightmost path of a, the new DFS code $\beta = \{\alpha_0, \alpha_1, ..., \alpha_m, b\}$ is also the Minimum DFS Code.
7. (**Short Answer**) Why does SVM with Gaussian kernel can achieve zero training error for any dataset with two classes?
8. (**Short Answer**) High quality clusters from clustering methods should enjoy what properties?

Problem 2. Association Rule (8 points) The following contingency table summarizes supermarket transaction data, where *hot dogs* refers to the transactions containing hot dogs, $\overline{hot\ dogs}$ refers to the transactions that do not contain hot dogs, *hamburgers* refers to the transactions containing hamburgers, and $\overline{hamburgers}$ refers to the transactions that do not contain hamburgers.

	hot dogs	$\overline{hot\ dogs}$	$\sum_{row}$
hamburgers	2000	500	2500
$\overline{hamburgers}$	1000	1500	2500
$\sum_{col}$	3000	2000	5000

(a) **(Strong Association Rule)** (2 points) Given a minimum support threshold of 25% and a minimum confidence threshold of 50%, try to determine whether ”*hot dogs* $\Rightarrow$ *hamburgers*” is a strong association rule.

(b) **(Correlation Relationship)** (2 points) Based on the given data, is the purchase of *hot dogs* independent of the purchase of *hamburgers*? If not, what kind of correlation relationship exists between the two?

(c) **(Different Measures)** (4 points) List the value of following measures between *hot dogs* and *hamburgers*: *all_confidence*, *max_confidence*, *Kulczynski*, *consine*, and *lift*.

Problem 3. Rule Mining (12 points) A database has four transactions. Let $min\_sup = 60\%$ and $min\_conf = 80\%$

cust_ID	TID	items_bought (in the form of brand-item_category)
01	T100	{King's-Crab, Sunset-Milk, Dairyland-Cheese, Best-Bread }
02	T200	{Best-Cheese, Dairyland-Milk, Goldenfarm-Apple, Tasty-Pie, Wonder-Bread}
01	T300	{ Westcoast-Apple, Dairyland-Milk, Wonder-Bread, Tasty-Pie}
03	T400	{Wonder-Bread, Sunset-Milk, Dairyland-Cheese}

(a) (3 points) At the granularity of *item_category* (e.g., $item_i$ could be "Milk"), for the rule template,

$$\forall X \in transaction, buys(X, item_1) \wedge buys(X, item_2) \Rightarrow buys(X, item_3)[s, c] \qquad (1)$$

list the frequent k-itemset for the largest k, and all the strong association rules (with their support s and confidence c) containing the frequent k-itemset for the largest k.

(b) (3 points) At the granularity of $brand - item\_category$ (e.g., $item_i$ could be "Sunset-Milk"), for the rule template,

$$\forall X \in customer, buys(X, item_1) \wedge buys(X, item_2) \Rightarrow buys(X, item_3) \qquad (2)$$

list the frequent $k - itemset$ for the largest k (but do not report any rules).

(c) (6 points) Please download the dataset "purchase_hisotry.csv" on Canvas and implement the Apriori algorithm. Given a minimum support threshold 1264. Run you code, output all the frequent patterns and submit the your code. Put all of your code in yourNetid_HW1_problem3.py, and the code should be bug-free by running "python yourNetid_HW1_problem3.py"

Problem 4. (10 points) **GSP Algorithm.** Given the following sequence dataset. Adopt GSP algorithm with min support as 3 to show all the frequent sub-sequences. Note that you need to show detailed intermediate results including the candidate sequences of different length at each iteration.

Table 1: Sequence Dataset

Sid	**Sequence**
1	$< (ad)c(ac) >$
2	$< bd >$
3	$< (ab)ab >$
4	$< ad(ab) >$
5	$< a(bd)dd(ac) >$

Problem 5. Frequent Graph Mining (10 points) Read the chapter 5 in the textbook and the paper "gspan: Graph-based substructure pattern mining".

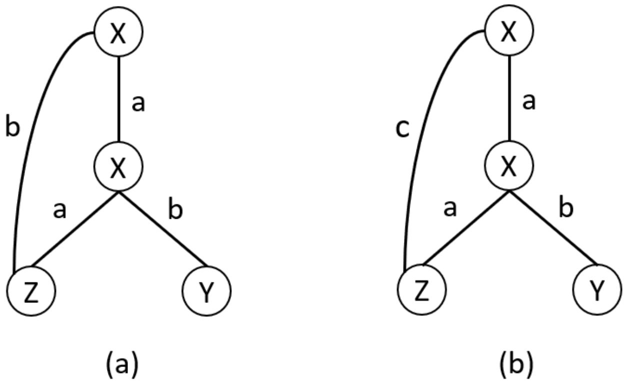


Figure 1: GSpan

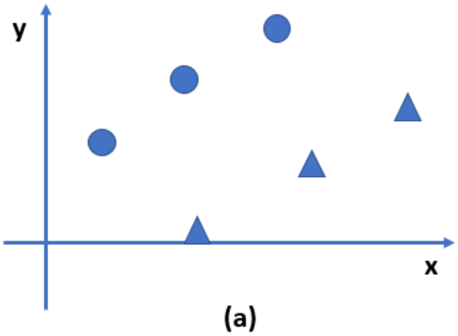
Given a graph data set D which contains the two graphs in Figure 1, and the support threshold minsup = 2, find all frequent subgraphs. Please draw the DFS Code Tree of the whole search process. You can denote each node in the tree by the corresponding subgraph or the DFS code of this subgraph. Please also draw the empty root node in your answer.

Problem 6. SVM (12 points)

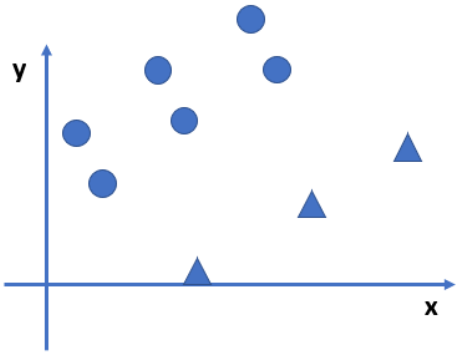
(a) (2 points) when ignoring regularization, given the following 6 data points in Figure (a), please draw the decision boundary of linear SVM and logistic regression.

If we add three more nodes, and get Figure (b), please draw the decision boundary of linear SVM and logistic regression.

Comparing the decision boundary you draw in (a) and (b), briefly analyze the advantages of linear SVM compared with linear regression.



(a)



(b)

(b) (4 points) Write down the loss function of soft-margin linear SVM and logistic regression with L2 normalization in min $L() + regularization$ or min max $L() + regularization$ form. Briefly discussing the commonality and difference between soft-margin linear SVM and logistic regression with L2 normalization

(c) (6 points) If $k_1(x_i, x_j)$ and $k_2(x_i, x_j)$ are kernel functions, prove that the following is also kernel functions such that they can be formulated as inner products.

$c_1 k_1(x_i, x_j) + c_2 k_2(x_i, x_j)$

$k_1(x_i, x_j) k_2(x_i, x_j)$

$f(x_i) k_1(x_i, x_j) f(x_j)$

Problem 7. Classifier Implementation. (10 points)

Given the dataset 'car_data.csv', the "Age" and "AnnualSalary" columns are selected as features and the "Purchased" column denotes labels. We set the first 80% rows as the training set and the rest rows as the test set. You need to:

- normalize every feature column by the maximum of the corresponding feature,
- implement a logistic regression, a linear SVM and a RBF kernel SVM (with $\gamma = 1$) and train them on the training set,
- report the accuracy of above classifiers on test set,
- try to remove the feature normalization process, set the maximum number of training iteration as 500, and re-report the accuracy of above classifiers on test set.

All you implementations should only use python standard library, Numpy, and scikit-learn.

Problem 8. Random Walk with Restart (9 points).

(a) (3 points) Given a binary adjacency matrix $\mathbf{A} \in \{0,1\}^n$ for a graph, $\mathbf{A}[i,j]$ denotes the node i and node j are connected (i.e., $\mathbf{A}[i,j] = 1$) or not. Why $\mathbf{A}^K[i,j]$ denotes the number of K-hop paths between the node i and node j? Here a path is a sequence of edges which joins a sequence of nodes. For example, $i \to j \to i \to j$ is a 3-hop path. Justify your answer mathematically.

(b) (6 points) Let us explore the difference between random walk and random walk with restart with following steps.

- (Provided in the starting code) Load the given **email-EU-core** graph and select its largest component. Note that this dataset is a directed graph but we view it as undirected for convenience.
- Symmetrically normalize its adjacency matrix as $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{D}$ is a diagonal matrix s.t. $\mathbf{D}[i,i] = \sum_j \mathbf{A}[i,j]$.
- Run random walk algorithm as

$$\mathbf{r}^{(t+1)} = \tilde{\mathbf{A}}\mathbf{r}^{(t)}.$$

  Try different one-hot initialization (e.g., $[1, 0, \ldots, 0]$) of the ranking vector $\mathbf{r}^{(0)}$ and report what you observed from the converged results.
- Run random walk with restart algorithm as

$$\mathbf{r}^{(t+1)} = c\tilde{\mathbf{A}}\mathbf{r}^{(t)} + (1-c)\mathbf{e}.$$

  In this homework, we select the damping factor c as 0.9 and set the restart vector $\mathbf{e}$ as $\mathbf{r}^{(0)}$. Try different one-hot initialization (e.g., $[1, 0, \ldots, 0]$) of the ranking vector $\mathbf{r}^{(0)}$, report what you observed from the converged results, and compare them with the converged results from the last step.

Please write your code under the provided starting code. Except the loading dataset part, all you implementations should only use python standard library and Numpy.

Problem 9. Gaussian Mixture Model (12 points). We explore the property of Gaussian mixture model (GMM) by a 1-D case.

(a) (3 points) For a 1-D GMM model, the probability of a data point x_i belongs to the cluster j is computed as:

$$w_{ij}^{t+1} = \frac{w_j^t P(x_i|\mu_j^t, \sigma_j^t)}{\sum_k w_k^t P(x_i|\mu_k^t, \sigma_k^t)} \tag{3}$$

where μ_j and σ_j are the mean and standard deviation of the cluster j. Explain

- what is w_j^t?
- what is $P(x_i|\mu_j^t, \sigma_j^t)$ and how to compute it?
- why we need the denominator $\sum_k w_k^t P(x_i|\mu_k^t, \sigma_k^t)$?

(b) (6 points) Try to implement GMM models with following steps.

- (Provided in the starting code) generate a synthetic 1-D dataset with 4000 samples from two Gaussian distributions. Initialize the parameters of the GMM model (with 2 clusters).
- Study lecture notes and apply EM algorithm to update the parameters of GMM model. Try to update in 10 iterations and report $\{w_j^t\}$, $\{\mu_j^t\}$, $\{\sigma_j^t\}$ from every iteration. Explain that whether GMM model works for our synthetic data.

All you implementations should only use python standard library and Numpy.

(c) (3 points) Assume that for both two clusters, their standard deviation $\sigma_1^t = \sigma_2^t = \epsilon$. Assume that $w_1^t \neq 0$, $w_2^t \neq 0$, and $P(x_i|\mu_1^t, \sigma_1^t) \neq P(x_i|\mu_2^t, \sigma_2^t)$, $\forall i$. Mathematically show that why GMM compute hard cluster assignments (like KMeans) when $\epsilon \to 0$.

Problem 10. 2-way Spectral Graph Partitioning (9 points). From lecture notes, given an adjacency matrix $\mathbf{A} \in \{0, 1\}^n$, the relaxed 2-way spectral graph partitioning method can be presented as

$$\begin{aligned} \mathbf{q}^* &= \arg\min_{\mathbf{q}} \mathbf{q}^T(\mathbf{D} - \mathbf{A})\mathbf{q} \\ s.t. &\sum_k \mathbf{q}[k]^2 = n. \end{aligned} \tag{4}$$

(a) (3 points) $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is named as graph Laplacian matrix whose minimum eigenvalue is 0. What is the eigenvector corresponding to the eigenvalue 0? Can we use this vector to conduct 2-way partitioning on the given graph? Provide your answer with mathematical justification.

(b) (6 points) Implement 2-way Spectral Graph Partitioning on the **email-EU-core** graph in following steps.

- (Provided in the starting code) Load the given **email-EU-core** graph and select its largest component. Note that we still view it as an undirected graph.
- Eigen decompose the graph Laplacian matrix and obtained the eigenvector corresponding to the second minimum eigenvalue.
- Partition the given graph into two subgraphs according to the eigenvector from the last step. What is the size of the partitioned subgraphs? Report the number of cut between two subgraphs. Note that for undirected case 1 edge should be counted twice.

Please write your code under the provided starting code. Except the loading dataset part, all you implementations should only use python standard library and Numpy.