IN4320 Machine Learning Exercise 1

Xiang Teng (4574060)

February 28, 2017

Question1

1. Figure 1 shows the drawing of the loss function as a function of m_+

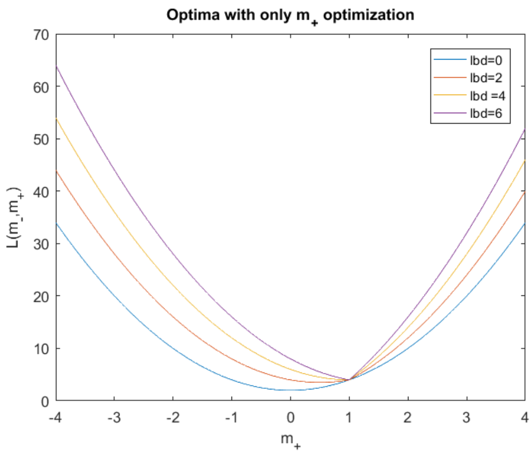


Figure 1: the drawing of the loss function as a function of m_+ for all $\lambda \in$ 0,2,4,6

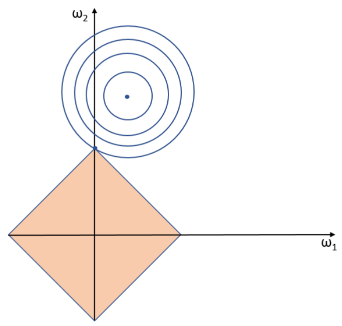
2. Derive the minimizer and their minimum values
   - The minimizer and their minimum values:
     (a) $\lambda = 0$, min = 2 with minimizer (0,2);
     (b) $\lambda = 2$, min = 3.5 with minimizer (0.5,3.5);
     (c) $\lambda = 4$, min = 4 with minimizer (1, 4);
     (d) $\lambda = 6$, min = 4 with minimizer (1, 4);
   - the points where their derivative equals to 0 (or around 0) are:
     (a) $\lambda = 0$, min = 2 with minimizer (0,2);
     (b) $\lambda = 2$, min = 3.5 with minimizer (0.5,3.5);
     (c) $\lambda = 4$, min = 4.0008 with minimizer (0.98, 4.0008);
     (d) $\lambda = 6$, min = 4.0408 with minimizer (0.98, 4.0408);

   The related gradients (around 0) are 0, 0, 0.0016 and 0.0416 respectively while λ has the value 0, 2, 4 or 6. This concludes that for both methods, The same minimum points are found.

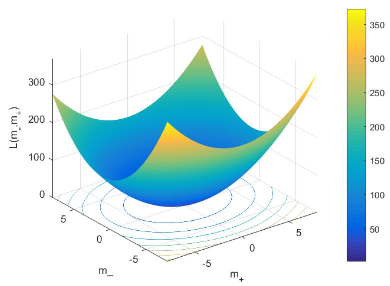
Question2

- a) The regularizer is trying to enforce the difference between two mean values. If λ gets larger and larger, the regularization term will have a dominant position in the loss function. The influence of the data-dependent error decrease. If λ is sufficiently large, some of the coefficients w_j are driven to zero, leading to a sparse model in which the corresponding basis functions play no role. Therefore, the problem of over-fitting is solved.

- b) This is the case of $q = 1$, it is known as *lasso* in the statistics literature. The unregularized error functions ($\lambda = 0$) has the shape of the circle. multiple circles have the different size but with the same central point. Each circle stands for the points which have the same value in 3-d space The regularized error function has the shape of the rotated rectangle. Figure 2a shows how the contours of these two basic geometric shapes look like.



(a) The contours of the unregularized error function (blue) along with the constraint region for the lasso regularizer $q = 1$



(b) A given 3d drawing after minimizing L for both $m\_$ and m_+ with $\lambda = 4$. The shape of the contours are the circles lay on the ground plane

Figure 2: Contours analysis for the regularized loss function

- c) If λ is sufficiently large, the corresponding basis functions play no role. The loss function now could be written as:

$$L(m\_, m_+) = \lambda ||m\_ - m_+||_1 \tag{1}$$

The minimum value is 3.2 and remains constant while λ is bigger than 4, the mean values are $m\_ = 0.6$ and $m_+ = 0.6$.

Question3

In this section, some progamming and experimenting are done by using the given data file. The loss function is

$$L(m_{\_}, m_{+}) = \sum_{i}^{N} ||X_i - m_{y_i}||^2 + \lambda ||m_{\_} - m_{+}||_1 \tag{2}$$

and its derivative respect to $m_{\_}$ and m_{+} are

$$\frac{\partial L(m_{\_}, m_{+})}{\partial m_{\_}} = 2 \sum_{i}^{N} (m_{\_} - x_i) + \lambda * sign(m_{\_} - m_{+}) \tag{3}$$

$$\frac{\partial L(m_{\_}, m_{+})}{\partial m_{+}} = 2 \sum_{i}^{N} (m_{+} - x_i) + \lambda * sign(m_{+} - m_{\_}) \tag{4}$$

respectively.

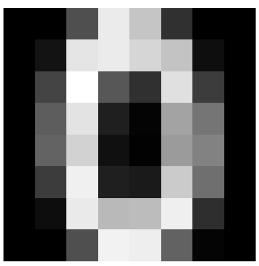
- a) Gradient descent is used as the search strategy. Firstly, a random mean value for both $m_{\_}$ and m_{+} are picked. And then the derivative of the loss function respect to both mean values are calculated. After that, an iteration process is executed based on the following equations:

$$m_{\_} = m_{\_} - \alpha * \frac{\partial F(m_{\_}, m_{+})}{\partial m_{\_}} \tag{5}$$

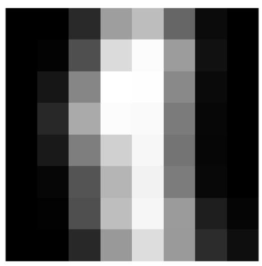
$$m_{+} = m_{+} - \alpha * \frac{\partial F(m_{\_}, m_{+})}{\partial m_{+}} \tag{6}$$

  with α is the learning rate. The iteration stops while the difference between $m_{\_}$ and m_{+} is smaller than a certain value.

- b) Figure 3 shows the mean images for both + and - class while $\lambda = 0$. And Figure 4 shows the images while λ is sufficiently large (the solution does not change anymore). In Figure 3, we can still recognize the digits in the images. However, It is already hard to distinguish the two pictures in 4 with $\lambda = 10^6$.

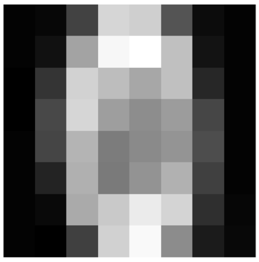


(a) Mean image for class -.

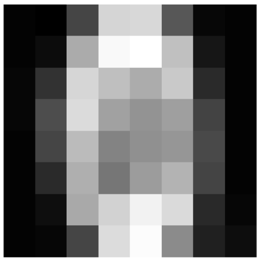


(b) Mean image for class +.

Figure 3: Pictures of mean values for $\lambda = 0$



(a) Mean image for class -.



(b) Mean image for class +.

Figure 4: Pictures of mean values a sufficiently large $\lambda = 10^6$

Source Code

- Function: gradientDescent()

```
for iter = 1:num_iters
    [dF_mn,dF_mp] = derivative(X,mn,mp,lambda);

    mn = mn - alpha*dF_mn;
    mp = mp - alpha*dF_mp;
end
```

- Function: derivative()

```
function [dF_mn,dF_mp] = derivative(X,mn,mp,lambda)

        dF_mn = 554*2*mn-sum(2*X((1:554),:))
        +lambda*sign(mn-mp);

        dF_mp = 571*2*mp-sum(2*X((555:end),:))
        +lambda*sign(mp-mn);

end
```

- Function: lossFunction()

```
function [cost] = lossFunction(mp,mn,X,lambda)
        cost1 = sum((X((1:554),:)-mp).^2);
        cost2 = sum((X((555:end),:)-mn).^2);
        cost3 = lambda*abs(mn-mp);
        cost = cost1+cost2+cost3;
end
```

- Question3

```
% initialize fitting parameters
initial_mn = zeros(1,64);
initial_mp = zeros(1,64);
lambda = [0 10^2 10^3 10^6];
for i = 1:1:4
% Some gradient descent settings
iterations = 300;
alpha = 0.0000001;
% run gradient descent
[mn,mp] = gradientDescent
(X,initial_mn,initial_mp,alpha,iterations,lambda(i));
fprintf('Mean found by gradient descent: \n');
% mp
img = reshape(mp,[8,8]);
img = transpose(img);
img = mat2gray(img);
figure
imshow(img,'InitialMagnification','fit');
% mn
img = reshape(mn,[8,8]);
img = transpose(img);
img = mat2gray(img);
figure
imshow(img,'InitialMagnification','fit');
end
```