IN 4320 Machine Learning Assignment 1

Xiangwei Shi
4614909

Some Optima & Some Geometry

1

a. According to exercise 1, r_- is fixed to 1, and there are only two observations for + class. Therefore, the loss function will be

$$L(1, r_+) := 0.5 * (\|-1 - r_+\|^2 + \|1 - r_+\|^2) + \lambda\|1 - r_+\|_1.$$

The figure of the loss function as a function of m_+ for all λ∈{0,1,2,3} is shown below.

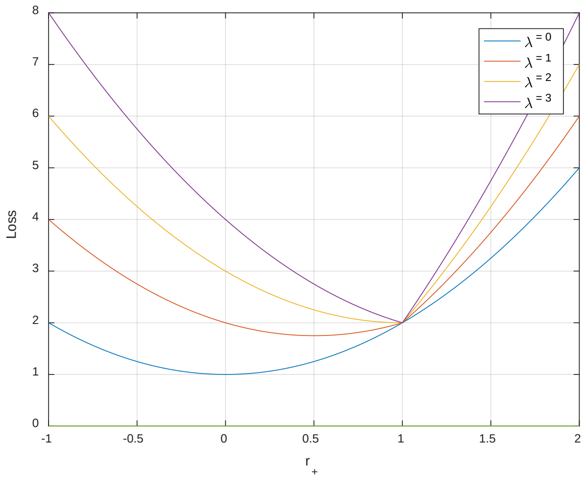

Figure 1 Loss function with different λ

b. The derivative of loss function will be

$$\frac{\mathrm{d}L}{\mathrm{d}r_+} = 2r_+ - \lambda((1 - r_+)/|1 - r_+|)$$

The plot of the derivative of loss function is shown below.

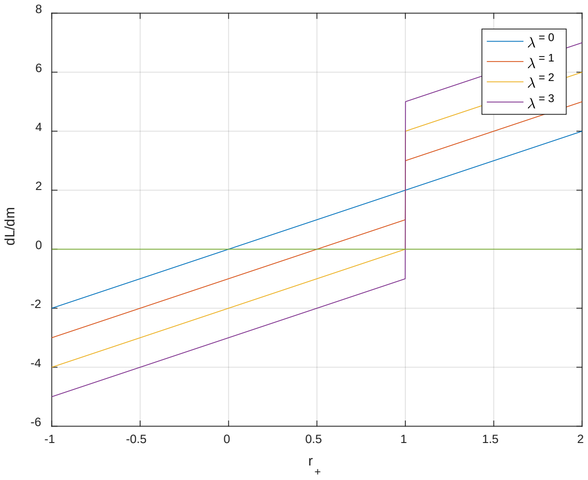

Figure 2 Derivative of loss function

From figure 2, we can see the minimizers for the four λ (0, 1, 2, 3) are 0, 0.5, 1 and 1. And from figure 1, we can get the minimum values are 1, 1.75, 2 and 2. As for the points where the derivative equals 0, when λ=0, the point is 0, and when λ=2, the point is 0.5. But when λ=4 or 6, there is no point where the derivative equals 0 from the derivative equation.

2

The regularizer in Equation (1) tries to enforce r_+ to get close to r_-. And when λ gets larger and larger, r_+ will equal to r_- eventually, which also means the loss function should be

$$L(r_-, r_+) := \sum_{i}^{N} \frac{1}{N_{y_i}} \left\| x_i - r_{y_i} \right\|^2$$

at the point where r_+ equals to r_-. And there is no derivative at $r_+ = r_-$, when λ gets larger and larger.

3

a. Since the both representors have to be determined through a minimization of the loss where $\mathrm{d} = 1$, the loss function can be formed by three parts, which is shown below,

$$L(r_-, r_+) := \sum_{i}^{N_+} \frac{1}{N_+} \|x_i - r_+\|^2 + \sum_{i}^{N_-} \frac{1}{N_-} \|x_i - r_-\|^2 + \lambda \|r_- - r_+\|_1.$$

In a 2-dimensional coordinate space, the two axes are representing r_+ and r_-. Considering the first two parts in the loss function *L*, the contour lines for this are concentric ellipse. And the contour lines for the last part of the loss function *L*, the regularizer, are parallel lines, one of which is the bisector line of the first and third quadrants. Therefore, the contour lines for the general function *L* can be described as the concatenation of an ellipse and a line. An example is listed below.

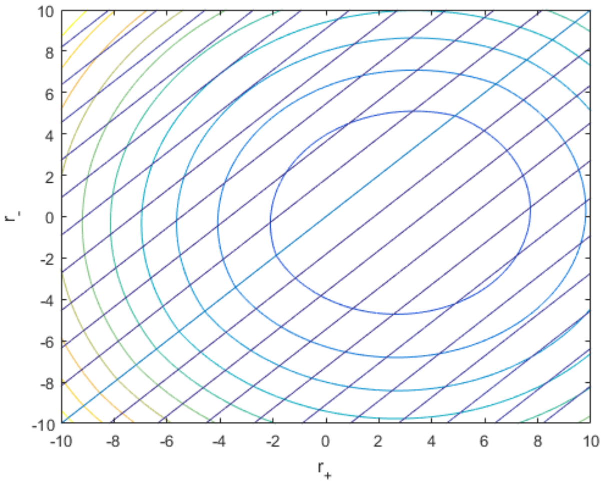

Figure 3 An example of contour lines for the loss function *L*

b. When λ is larger enough, according to 2, $r_+ = r_-$ and the loss function will be

$$\mathrm{L(r)} = (-1 - r)^2 + (1 - r)^2 + (3 - r)^2 + (-1 - r)^2$$

Therefore, the derivative of the loss function will be

$$\frac{\mathrm{d}L}{\mathrm{dr}} = 8r - 4$$

Let the derivative equal 0, we can get $r = 0.5$.

Therefore, the exact solution $(r_-,\ r_+)$ in this case is (0.5, 0.5).

Some Programming & Some Experimenting

4

a. I used gradient descent method to optimize. Gradient descent is a way to calculate the local minimum or maximum. In this case, I need to calculate the partial derivations of r_+ and r_-, which are

$$\frac{\partial L}{\partial r_+} = 2r_+ - 2\sum_{i=1}^{554}\frac{x_i}{554} - \lambda\frac{r_- - r_+}{|r_- - r_+|},$$

$$\frac{\partial L}{\partial r_-} = 2r_- - 2\sum_{i=1}^{571}\frac{x_i}{571} + \lambda\frac{r_- - r_+}{|r_- - r_+|}.$$

And by using the partial derivations, we can get the new r_+ and r_- which are closer than the previous r_+ and $r_{-,}$ using the equations below.

$$r_+ = r_+ - \alpha\frac{\partial \mathrm{L}}{\partial r_+}(r_+),$$

$$r_- = r_- - \alpha\frac{\partial \mathrm{L}}{\partial r_-}(r_-),$$

where α is the learning rate.

However, in terms of the derivations above, there are points where makes the loss function non-differentiable. In this scenario, subgradient method is applied to deal with the non-differentiable points. Before each iteration, the criterion of whether the point in this iteration is differentiable or not will be considered. If it is non-differential, the values of derivations from the previous iteration will be used to get a new value again. As for the stop criterion of how to decide a minimum, setting a threshold as the stop criterion is adopted in my optimization. When the difference between the values of loss function from two connecting iterations is smaller than this threshold, we consider the minimum has been found.

b. When λ = 0, the two solution images are shown below.

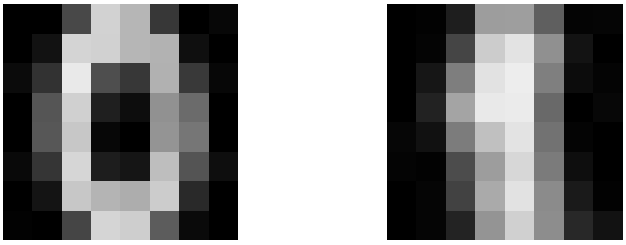

Figure 4 Two images of two classes when λ = 0

I set the learning rate (step length) as 0.0001 and the threshold for stop criterion was set as 2. Eventually, we can get the minimum loss in this case as 1.837e+08.

When λ is large enough, in theory, r_+ should equal $r_{-.}$ However, we cannot do that in Matlab. In this case, I set the learning rate as 0.0001, the stop threshold as 2000 and λ = 100000. At last, I get the minimum loss as 3.4422e+08. The two images in this case are shown below.

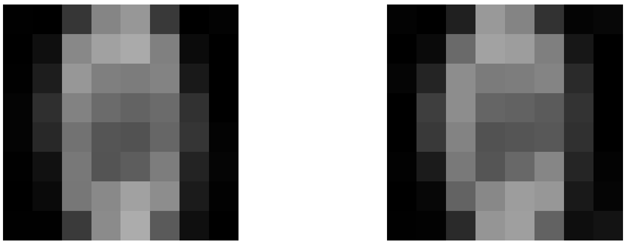

Figure 5Two images of two classes when λ =100000

c. In this scenario, since there is only one training example for each class, the apparent error is always 0 regardless of different values of λ. To get smooth curves, I repeated the experiment for 50 times. Each time a different training example was selected for both classes, and the rest images were used for test. To find the representors for each class, the gradient descent and subgradient descend methods were adopted like explained in 4a. The learning rate was set as 0.0001 and the stop criterion was set as $min(\exp(\lambda)+1,5)$. The regularization curves are shown below. The average true errors are 0.1464, 0.1463, 0.1443, 0.1426, 0.1356 and 0.1276.

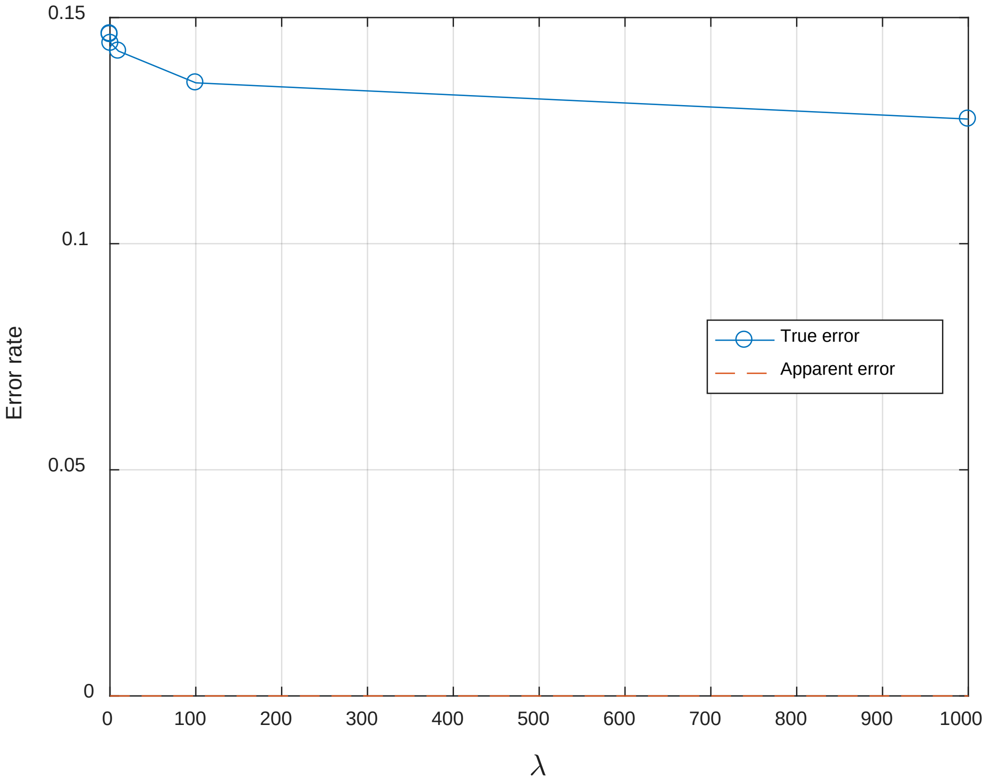

Figure 6 Regularization curves of the apparent and true error