Homework 4

Young Won Kim (yk41) and Minh Pham (mnp7)

Spring 2017

1 Kernelizing k-nearest neighbors

Distance between two training examples in K-nearest neighbors

$\|\phi(x) - \phi(x')\|^2 = \Sigma(\phi_i(x) - \phi_i(x'))^2$

$\|\phi(x) - \phi(x')\|^2 = \Sigma(\phi_i(x)^2 - \phi_i(x')^2 - 2\phi_i(x)\phi_i(x')) = k(x,x) - k(x',x') - 2k(x,x')$

KNN algorithms is 3 kernelized functions of $k(x,x), k(x',x')$, and $k(x,x')$.

2 Constructing kernels

$k(x,x') = ck_1(x,x')$ for $c > 0$
$\alpha^T k\alpha = c\alpha^T k_1 \alpha \geq 0$
$k(x,x')$ is positive semi definite and thus, $k(x,x')$ is a valid kernel

$k(x,x') = f(x)k_1(x,x')f(x')$
$k(x,x') = f(x)f(x')k_1(x,x')$
$k(x,x') = (\psi(x)^T\psi(x'))k_1(x,x')$
$k(x,x') = k(\psi(x),\psi(x'))k_1(x,x')$
We have, $k(\psi(x),\psi(x')) \geq 0$ and $k_1(x,x') \geq 0$
Therefore, $k(x,x')$ is positive semi definite and thus, $k(x,x')$ is a valid kernel

$k(x,x') = k_1(x,x') + k_2(x,x')$
$\alpha^T k\alpha = \alpha^T k_1\alpha + \alpha^T k_2\alpha \geq 0$
$k(x,x')$ is positive semi definite and thus, $k(x,x')$ is a valid kernel

3 Fitting an SVM classifier by hand

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

$min\frac{1}{2}\|\theta\|^2$

s.t. $y^{(1)}(\theta^T\phi(x^{(1)})+\theta_0)\geq 1$ and $y^{(2)}(\theta^T\phi(x^{(2)})+\theta_0)\geq 1$

Point 1: $(0,-1)\rightarrow(1,0,0)$ and Point 2: $(\sqrt(2),+1)\rightarrow(1,2,3)$

The vector parallel with θ is $\theta^*=\phi(x_2)-\phi(x_1)=[0,2,2]$

To solve for margin:
$margin=\frac{1}{\|\theta\|}$
$\rightarrow\|\theta\|=\frac{1}{\sqrt{2}}$

To solve for θ:
We have, $\theta=k\theta^*$
$\rightarrow\sqrt{x_1^2+x_2^2+x_3^2}=\sqrt{k^2(x_{1*}^2+x_{2*}^2+x_{3*}^2)}=\frac{1}{\sqrt{2}}$
$\rightarrow k\sqrt{0+4+4}=\frac{1}{\sqrt(2)}\rightarrow k=\frac{1}{4}$

Therefore, $\theta=k\theta^*=[0,0.5,0.5]$

To solve for the intercept θ_0:
Since the two points are supporting vectors, the inequalities are tight.
$y^{(1)}(\theta^T\phi(x^1)+\theta_0)=-1([0,0.5,0.5]^T[1,0,0]+\theta_0]=-\theta_0=1$
$y^{(1)}(\theta^T\phi(x^1)+\theta_0)=+1([0,0.5,0.5]^T[1,2,2]+\theta_0]=2+\theta_0=1$
$\rightarrow\theta_0=-1$

The equation for the decision boundary:
$y=\theta^T\phi(x)+\theta_0=[0,0.5,0.5]^T\phi(x)-1$ where $\phi(x)=(1,\sqrt{2}x,x^2)$

4 Support vector machines for binary classification

4.1A The hinge loss function and gradient

We get: J = 1.0 grad = [-0.12956186 -0.00167647]

4.1B Example dataset1: impact of varying C

When C=1, SVM misclassifies one data point (Figure 1).
When C=100, SVM classifies every point correctly, but the margin is narrower (Figure 2).

4.1C Gaussian Kernel

We get Gaussian kernel value of 0.324652467358.

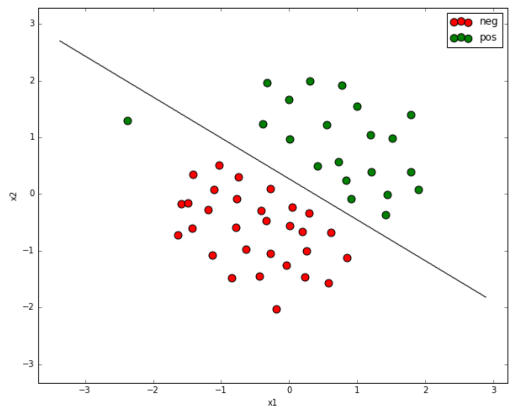

Figure 1: SVM decision boundary with C=1

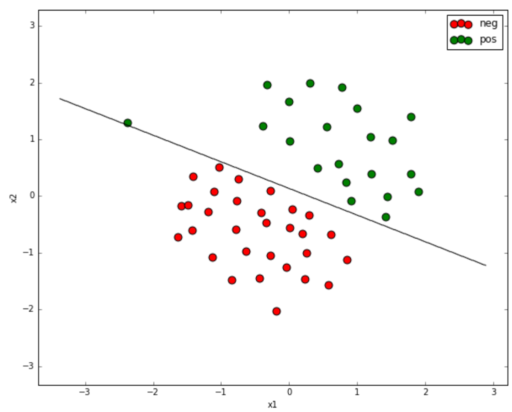

Figure 2: SVM decision boundary with C=100

4.2 Example dataset3: selecting hyperparameters for SVMs

The hyperparameters that give the best accuracy (=0.96) are sigma= 0.1 c= 0.1.

The decision boundary returned with our best parameters is shown in Figure 3.

4.3 Spam Classification with SVMs

We performed svm classifier without kernel nor scaling. We performed cross-validation in which 0.8 of data is training set and the rest is the validation set to pick out the best hyperparameters. We ran across a number of combinations of hyperparameters.
$C = [0.1, 0.3, 1, 3, 10, 30]$
$lr = [1e-2, 3e-2, 1e-1, 3e-1, 1, 3]$

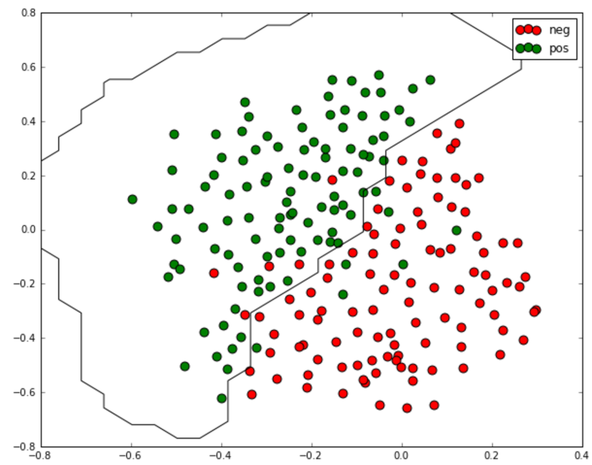

Figure 3: SVM gaussian kernel decision boundary with best hyperparameters

$iter = [1000, 5000, 10000, 25000]$

At C = 0.1, lr = 0.03, and number of iterations = 5000, we got an accuracy of 0.97 on the validation set. Using these hyperparameters, we achieved accuracy of 0.98825 on overall training data and accuracy of 0.988 on test data.

We also tried another set of hyperparameters that have high accuracy in the cross-validation, which are C = 0.1, lr = 0.1, number of iterations = 10000. This set of hyperparameters has accuracy of 0.96375 on cross-validation training set, of 0.993 on overall training data, and of 0.991 on test data. The set of hyperparameters might be less overfitting than the previously mentioned set of hyperparameters, allowing it to achieve higher accuracy on the training and test data.

Our top 15 predictors of spam for non-kernelized, non-scaled SVM classifier at C = 0.1, lr = 0.03, and iter = 5000 are: click, remov, our, basenumb, guarante, pleas, you, free, visit, here, nbsp, will, hour, most, dollar.

5 Support vector machines for multi-classification

SVM takes longer to train and achieves lower performance. Best accuracy for SVM was 0.393, whereas best accuracy for Softmax was 0.411. The visualizations of parameters learned by both methods look very similar. Best hyperparamers for SVM were lr=5.0e-07 and reg=1.0e+04, and best hyperparameters for Softmax were lr=1.0e-06 and reg=5.0e+04.

For the same hyperparameters, at the small learning rate, softmax has higher train and test accuracy than svm, and vice versa. With a higher regularization, the train accuracy

C	Learning rate	Iteration	Validation accuracy
0.1	0.01	1000	0.96
0.1	0.01	5000	0.96875
0.1	0.01	10000	0.97
0.1	0.01	25000	0.96625
0.1	0.03	1000	0.965
0.1	0.03	5000	0.97125
0.1	0.03	10000	0.96625
0.1	0.03	25000	0.96625
0.1	0.1	1000	0.97125
0.1	0.1	5000	0.965
0.1	0.1	10000	0.96375
0.1	0.1	25000	0.95375
0.1	0.3	1000	0.96625
0.1	0.3	5000	0.96
0.1	0.3	10000	0.95625
0.1	0.3	25000	0.94125
0.1	1	1000	0.96375
0.1	1	5000	0.95125
0.1	1	10000	0.95
0.1	1	25000	0.94625
0.1	3	1000	0.9575
0.1	3	5000	0.96375
0.1	3	10000	0.95375
0.1	3	25000	0.9625

Figure 4: Accuracy with C = 0.1

in svm got a little bit worse while the train accuracy in softmax improves. No significant change in test accuracy was observed in softmax and svm when increasing regularization parameters at the same learning rates.

C	Learning rate	Iteration	Validation accuracy
0.3	0.01	1000	0.965
0.3	0.01	5000	0.97
0.3	0.01	10000	0.965
0.3	0.01	25000	0.96375
0.3	0.03	1000	0.97
0.3	0.03	5000	0.965
0.3	0.03	10000	0.96375
0.3	0.03	25000	0.95125
0.3	0.1	1000	0.96625
0.3	0.1	5000	0.95875
0.3	0.1	10000	0.95375
0.3	0.1	25000	0.94875
0.3	0.3	1000	0.9625
0.3	0.3	5000	0.95375
0.3	0.3	10000	0.95375
0.3	0.3	25000	0.9475
0.3	1	1000	0.95875
0.3	1	5000	0.96
0.3	1	10000	0.95375
0.3	1	25000	0.955
0.3	3	1000	0.96125
0.3	3	5000	0.96375
0.3	3	10000	0.9625
0.3	3	25000	0.95875

Figure 5: Accuracy with C = 0.3

C	Learning rate	Iteration	Validation accuracy
1	0.01	1000	0.97125
1	0.01	5000	0.965
1	0.01	10000	0.96
1	0.01	25000	0.95125
1	0.03	1000	0.965
1	0.03	5000	0.96
1	0.03	10000	0.9525
1	0.03	25000	0.95375
1	0.1	1000	0.96375
1	0.1	5000	0.9575
1	0.1	10000	0.95375
1	0.1	25000	0.94625
1	0.3	1000	0.96
1	0.3	5000	0.95875
1	0.3	10000	0.95625
1	0.3	25000	0.95
1	1	1000	0.96
1	1	5000	0.96125
1	1	10000	0.95875
1	1	25000	0.96125
1	3	1000	0.96
1	3	5000	0.965
1	3	10000	0.965
1	3	25000	0.96375

Figure 6: Accuracy with C = 1

C	Learning rate	Iteration	Validation accuracy
3	0.01	1000	0.96625
3	0.01	5000	0.96
3	0.01	10000	0.9525
3	0.01	25000	0.955
3	0.03	1000	0.96375
3	0.03	5000	0.95875
3	0.03	10000	0.95625
3	0.03	25000	0.95125
3	0.1	1000	0.9625
3	0.1	5000	0.95875
3	0.1	10000	0.95875
3	0.1	25000	0.95
3	0.3	1000	0.96125
3	0.3	5000	0.95875
3	0.3	10000	0.9575
3	0.3	25000	0.955
3	1	1000	0.95875
3	1	5000	0.96625
3	1	10000	0.95875
3	1	25000	0.96
3	3	1000	0.9575
3	3	5000	0.95625
3	3	10000	0.9625
3	3	25000	0.96

Figure 7: Accuracy with C = 3

C	Learning rate	Iteration	Validation accuracy
10	0.01	1000	0.96375
10	0.01	5000	0.95875
10	0.01	10000	0.955
10	0.01	25000	0.95125
10	0.03	1000	0.96375
10	0.03	5000	0.95875
10	0.03	10000	0.955
10	0.03	25000	0.95375
10	0.1	1000	0.95875
10	0.1	5000	0.96
10	0.1	10000	0.96125
10	0.1	25000	0.95875
10	0.3	1000	0.95875
10	0.3	5000	0.96
10	0.3	10000	0.95875
10	0.3	25000	0.96125
10	1	1000	0.95875
10	1	5000	0.9625
10	1	10000	0.9625
10	1	25000	0.95625
10	3	1000	0.9575
10	3	5000	0.9625
10	3	10000	0.95875
10	3	25000	0.96

Figure 8: Accuracy with C = 10

C	Learning rate	Iteration	Validation accuracy
30	0.01	1000	0.96375
30	0.01	5000	0.96
30	0.01	10000	0.955
30	0.01	25000	0.955
30	0.03	1000	0.96
30	0.03	5000	0.95875
30	0.03	10000	0.95875
30	0.03	25000	0.95875
30	0.1	1000	0.95875
30	0.1	5000	0.95875
30	0.1	10000	0.96
30	0.1	25000	0.9575
30	0.3	1000	0.9575
30	0.3	5000	0.96
30	0.3	10000	0.9575
30	0.3	25000	0.96
30	1	1000	0.95625
30	1	5000	0.96
30	1	10000	0.95625
30	1	25000	0.96
30	3	1000	0.95875
30	3	5000	0.9575
30	3	10000	0.9575
30	3	25000	0.96

Figure 9: Accuracy with C = 30

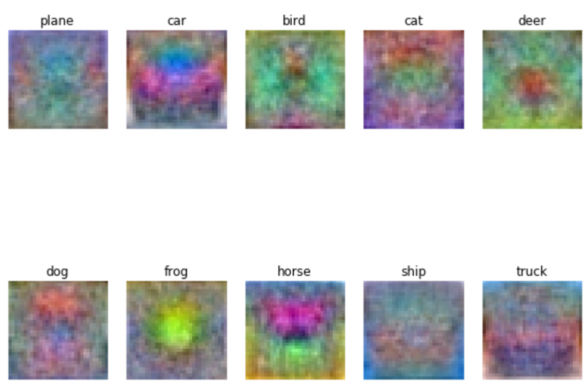

Figure 10: Visualization - SVM

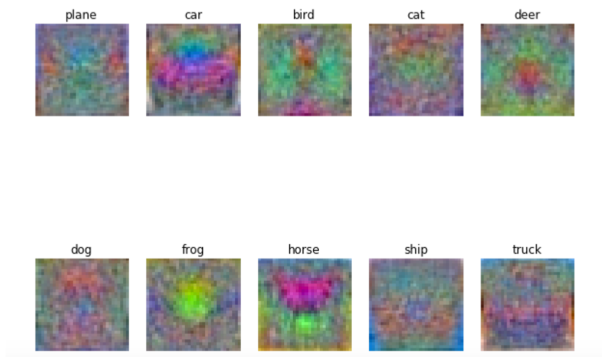

Figure 11: Visualization - Softmax

		SVM		Softmax	
Learning rate	Reg	Train accuracy	Test accuracy	Train accuracy	Test accuracy
0.0000001	50000	0.377673	0.383	0.1939	0.205
0.0000001	100000	0.364612	0.381	0.2216	0.227
0.0000005	50000	0.377531	0.384	0.391	0.407
0.0000005	100000	0.364714	0.38	0.4003	0.402
0.000001	50000	0.32049	0.34	0.3995	0.411
0.000001	100000	0.304102	0.324	0.4038	0.412

Figure 12: Comparison of train and test accuracy between softmax and svm at the same learning rate and reg