comp540 Homework4

Xiaoye Steven Sun (xs6), Wanyi Ye (wy13)

February 23rd 2018

1 Intuitions about support vector machines

1.1

Generally, a bigger margin can produce less over-fitting, since the margin will affect the regulization of weights.

1.2

Since the hinge loss is used for "maximum-margin" classification, we can see the definition below:

$$\ell(y) = \max(0, 1 + \max_{t \neq y} \mathbf{w}_t \mathbf{x} - \mathbf{w}_y \mathbf{x})$$

If the points which are not support vectors further away from the decision boundary, the output of the hinge loss will always be zero. Hence, moving them will not effect the hinge loss.

2 Fitting an SVM classifier by hand

2.1

Since

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

We have:

$$\phi(x^(1)) = [1, \sqrt{2} * 0, 0^2], \phi(x^(2))[1, \sqrt{2} * \sqrt{2}, \sqrt{2}^2] \tag{2}$$

The two points are:

$$[1, 0, 0], [1, 2, 2] \tag{3}$$

Since θ is perpendicular to the decision boundary between two points, it is actually parallel to the line that join these two points. Hence, a vector that is parallel to the optimal vector θ is:

$$[0, 2, 2] \tag{4}$$

2.2

We know that in 3 dimensions the 2 points are on 2 support vectors respectively, so the distance from each support vector to decision boundary is half of the distance between these 2 points. Hence the margin is the distance between these 2 points:

$$margin = \sqrt{(1-1)^2 + (2-0)^2 + (2-0)^2} = 2\sqrt{2} \tag{5}$$

2.3

Let the optimal $\theta = [\theta_1, \theta_2, \theta_3]$

$$margin = \frac{2}{||\theta||} \Rightarrow ||\theta|| = \frac{1}{\sqrt{2}} \Rightarrow (\theta_1^2 + \theta_2^2 + \theta_3^2) = \frac{1}{2} \tag{6}$$

From above we know that θ is parallel to [0, 2, 2]. Hence,

$$\theta_1 = 0, \theta_2 = \theta_3 \tag{7}$$

Let $\theta = [0, 2a, 2a]$

$$\begin{aligned} 0^2 * a^2 + 2^2 * a^2 + 2^2 * a^2 &= \frac{1}{2} \\ 8 * a^2 &= \frac{1}{2} \\ a^2 &= \frac{1}{16} \\ a &= \frac{1}{4} \end{aligned}$$

Hence, $\theta = [0, \frac{1}{2}, \frac{1}{2}]$

2.4

We have:

$$y_1(\theta^T \phi(x_1) + \theta_0) \geq 1 \tag{8}$$

$$y_2(\theta^T \phi(x_2) + \theta_0) \geq 1 \tag{9}$$

$$-1(0 * 1 + \frac{1}{2} * 0 + \frac{1}{2} * 0 + \theta_0) \geq 1 \Rightarrow \theta_0 \leq -1 \tag{10}$$

$$1(0 * 1 + \frac{1}{2} * 2 + \frac{1}{2} * 2 + \theta_0) \geq 1 \Rightarrow \theta_0 \geq -1 \tag{11}$$

Hence, $\theta_0 = -1$

2.5

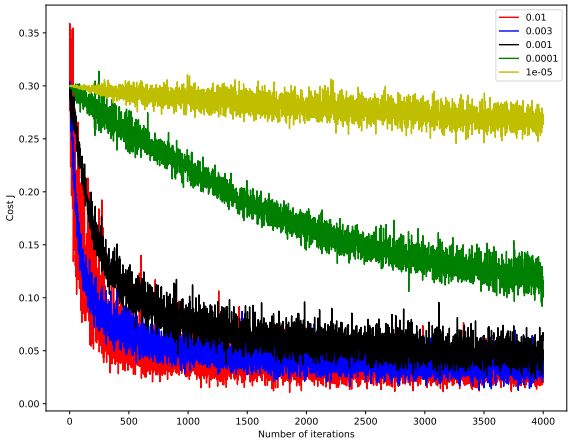
The decision boundary (or hyperplane) is defined by:

$$x^T \theta + \theta_0 = 0 \tag{12}$$

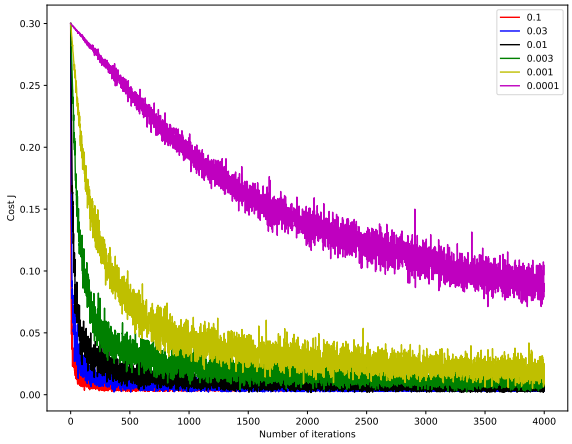
3 Support vector machines for binary classification

3.1 Support vector machines

hinge loss and gradient: Reflect on the gradient: The gradient of the hinge loss has two sections. The first section is for the case where $\max(0, 1 - y^{(i)} h_\theta(x^{(i)})) = 1 - y^{(i)} h_\theta(x^{(i)})$, the second section is for the case where the max function in the loss function equals 0. Intuitively, a θ_j only needs to be updated for the data samples which result in an incorrect classification.



(a) Gaussian kernel



(b) Linear Kernel

Figure 1: loss vs iteration with different learning rates

3.2 Selecting hyper parameter for SVMs

loss and gradient:

$$J(\theta) = \frac{\lambda}{2m}\sum_{l=0}^{d}\sum_{j=1}^{K}{\theta_l^{(j)}}^2 + \frac{1}{m}\sum_{i=1}^{m}\sum_{j\neq y^{(i)}}\max\left(0, {\theta^{(j)}}^T x^{(i)} - {\theta^{y^{(i)}}}^T x^{(i)} + \Delta\right) \tag{13}$$

$$\frac{\partial J(\theta)}{\partial \theta_l^{(j)}} = \begin{cases} \frac{\lambda}{m}\theta_l^{(j)} + \frac{1}{m}\sum_{i=1}^{m}\sum_{j'\neq y^{(i)}}(-x_l^{(i)}\mathbb{1}(j = y^{(i)}) + x_l^{(i)}\mathbb{1}(j \neq y^{(i)})) \text{ if } {\theta^{(j)}}^T x^{(i)} - {\theta^{y^{(i)}}}^T x^{(i)} + \Delta > 0 \\ \frac{\lambda}{m}\theta_l^{(j)} \end{cases} \tag{14}$$

gradient dimension does not match: The loss function is not always differentiable in all its domain since it is piecewise differentiable. When using the numerical method to compute the derivative, the generated random data point may be very close to the in-differentiable boundary so that the h is larger than the distance between the randomly generated point and the in-differentiable boundary. In this case, the gradient calculated using the numerical method is inaccurate. This is should NOT be a big concern since this is the drawback of the numerical method. Reducing h to a small enough value could make the numerical method calculate more accurate gradients.

3.3 Spam classification with SVMs

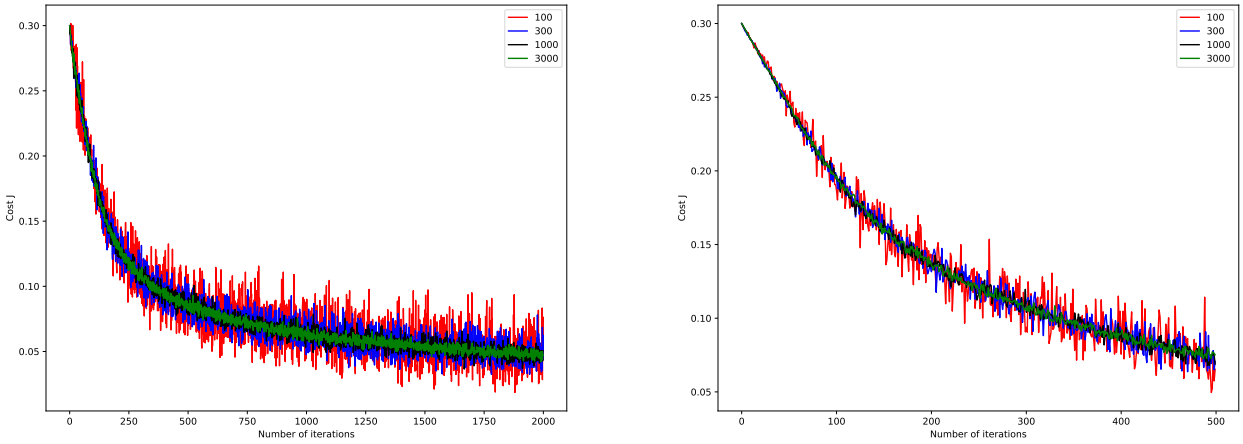
We try both linear kernel and Gaussian distance kernel. The file using linear kernel has file name "svm_spam_linear.*". The file using Gaussian kernel has file name "svm_spam.*". We found that Gaussian kernel performs slightly better than the linear kernel. The best hyper parameter in the Gaussian kernel is: LearningRate=0.001, C=30, BatchSize=300, NumberOfIterations=10000. The best accuracy on the testing data is 98.9%.

Scale data matrix: yes. the original data matrix (in linear kernel method) and the Gaussian distance matrix (in Gaussian kernel method) are normalized.

How to set learning rate: We choose the learning rate that is able to reach the minimum loss within reasonable number of iterations. See Figure 1. In addition, this learning rate should be small enough to avoid large fluctuation on the loss curve.

How to set batch size: We choose the smallest batch size that is able to reach the minimum loss with small fluctuation on the loss curve. See Figure 2.

How to set the number of iterations: We try different numbers of iterations and pick the best validation accuracy. (see the HTML file)



(a) Gaussian kernel

(b) Linear Kernel

Figure 2: loss vs iteration with different batch sizes

How to set the parameter C: We try different C values and pick the best validation accuracy. (see the HTML file)
What kernel is best: Linear is a good enough kernel to reach high classification accuracy. Gaussian performance a little bit better but the computation cost is much higher than the linear kernel.
How to select kernel hyper parameters: Only the Gaussian distance kernel has a hyper parameter σ. We try different σ values and pick the best validation accuracy.

4 Support vector machines for multi-class classification

final cell visualization: compare multiclass SVM and softmax regression:
training time: In terms of time complexity in getting the gradient, both are $\mathcal{O}(kmd)$. In practical, SVM takes longer time to train.
testing accuracy: multi-class SVM has lower test accuracy.
visualization results: the visualized figures looks similar.
hyper parameters: compare with our last homework, multi-class softmax needs lower learning rate and larger regularization strength.