Homework 2

CSCI 5525: Machine Learning

Due on Oct 10th 11am (before class)

Please type in your info:

- **Name**:Anubhav Panda
- **Student ID**:5509727
- **Email**:panda047@umn.edu
- **Collaborators, and on which problems:**

Homework Policy. (1) You are encouraged to collaborate with your classmates on homework problems, but each person must write up the final solutions individually. You need to fill in above to specify which problems were a collaborative effort and with whom. (2) Regarding online resources, you should **not**:

- Google around for solutions to homework problems,
- Ask for help on online.
- Look up things/post on sites like Quora, StackExchange, etc.

Submission. Submit a PDF using this LaTeX template for written assignment part and submit .py files for all programming part. You should upload all the files on Canvas.

Written Assignment

Instruction. For each problem, you are required to write down a full mathematical proof to establish the claim.

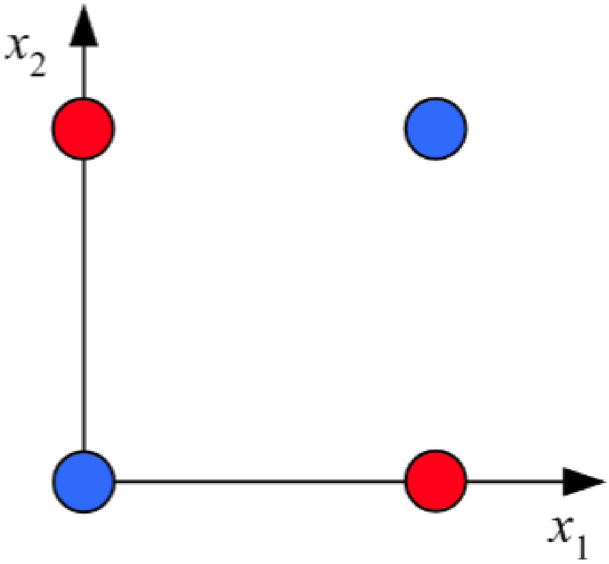
Problem 1. Separability.

(**3 points**) Formally show that the XOR data set (see Lecture 4 note) is not linearly separable.
Hint: A data set $\{(x_i, y_i)_{i=1}^{N}\}$ where $y_i \in \{-1, 1\}$ is linearly separable if $\exists \mathbf{w} \in \mathbb{R}^d$ and $\exists b \in \mathbb{R}$ s.t.

$$\text{sign}(\langle \mathbf{w}, x_i \rangle + b) = y_i \qquad \forall i$$

Your answer. for simplicity lets assume n=4 and xor function can be defined as the following figure.i am trying to compute w_1, w_2, b so that a linear line can separate those two classes if the data is linearly separable.
the linear line that will separate the points can be defined as
$w_1 X_i + w_2 X_i + b = y_i$
No set of w_1, w_2, b can satisfy the following things
for($x_1 = 0$ and ($x_2 = 0$) $y_i = b$ and $y_i <= 0$
for($x_1 = 0$ and ($x_2 = 1$) $y_i = w_2 + b$ and it should be $y_i >= 0$
for($x_1 = 1$ and ($x_2 = 0$) $y_i = w_1 + b$ and it should be $y_i > 0$
for($x_1 = 1$ and ($x_2 = 1$) $y_i = w_1 + w_2 + b$ and it should be $y_i <= 0$ Hence proved the data is not linearly separable.

Figure 1: Xor Figure



Problem 2. Kernels.

(**4 points**) As you learned in the class, Kernels provide a powerful method to traverse between kernel space and feature space. Using Kernel's properties (mention the properties used):

- (**2 points**) Show that $K(x, y) = K_1(x, y)K_2(x, y)$ is a valid kernel where K_1 and K_2 are valid kernels.
- (**2 points**) Show that the function $K(x, y) = K_1(x, y) + K_2(x, y)$ is a valid kernel function where $x, y \in \mathbb{R}$.

Your answer. lets say K_1 and K_2 are kernels having feature map ϕ_1 and ϕ_2 respectively.and
$K_1(x, y)K_2(x, y) = (\Phi_1(x_1)^\intercal \Phi_1(y_1))(\Phi_2(x_1)^\intercal \Phi_2(y_1))$
$= (\sum_{n=1}^{\infty} f_n(x)f_n(y))(\sum_{m=1}^{\infty} g_m(x)g_m(y))$
$= \sum_{n,m=1}^{\infty}(f_n(x)f_n(y))(g_m(x)g_m(y))$
$= \sum_{n,m=1}^{\infty}(f_n(x)g_m(x))(f_n(y)g_m(y))$

$= \sum_{n,m=1}^{\infty} f_{n,m}(x)g_{n,m}(y)$
$= \Phi_3(x)\Phi_3(y)$
$= K(x,y)$(proved)
b)
$K_1(x,y) + K_2(x,y) = \Phi_1(x)\Phi_1(y) + \Phi_2(x)\Phi_2(y)$
$= \Phi_3(x)\Phi_3(y)$
= K(x,y) *(proved)*

Problem 3. Derivation of SVM Solution.

(**5 points**) In the lecture, we derived the soft-margin SVM solution without the intercept term. Now derive the solution with the intercept term b by going through Lagrange duality. Here the predictor will be $\hat{f}(x) = \text{sign}(\mathbf{w}^\intercal x + b)$. You should use the same conventions for the scalar and vector variables as used in the course.

- What is the Lagrange dual objective?
- State the two relevant KKT conditions: stationarity and complementary slackness.
- What is the condition for which one would get support vectors?
- What is the optimum $\mathbf{w}$ and b?

Note: Be concise in answering the questions above.

Your answer. $\hat{f}(x) = \text{sign}(\mathbf{w}^\intercal x + b)$.
for soft margin svm we have the following condition
$\min_{\mathbf{w}} \ \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\xi$
$\forall i, \quad y_i(\mathbf{w}^\intercal x_i + b) >= 1 - \xi$
$\forall i, \quad \xi >= 0$
For each of the constraint we can introduce Lagrangian multipliers $\lambda_j >= 0, \alpha_j >= 0$
so the it can be defined as
$L(\mathbf{w}, \sigma, \lambda, \alpha)= \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\xi + \sum_{i=1}^{N}\lambda(1 - \xi - y_i(\mathbf{w}^\intercal x_i + b)) - \sum_{i=1}^{N}\alpha\xi$
This is defined as Loss function
The KKT Condition can be defined as $L(\mathbf{w}, \sigma, \lambda, \alpha)= \frac{1}{2}\|\mathbf{w}\|_2^2$
$\sum_{i=1}^{N}\lambda(1 - \xi - y_i(\mathbf{w}^\intercal x_i + b)) = 0, \sum_{i=1}^{N}\alpha\xi = 0$(Complementary slackness)
$\Delta_w L(\mathbf{w}, \sigma, \lambda, \alpha) = \Delta_w(\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\xi + \sum_{i=1}^{N}\lambda(1 - \xi - y_i(\mathbf{w}^\intercal x_i + b)) - \sum_{i=1}^{N}\alpha\xi) = 0$ (Stationarity)
as we have seen above $(\sum_{i=1}^{N}\lambda(1 - \xi - y_i(\mathbf{w}^\intercal x_i + b)) = 0)$support vectors can be calculated by the value of x and y for which $(1 - \xi - y_i(\mathbf{w}^\intercal x_i + b)) = 0)$
after solving $\Delta L(\mathbf{w}, \sigma, \lambda, \alpha) = 0$ we get
$\mathbf{w} = \sum_{i=1}^{N} y_i\lambda_i x_i \qquad \frac{\partial L}{\partial w} = 0$
$C - \lambda_i - \alpha_i = 0 \qquad \frac{\partial L}{\partial \xi_i} = 0$
$\sum_{i=1}^{N}\lambda_i y_i = 0 \qquad \frac{\partial L}{\partial b} = 0$
As we have seen above optimum w can be calculated by
$\mathbf{w} = \sum_{i=1}^{N} y_i\lambda_i x_i$
and b can be calculated by
$b = y_i - w^\intercal X_i$ and we can compute it by the first value of training data ie it can be written as

$b = y_i[0] - w^{\intercal} X_i[0]$
The dual Can be defined by $L(\mathbf{w}, \sigma, \lambda, \alpha)$= $\sum_i \lambda_i - \frac{1}{2} \sum_{i,j \epsilon [n]} \lambda_i \lambda_j y_i y_j x_i^{\intercal} x_j$
such that $c = \lambda_i + \alpha_i$ and $\lambda_i, \alpha_i >= 0$

Problem 4. Softmax Regression

(**6 points**) Consider softmax regression with K classes, no bias terms, and inputs $x_i \in \mathbb{R}^d$. $w^1, ..., w^K \in \mathbb{R}^d$ denote the weight vectors corresponding to each class, and $W \in \mathbb{R}^{d \times K}$ is the weight matrix with w_k, $1 \leq k \leq K$, as its columns. According to softmax regression model:

$$p(Y = c|x, W) = S(W^T x)_k$$

where $S(W^T x)_k = \frac{\exp{(\langle w_k, x \rangle)}}{\sum_{k'} \exp{(\langle w_{k'}, x \rangle)}}$

1. (**2 points**) Let $v \in \mathbb{R}^d$ be some fixed vector. For $1 \leq k \leq K$, let $w'_k = w_k + v$ and W' is the corresponding weight matrix. Prove $S(W^T x) = S((W')^T x)$

2. (**3 points** Suppose we train a softmax regression model on a some given dataset, once by fitting all weight vectors and once by fixing $w_K = 0_d$ and fitting the remaining weight vectors. Does the likelihood of the training data differ if we use maximum likelihood estimation?

3. (**1 point**) Interpret the below ratio

$$\frac{S(W^T x)_{k1}}{S(W^T x)_{k2}}$$

for $1 \leq k_1, k_2 \leq K$.

1 $S((W')^T x) = \frac{\exp{(\langle w'_k, x \rangle)}}{\sum_{k'} \exp{(\langle w'_{k'}, x \rangle)}}$
$= \frac{\exp{(\langle w_k + v, x \rangle)}}{\sum_{k'} \exp{(\langle w_k + v, x \rangle)}}$
$= \frac{\exp{(\langle w_k, x \rangle)} \exp{(\langle v, x \rangle)}}{\sum_{k'} \exp{(\langle w_k, x \rangle)} \exp{(\langle v, x \rangle)}}$
$= \frac{\exp{(\langle w_k, x \rangle)}}{\sum_{k'} \exp{(\langle w_k, x \rangle)}}$
$= S((W)^T x)$(Proved)
2 No the likelihood of the training data wont differ if we use maximum likelihood estimation .If we fix $-w_k = v$ and as we have seen in the first proof $S((W)^T x) = S((W')^T x)$ the result is always the same. so it will adjust the other weights accordingly.
3

$$\frac{S(W^T x)_{k1}}{S(W^T x)_{k2}}$$

$=\exp{(\langle w_{k1}, x \rangle - \langle w_{k2}, x \rangle)}$ in case of K=2 it can be defined as odd's ratio.

Programming Assignment

Instruction. For each problem, you are required to report descriptions and results in the PDF and submit code as python file (.py) (as per the question).

- **Python** version: Python 3.
- Please follow PEP 8 style of writing your Python code for better readability in case you are wondering how to name functions & variables, put comments and indent your code
- **Packages allowed**: numpy, pandas, matplotlib, cvxopt
- **Submission**: Submit report, description and explanation of results in the main PDF and code in .py files.
- Please PROPERLY COMMENT your code in order to have utmost readability

Problem 5. Linear SVM dual form.

(**20 Points**)

SVM can be implemented in either primal or dual form. In this assignment, your goal is to implement a linear SVM in dual form using slack variables. **The quadratic program has a box-constraint on each Lagrangian multiplier α_i, $0 <= \alpha_i <= C$.**

For doing this you would need an optimizer. Use an optmizer cvxopt which can be easily installed in your environment either through pip or conda.

1. (**10 Points**) Implement SVM. Write a training function 'svmfit' to train your model and a prediction function 'predict' that predicts the labels using the model.

   You have to use this "hw2data.csv" dataset for this assignment. The labels are in the last column. It is a 2 class classification problem. Split this dataset into 80-20 % split and hold out the last 20% to be used as test dataset. Then, you have to implement k=10 fold cross validation on the first 80% of the data as split above.

   Report the test performance (performance on held out test data) of your trained model for the following cases and provide your reasoning (describing the result is not explanation but you must explain the variation 'why' the result is the way it is):

2. (**2 Points**) Summarize and explain the methods and equations you implemented.
3. (**4 Points**) Vary C in the range [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000] and report the average train, validation and test accuracy as C varies. Provide the plots.
4. (**4 Points**) Explain the train, validation and test performance with C? Reason about it. As a designer, what value of C (and on what basis) would you choose for your model - explain.

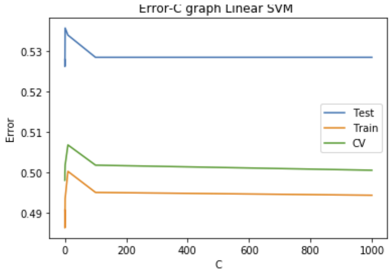
Submission: Submit all plots requested and generated while performing hyperparameter tuning and explanations in latex PDF. Submit your program in a file named (hw2_svm.py).

Your answer. 2 i have implemented the following equations.respectively.
$b = y_i[0] - w^\intercal X_i[0]$
$y_i = w^\intercal X_i + b$
w can be replace by the following equation $\mathbf{w} = \sum_{i=1}^{N} y_i \lambda_i x_i$ resulting $y_j = \sum_{i=1}^{N} y_i \lambda_i x_i^\intercal x_i + b$ and i have used it to determine the label of y.

I have used the following methods in the implementation of the code as well.

svm_fit:used to learn support_vectors, support_vector_labels, Lagrange Multiplier value, intercept predict:it is used to predict the label of the given data set using support_vectors, support_vector_labels, Lagrange Multiplier value, intercept error:used to calculate the error model:method which calls the above-mentioned methods in a proper coordinated way.
3

Figure 2: Linear SVM Error-C



4 As we are increasing C the value of the error decreases at first and then it increases as shown in the linear SVM Error -C curve .If the dataset is not linearly separable after certain iteration the error rate converges as shown in the figure.K - fold cross validation can be used to determine the hyper parameters. and as we have seen here C is the only hyper parameter in case of linear SVM we are determing it by using 10 fold cross validation.I am going to choose the c value for which the cross validation eroor is minimum.

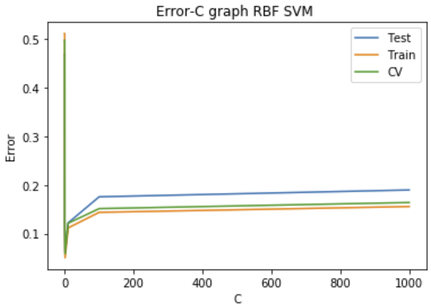
Problem 6. Kernel SVM.

(**7 Points**)

1. Implement a Kernel SVM for a generic kernel.
2. Now use the linear SVM implemented in Problem 4 and RBF-SVM (by making use of aforementioned Kernel SVM code in part 1) on the data set ("hw2data.csv"). Implement rbf_svm_train and rbf_svm_predict functions. Use the cross validation similar to Problem 4 and report validation and test error for each fold - plot it. Compare the accuracies achieved using linear SVM and RBF-SVM, briefly explain the difference.

Submission: Submit all plots requested and generated and explanations in latex PDF. Submit your program in files named (hw2_kernel_svm.py).

Figure 3: RBF Error-C



Your answer. As we have seen in the linear SVM the accuracy is around 50% and as i have gone through the data its not linearly separable.But if we want to transform the data to the high dimension using the rbf kernel and we can conclude that the data is linearly separable since i am getting minimum 6% error rate on test data and the data can be seen in the RBF Error-C plot as well.

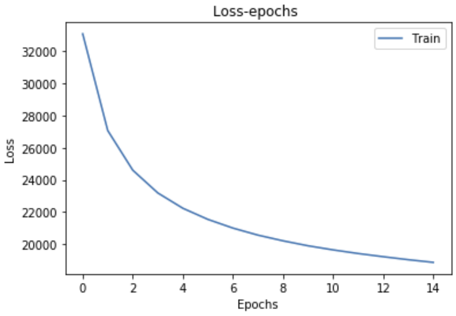
Problem 7. Multi-class logistic regression.

(**15 Points**) This problem is about multi class classification and you will use MNIST dataset. In the hw2 folder, we have put the mnist files in csv format. There are two .zip files: mnist_train and mnist_test. Use mnist_train for training your model and mnist_test to test. First column in these files are the digit labels.

1. (**7 Points**) Implement a multi-class logistic regression for classifying MNIST digits. Refer the class notes on classifying multi classes. You should use mini-batches for training the model. Save the final trained weights of your model in a file and submit it. The code should have at least two functions: multi_logistic_train and multi_logistic_predict.
2. (**3 Points**) Report and briefly explain loss function as training with mini-batches progresses.
3. (**3 Points**) Report confusion matrix and accuracy for the test data.

Submission: Submit all plots requested and generated along with explanations in latex PDF. Submit your program in files named (hw2_multi_logistic.py).

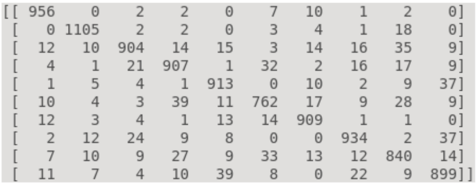
Figure 4: Loss-epochs



Your answer. As we can see in Loss-epochs as training with mini-batches progresses the loss value decreases.Since the parameters w,b are updated with each epoch and it uses gradient descent to get the optimum value of w and b.

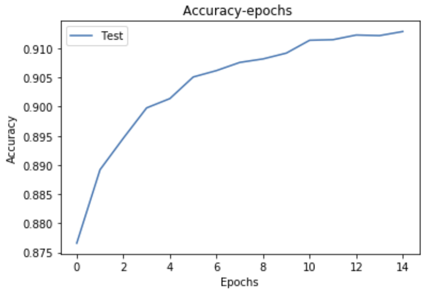
3 the test accuracy for 15 epochs [0.8766, 0.8892, 0.8946, 0.8998, 0.9014, 0.9051, 0.9062, 0.9076, 0.9082, 0.9092, 0.9114, 0.9115, 0.9123, 0.9122, 0.9129]
confusion matrix values [[956 0 2 2 0 7 10 1 2 0] [0 1105 2 2 0 3 4 1 18 0] [12 10 904 14 15 3 14 16 35 9] [4 1 21 907 1 32 2 16 17 9] [1 5 4 1 913 0 10 2 9 37] [10 4 3 39 11 762 17 9 28 9] [12 3 4 1 13 14 909 1 1 0] [2 12 24 9 8 0 0 934 2 37] [7 10 9 27 9 33 13 12 840 14] [11 7 4 10 39 8 0 22 9 899]]

Figure 5: confusion matrix



```
[[ 956    0    2    2    0    7   10    1    2    0]
 [   0 1105    2    2    0    3    4    1   18    0]
 [  12   10  904   14   15    3   14   16   35    9]
 [   4    1   21  907    1   32    2   16   17    9]
 [   1    5    4    1  913    0   10    2    9   37]
 [  10    4    3   39   11  762   17    9   28    9]
 [  12    3    4    1   13   14  909    1    1    0]
 [   2   12   24    9    8    0    0  934    2   37]
 [   7   10    9   27    9   33   13   12  840   14]
 [  11    7    4   10   39    8    0   22    9  899]]
```

Figure 6: Accuracy


Accuracy-epochs
Test
Accuracy
Epochs