COMP 540 Homework 4

Tony Ren (netid: lzr1), Hanyang Li (netid: hl43)

March 6, 2018

Problem 1: Intuitions about support vector machines

Problem 1.1

Intuitively, the further a data point is from the decision boundary, the more confident we are about the class of this point. By maximizing the margin, we will create a decision boundary that is the furthest away from the data points, which gives us the highest certainty about the classes of the data. Therefore, it makes sense that models with bigger margins will generalize better.

Problem 1.2

No. Moving data points that are not support vectors further away from the decision boundary will not affect the hinge loss. This can be shown mathematically. The hinge loss function is:

$$hingeloss = \sum_{i=1}^{m} max(0, 1 - h_\theta(x^{(i)})y^{(i)})$$

Assume we do not allow any misclassified data when training the model. Then, $h_\theta(x^{(i)})y^{(i)} = |h_\theta(x^{(i)})|$, which is equal to 1 when point i is a support vector or is greater than 1 if point i is not a support vector. For points that are not support vectors, $1 - h_\theta(x^{(i)})y^{(i)} < 0$ must be true. Therefore, their contribution to the hinge loss is $max(0, 1 - h_\theta(x^{(i)})y^{(i)}) = 0$. Thus, moving the points that are not support vectors further away from the decision boundary will not affect the hinge loss.

Note: Please find writeup for problem 2 and problem 3 attached at the end.

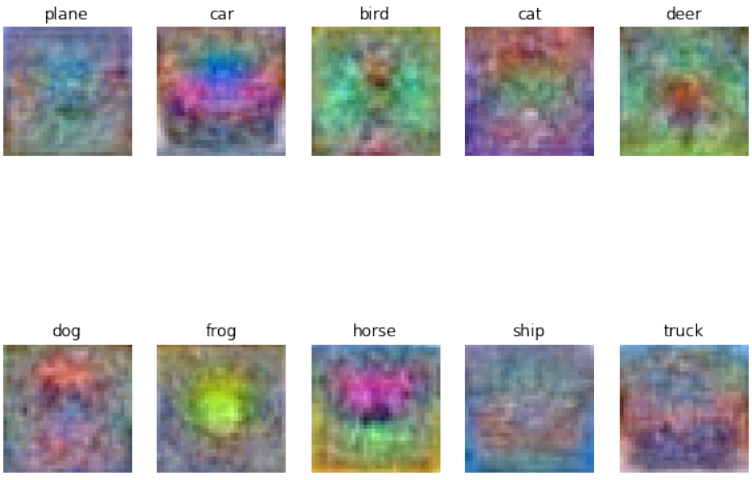
Problem 4: Support Vector Machine for Multi-class Classification

Problem 4A

Since the loss function of SVM is not strictly differentiable, the gradient we calculate may not be perfectly accurate. Therefore, it can be different from the gradient found by numerical methods.

Problem 4E

Training time
SVM: 9.028862s (learning rate = 1e-7, iteration = 1500)
Softmax: on the order of minutes
According to the results, Softmax takes longer time to train.
Test set accuracy
SVM: 0.365700
Softmax: approximately 0.30
The SVM model has a slightly higher test accuracy than the Softmax model.
Visualizations of the θ parameters learned
SVM:

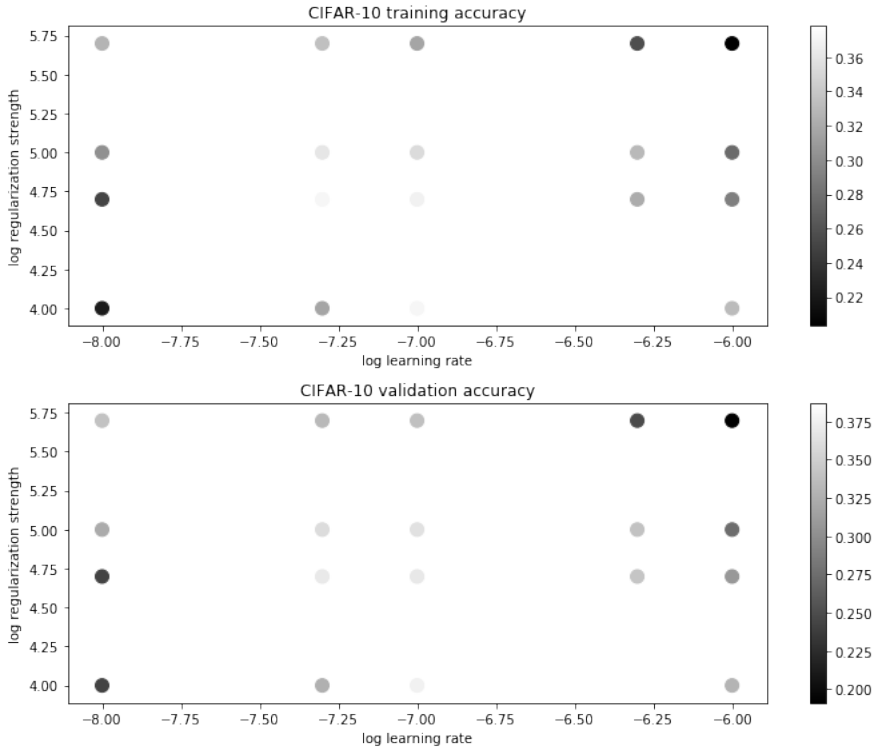


Softmax:

The figures of SVM's parameters are brighter and more colorful, indicating that SVM tends to have larger variance in its parameter values.

Hyper parameters selection

SVM:



Best hyper parameters are: learning rate = 5e-7, regularization strength = 1e4

Softmax:

Best hyper parameters are: learning rate = 1e-6, regularization strength = 1e8

According to our results, the optimal learning rates for Softmax and SVM are similar, while the optimal regularization strength of Softmax is much bigger than that of SVM.

HW4 Problem 2

$D = \{(0,-1), (\sqrt{2}, +1)\} = (x^1, y^1)$ and (x^2, y^2)

Kernel: $\phi(x) = (1, \sqrt{2}x, x^2)$

$\phi(x^1) = (1, \sqrt{2(0)}, 0^2) = (1, 0, 0)$

$\phi(x^2) = (1, \sqrt{2(2)}, \sqrt{2}^2) = (1, 2, 2)$

Want to minimize $\frac{1}{2}\|\theta\|^2$ such that

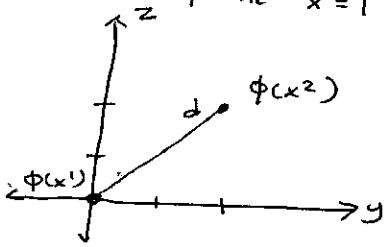
$y^1(\theta^T \phi(x^1) + \theta_0) \geq 1$

$y^2(\theta^T \phi(y^2) + \theta_0) \geq 1$

A vector parallel to the optimal vector θ is the distance vector between the two closest points of each class

$d = \phi(x^1) - \phi(x^2) = \boxed{(0, -2, -2)}$

Visualizing on the plane $x = 1$



This distance makes up the slab (2x the margin)

$\text{slab} = \sqrt{(0)^2 + (-2)^2 + (-2)^2} = \sqrt{8} = \sqrt{4 \cdot 2} = 2\sqrt{2}$

$\text{margin} = \frac{\text{slab}}{2} = \frac{2\sqrt{2}}{2} = \boxed{\sqrt{2}}$

$\text{margin} = \frac{1}{\|\theta\|} = \sqrt{2}$

$\|\theta\| = \frac{1}{\sqrt{2}} = \sqrt{\theta_1^2 + \theta_2^2 + \theta_3^2}$

$\theta_1^2 + \theta_2^2 + \theta_3^2 = \frac{1}{2}$

θ must be parallel to d $(0, -2, -2)$ and therefore takes on the form $(0, m, m)$

$0^2 + m^2 + m^2 = \frac{1}{2} = 2m^2$

$m = \sqrt{\frac{1}{4}} = \pm \frac{1}{2}$

$\boxed{\theta = (0, \frac{1}{2}, \frac{1}{2})}$ or $(0, -\frac{1}{2}, -\frac{1}{2})$

using $\theta = (0, \frac{1}{2}, \frac{1}{2})$:

$y^1(\theta^T \phi(x^1) + \theta_0) \geq 1 \longrightarrow -1([0, \frac{1}{2}, \frac{1}{2}]^T [1, 0, 0] + \theta_0) \geq 1$ ①

$y^2(\theta^T \phi(x^2) + \theta_0) \geq 1 \longrightarrow +1([0, \frac{1}{2}, \frac{1}{2}]^T [1, 2, 2] + \theta_0) \geq 1$ ②

From ① $-1(0(1) + \frac{1}{2}(0) + \frac{1}{2}(0) + \theta_0) \geq 1$

$-1(\theta_0) \geq 1$

$\theta_0 \leq -1$

From ② $+1(0(1) + \frac{1}{2}(2) + \frac{1}{2}(2) + \theta_0) \geq 1$

$+1(2 + \theta_0) \geq 1$

$\theta_0 \geq -1$

$\therefore \boxed{\theta_0 = -1}$

Equation of Decision Boundary

$\boxed{\theta^T x + \theta_0 = 0}$

$(0, \frac{1}{2}, \frac{1}{2})^T x + -1 = 0$

$x = (x, y, z)$

$0x + \frac{1}{2}y + \frac{1}{2}z = 1$

$\boxed{y + z = 2}$

using $\theta = (0, -\frac{1}{2}, -\frac{1}{2})$

$-1(0(1) - \frac{1}{2}(0) - \frac{1}{2}(0) + \theta_0) \geq 1$

$-\theta_0 \geq 1$

$\theta_0 \leq -1$

$+1(0(1) + -\frac{1}{2}(2) + -\frac{1}{2}(2) + \theta_0) \geq 1$

$+1(-2 + \theta_0) \geq 1$

$\theta_0 \geq 3$

$\therefore \theta_0 \neq (0, -\frac{1}{2}, -\frac{1}{2})$

Spam Filter

The higher dimensional dataset of the spam filter was difficult to visualize in a way that would make it easy to determine whether or not the data was linearly separable or not, which affected our decision of whether to kernelize the data. To test whether the data was linearly separable, we ran the two class linear SVM without a kernel to see its accuracy. The data was taking a very long time to run on the linear SVM without scaling, thus we scaled X utilizing the standard scaling in sklearn. We separated the training data 80/20 into a training and validation data set to help with our hyperparameter search. To find the best hyperparameters including C, number of iterations, and learning rate, we searched the entire space starting with C values. We varied 9 C values to choose the best validation accuracy with a small number of iterations and step size of 1e-4 to make sure the values converged. The results are shown in Table 1 below.

C value	Validation Accuracy
0.01	0.94625
0.03	0.94625
0.1	0.94625
0.3	0.95875
1	0.95875
3	0.9625
10	0.97375
30	0.98
100	0.98625

Table 1: Hyperparameter C Search

Next, after choosing the best C (C=100), we wanted to optimize the number of iterations and the learning rate. We tested different values of these two parameters and found the following validation accuracies.

Learning Rate	Number of Iterations	Validation Accuracy
1e-2	1000	0.99
1e-2	5000	0.99
1e-2	10000	0.99
1e-2	20000	0.99
1e-2	50000	0.98875
1e-3	1000	0.98
1e-3	5000	0.98
1e-3	10000	0.98
1e-3	20000	0.98
1e-3	50000	0.98
1e-4	1000	0.98625
1e-4	5000	0.97875
1e-4	10000	0.97875
1e-4	20000	0.9775
1e-4	50000	0.9775
1e-5	1000	0.97535
1e-5	5000	0.985

1e-5	10000	0.98625
1e-5	20000	0.98
1e-5	50000	0.97875

Table 2: Hyperparameter Learning Rate and Number of Iterations Search

To save time, we performed the C calculation separately from the learning rate and number of iterations but to do get a complete hyperparameter search, we would have varied C with every number of iterations and learning rate as well. The best learning rate was found to be 0.01 and the best number of iterations was found to be 1000.

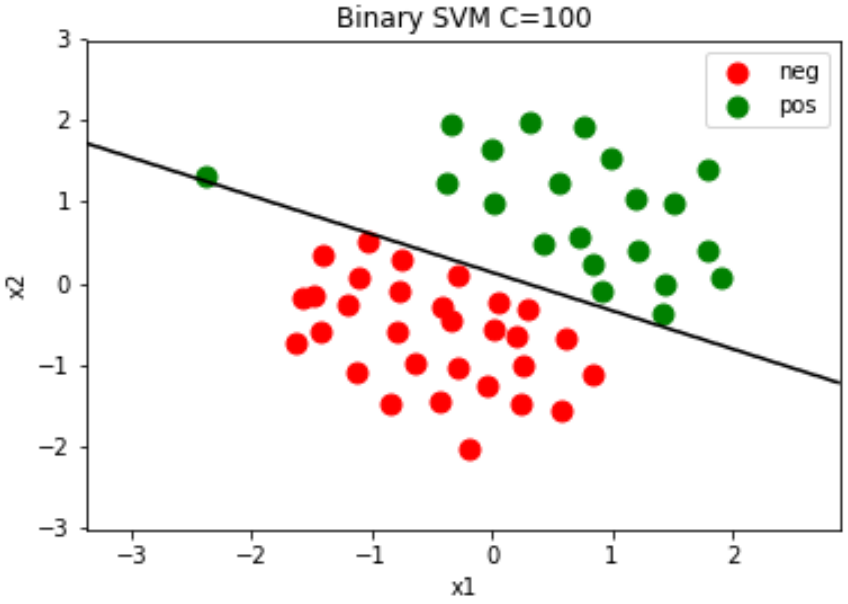
After testing these hyperparameters on the testing set, we obtained a test set accuracy of 0.98 and training data accuracy of 98.425. We thought that this accuracy was very high for this dataset and that this meant the data was linearly separable. This made utilizing a kernel unnecessary as it would expend additional computational power for similar accuracies. We did make a simple Gaussian kernel with the best hyperparameters above and sigma = 1 that had a test accuracy of 0.822. This kernel performed less well than the linear SVM and therefore we though the linear SVM was the correct way to go.

We obtained a list of the top 15 spam and ham words and have listed them in the table below.

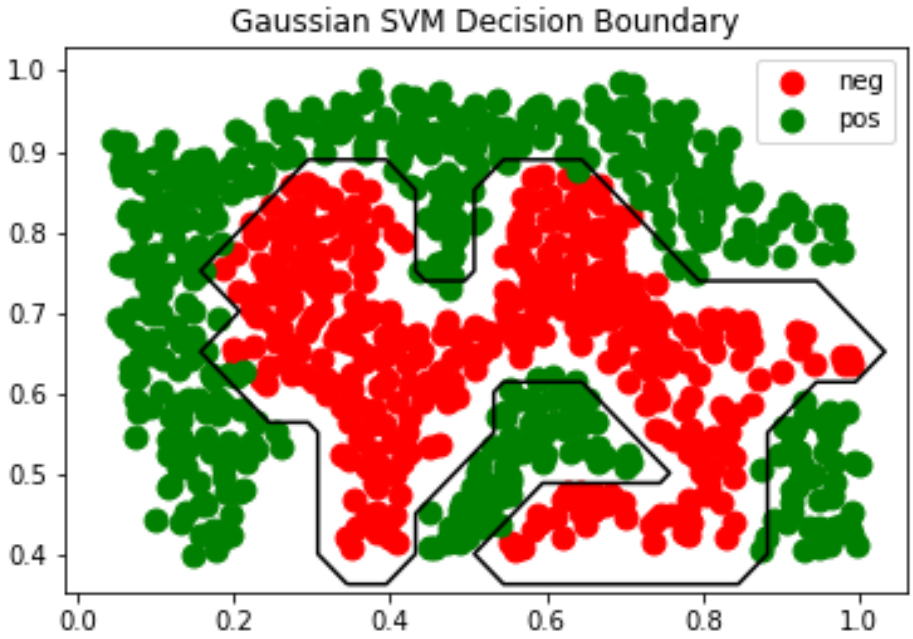
Spam	Ham
clearly	instant
otherwiss	urgent
remot	datapow
believ	wrong
file	numberth
franc	issu
natur	that
dollarac	useless
water	predict
creativ	submit
off	august
young	these
gt	anyth
herba	url
reason	new

Figures for Problem 3

Binary_SVM_C=100



Gaussian_Dataset2_DecisionBoundary



Gaussian_Dataset3_DecisionBoundary