HOMEWORK 6

Arnav Sharma
9074042756

Instructions: You can choose any programming language as long as you implement the algorithm from scratch. Use this latex file as a template to develop your homework. Submit your homework on time as a single pdf file to Canvas. Please check Piazza for updates about the homework.

1 Kernel SVM [30pts]

Consider the following kernel function defined over $\mathbf{z}, \mathbf{z}' \in Z$, where Z is some set:

$$\kappa(\mathbf{z}, \mathbf{z}') = \begin{cases} 1 & \text{if } \mathbf{z} = \mathbf{z}', \\ 0 & \text{otherwise.} \end{cases}$$

1. (10 pts) Prove that for any positive integer m, any $\mathbf{z}_1, \dots, \mathbf{z}_m \in Z$, the $m \times m$ kernel matrix $\mathbf{K} = [\mathbf{K}_{ij}]$ is positive semi-definite, where $\mathbf{K}_{ij} = \kappa(\mathbf{z}_i, \mathbf{z}_j)$ for $i, j = 1 \dots m$. (Let us assume that for $i \neq j$, we have $\mathbf{z}_i \neq \mathbf{z}_j$) Hint: An $m \times m$ matrix K is positive semi-definite if $\forall \mathbf{u} \in \mathbb{R}^d : \mathbf{u}^\top \mathbf{K} \mathbf{u} \geq 0$.
We're Given:

$$\mathbf{K} = [\mathbf{K}_{ij}] \quad \& \quad \mathbf{z}_i \neq \mathbf{z}_j \ \text{ if } \ \mathbf{i} \neq \mathbf{j}$$
$$\implies \mathbf{K} = [\kappa(\mathbf{z}_i, \mathbf{z}_j)] \quad \& \quad \mathbf{z}_i \neq \mathbf{z}_j \ \text{ if } \ \mathbf{i} \neq \mathbf{j}$$
$$\implies \mathbf{K}_{ij} = \begin{cases} 1 & \text{if } \mathbf{i} = \mathbf{j}, \\ 0 & \text{otherwise.} \end{cases}$$

Since $\mathbf{K}$ is an $m \times m$ Matrix,

$$\implies \mathbf{K} = \mathbf{I_{m \times m}}$$

To Prove:

$$\mathbf{x}^\top \mathbf{K} \mathbf{x} \geq 0 \ \forall \mathbf{x} \in \mathbb{R}^m$$
$$\implies x^\top \mathbf{K} \mathbf{x} = \mathbf{x}^\top I_{m \times m} \mathbf{x}$$
$$\implies x^\top \mathbf{K} \mathbf{x} = \begin{bmatrix} \sum_{l=1}^m \mathbf{x}_l I_{l,1} & \sum_{l=1}^m \mathbf{x}_l I_{l,2} & \dots & \sum_{l=1}^m \mathbf{x}_l I_{l,m} \end{bmatrix} \mathbf{x}$$

Since $I_{i,j} = 1$ if i=j, else $I_{i,j} = 0$

$$\forall i \in [m], \sum_{l=1}^{m} \mathbf{x}_l I_{l,i} = 0 + 0 + \dots + \mathbf{x_i} + \dots + 0 = \mathbf{x_i}$$
$$\implies x^\top \mathbf{K} \mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_m \end{bmatrix} \mathbf{x}$$
$$\implies x^\top \mathbf{K} \mathbf{x} = \mathbf{x}^\top \mathbf{x}$$

Since $\mathbf{x}^\top \mathbf{x} = \|x\|^2$ which is $\geq 0 \forall x \in R^m$

$$\implies x^\top \mathbf{K} \mathbf{x} \geq 0 \ \forall \mathbf{x} \in \mathbb{R}^m$$

Therefore, proved that K is a positive semi-definite matrix

2. (10 pts) Given a training set $(\mathbf{z}_1, y_1), \ldots, (\mathbf{z}_n, y_n)$ with binary labels, the dual SVM problem with the above kernel κ will have parameters $a_1, \ldots, a_n, b \in \mathbb{R}$. (Let us assume that for $i \neq j$, we have $\mathbf{z}_i \neq \mathbf{z}_j$) The predictor for input $\mathbf{z}$ takes the form

$$f(\mathbf{z}) = \sum_{i=1}^{n} a_i y_i \kappa(\mathbf{z}_i, z) + b \ .$$

Recall that the label prediction is $\text{sgn}(f(\mathbf{z}))$. Prove that there exist $a_1, \ldots, a_n, b$ such that f correctly separates the training set. In other words, κ induces a feature space rich enough such that in it, any training set can be linearly separated.

3. (10 pts) How does that f predict input $\mathbf{z}$ that is not in the training set?

Comment: One useful property of kernel functions is that the input space Z does not need to be a vector space; in other words, $\mathbf{z}$ does not need to be a feature vector. For all we know, Z can be all the turkeys in the world. As long as we can compute $\kappa(\mathbf{z}, \mathbf{z}')$, kernel SVM works on turkeys.

2 Chow-Liu Algorithm [30 pts]

Suppose we wish to construct a directed graphical model for 3 features X, Y, and Z using the Chow-Liu algorithm. We are given data from 100 independent experiments where each feature is binary and takes value T or F. Below is a table summarizing the observations of the experiment:

X	Y	Z	Count
T	T	T	36
T	T	F	4
T	F	T	2
T	F	F	8
F	T	T	9
F	T	F	1
F	F	T	8
F	F	F	32

1. Compute the mutual information $I(X, Y)$ based on the frequencies observed in the data. (5 pts)

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} P_{X,Y}[x,y] \; log(\frac{P_{X,Y}[x,y]}{P_X[x] P_Y[y]})$$

$$I(X,Y) =$$

$$P_{X,Y}[x=F,y=F] \; log(\frac{P_{X,Y}[x=F,y=F]}{P_X[x=F]P_Y[y=F]}) + P_{X,Y}[x=F,y=T] \; log(\frac{P_{X,Y}[x=F,y=T]}{P_X[x=F]P_Y[y=T]})$$

$$+P_{X,Y}[x=T,y=F] \; log(\frac{P_{X,Y}[x=T,y=F]}{P_X[x=T]P_Y[y=F]}) + P_{X,Y}[x=T,y=T] \; log(\frac{P_{X,Y}[x=T,y=T]}{P_X[x=T]P_Y[y=T]})$$

$$= \frac{40}{100} \; log(\frac{\frac{40}{100}}{\frac{50}{100}\frac{50}{100}}) + \frac{10}{100} \; log(\frac{\frac{10}{100}}{\frac{50}{100}\frac{50}{100}})$$

$$+\frac{10}{100} \; log(\frac{\frac{10}{100}}{\frac{50}{100}\frac{50}{100}}) + \frac{40}{100} \; log(\frac{\frac{40}{100}}{\frac{50}{100}\frac{50}{100}})$$

$$= \frac{80}{100} \; log(\frac{\frac{40}{100}}{\frac{50}{100}\frac{50}{100}}) + \frac{20}{100} \; log(\frac{\frac{10}{100}}{\frac{50}{100}\frac{50}{100}})$$

$$= \frac{1}{5}(4log(\frac{\frac{40}{100}}{\frac{50}{100}\frac{50}{100}}) + log(\frac{\frac{10}{100}}{\frac{50}{100}\frac{50}{100}}))$$

$$= \frac{1}{5}(0.81648 - 0.39794)$$

$$= \frac{0.41854}{5}$$
$$= 0.08371$$

2. Compute the mutual information $I(X, Z)$ based on the frequencies observed in the data. (5 pts)

$$I(X,Z) = \sum_{x \in X} \sum_{z \in Z} P_{X,Z}[x,z]\ log(\frac{P_{X,Z}[x,z]}{P_X[x]P_Z[z]})$$
$$I(X,Z) =$$
$$P_{X,Z}[x=F,z=F]\ log(\frac{P_{X,Z}[x=F,z=F]}{P_X[x=F]P_Z[z=F]}) + P_{X,Z}[x=F,z=T]\ log(\frac{P_{X,Z}[x=F,y=T]}{P_X[x=F]P_Z[z=T]})$$
$$+P_{X,Z}[x=T,z=F]\ log(\frac{P_{X,Z}[x=T,z=F]}{P_X[x=T]P_Z[z=F]}) + P_{X,Z}[x=T,z=T]\ log(\frac{P_{X,Z}[x=T,z=T]}{P_X[x=T]P_Z[z=T]})$$
$$= \frac{33}{100}\ log(\frac{\frac{33}{100}}{\frac{50}{100}\frac{45}{100}}) + \frac{17}{100}\ log(\frac{\frac{17}{100}}{\frac{50}{100}\frac{55}{100}})$$
$$+\frac{12}{100}\ log(\frac{\frac{12}{100}}{\frac{50}{100}\frac{45}{100}}) + \frac{38}{100}\ log(\frac{\frac{38}{100}}{\frac{50}{100}\frac{55}{100}})$$
$$= 0.33log(\frac{22}{15}) + 0.17log(\frac{34}{55}) + 0.12log(\frac{8}{15}) + 0.38log(\frac{76}{55})$$
$$= 0.05489 - 0.03551 - 0.03276 + 0.05337$$
$$= 0.03999$$

3. Compute the mutual information $I(Z, Y)$ based on the frequencies observed in the data. (5 pts)

$$I(Z,Y) = \sum_{z \in Z} \sum_{y \in Y} P_{Z,Y}[z,y]\ log(\frac{P_{Z,Y}[z,y]}{P_Z[z]P_Y[y]})$$
$$I(Z,Y) =$$
$$P_{Z,Y}[z=F,y=F]\ log(\frac{P_{Z,Y}[z=F,y=F]}{P_Z[z=F]P_Y[y=F]}) + P_{Z,Y}[z=F,y=T]\ log(\frac{P_{Z,Y}[z=F,y=T]}{P_Z[z=F]P_Y[y=T]})$$
$$+P_{Z,Y}[z=T,y=F]\ log(\frac{P_{Z,Y}[z=T,y=F]}{P_Z[z=T]P_Y[y=F]}) + P_{Z,Y}[z=T,y=T]\ log(\frac{P_{Z,Y}[z=T,y=T]}{P_Z[z=T]P_Y[y=T]})$$
$$= \frac{40}{100}\ log(\frac{\frac{40}{100}}{\frac{45}{100}\frac{50}{100}}) + \frac{5}{100}\ log(\frac{\frac{5}{100}}{\frac{45}{100}\frac{50}{100}})$$
$$+\frac{10}{100}\ log(\frac{\frac{10}{100}}{\frac{55}{100}\frac{50}{100}}) + \frac{45}{100}\ log(\frac{\frac{45}{100}}{\frac{55}{100}\frac{50}{100}})$$
$$= 0.4log(\frac{16}{9}) + 0.05log(\frac{2}{9}) + 0.1log(\frac{4}{11}) + 0.45log(\frac{18}{11})$$
$$= 0.09995 - 0.03266 - 0.04393 + 0.09625$$
$$= 0.11960$$

4. Which undirected edges will be selected by the Chow-Liu algorithm as the maximum spanning tree? (5 pts)
The undirected edges between (Z, Y) and the between (X, Y) will be the ones slecedted as the maximum spanning tree.

5. Root your tree at node X, and assign directions to the selected edges. (10 pts)
Starting at node X, an edge can go from vertex X into vertex Y, and a second edge from vertex Y intwo vertex Z.

3 Game of Classifiers [60pts]

3.1 Implementation

Implement the following models in the choice of your programming language. Include slack variables in SVM implementation if needed. You can use autograd features of PyTorch, TensorFlow, etc., or derive gradients on your own (but do not use inbuilt models for SVM, Kernel SVM, and Logistic Regression from libraries).

- Implement Linear SVM (without kernels).
- Implement Kernel SVM, with options for linear, rbf, and polynomial kernels. You should keep the kernel parameters tunable (e.g., do not fix the degree of polynomial kernels but keep it as a variable and play with different values of it.) Is Linear SVM a special case of Kernel SVMs?
- Implement Logistic Regression with and without kernels (use same kernels as above).

3.2 Synthetic Dataset-1 (20 pts)

Generate a 2-D dataset as follows: Let $\mu = 2.5$ and $\mathbf{I}_2$ be the 2×2 identity matrix. Generate points for the positive and negative classes, respectively from $\mathcal{N}([\mu, 0], \mathbf{I}_2)$, and $\mathcal{N}([-\mu, 0], \mathbf{I}_2)$. For each class, generate 750 points (1500 in total). Randomly create train, validation, and test splits of 1000, 250, and 250 points, respectively. Do the following with this dataset:

1. (5 pts) Train your Linear SVM, Logistic Regression models and report decision boundaries and test accuracies.
2. (5 pts) Show the decision boundaries with k-NN and Naive Bayes Classifiers. (You can use library implementations or implement from scratch. Figure out the hyper-parameters using the validation set.)
3. (5 pts) Repeat the process by varying μ from 1.0 to 2.4 with a step size of 0.2 for each value of μ to obtain test accuracies of the models and plot (μ on x-axis and test accuracy on y-axis). (You will have a curve for each of the 4-classifiers mentioned above.)
4. (5 pts) What are your conclusions from this exercise?

3.3 Synthetic Dataset-2 (20 pts)

Generate 1500 data points from the 2-D circles dataset of sklearn:

```
sklearn.datasets.make_circles
```

Randomly create train, validation, and test splits of 1000, 250, and 250 points, respectively. Evaluate the above classifiers in this setting.

1. (5 pts) Show decision boundaries for Linear SVM and Logistic Regression classifiers.
2. (5 pts) Show decision boundaries for Kernel SVM and Kernel Logistic Regression (use rbf, polynomial kernels). Try different values of hyperparameters, and report results with whichever works best.
3. (5 pts) Train Neural Network from HW4, and k-NN classifiers on this dataset and show decision boundaries. (You can use library implementation for these classifiers).
4. (5 pts) What are your conclusions from this exercise?

3.4 Evaluation on Real Dataset (20 pts)

Let's put all this to some real use. For this problem, use the Wisconsin Breast Cancer dataset. You can download it from the sklearn library:

```
sklearn.datasets.load_breast_cancer
```

1. (10 pts) Do all the points of Section 3.3 in this dataset. Since these are high-dimensional data, you do not have to show the decision boundaries. Report test accuracies for these classifiers and discuss your findings.

2. (10 pts) In addition, you also want to figure out the important features which determine the class. Which regularization will you use for this? Upgrade your SVM, Kernel SVM implementation to include this regularization. Discuss the important features that you obtain by running your regularized SVM on this dataset. (You might need to normalize this dataset before training any classifier.)