IN4320 Machine Learning

Exercise 1

25 February 2019

TU Delft
Coordinated by Prof Marco Loog

Matthijs Bekendam 4725751 J.M.Bekendam@student.tudelft.nl

Contents

1 Question 1

The goal is to prove that the following function is convex:

$$L(m_-, m_+, a) := \Bigg(\sum_{c\in\{-,+\}} \sum_{i=1}^{N_c} ||x_c^i - m_c||_1 \Bigg) + \lambda||m_+ - m_- + a||_1. \tag{1.1}$$

Since the sum of convex functions preserves convexity, it would be convenient for our goal to examine the three norms inside the $L(m_-, m_+, a)$ separately for convexity. Making use of the fact that a 1-norms and affine functions are convex, we can establish that

$$\lambda||m_+ - m_- + a||_1$$

is convex. Next we will prove convexity on

$$\sum_{c\in\{-,+\}} \sum_{i=1}^{N_c} ||x_c^i - m_c||_1.$$

Again, the 1-norm is convex and $x_c^i - m_c$ is affine, hence convex. Now let, $x_c^i - m_c = a_i$ and $y_c^i - m_c = b_i$ and $c \in [0, 1]$. We have

$$f(ca + (1-c)b) = \sum_{i=1}^{N_c} |ca_i + (1-c)b_i|_1$$

$$|ca_i + (1-c)b_i|_1 \leq c|a_i|_1 + (1-c)|b_i|_1 \quad \text{(Triangle inequality)}$$

$$f(ca + (1-c)b) \leq \sum_{i=1}^{N_c} (c|a_i|_1 + (1-c)|b_i|_1)$$

$$f(ca + (1-c)b) \leq \sum_{i=1}^{N_c} c|a_i|_1 + (1-c)\sum_{i=1}^{N_c} |b_i|_1$$

$$f(ca + (1-c)b) \leq cf(a) + (1-c)f(b)$$

So f is convex. This can be generalized for both classes, hence using the above reasoning we can conclude that our objective function is also convex.

2 Question 2

To determine the corner points of the contour $\{(m_-, a)|L(0, m_-, a) = \pi\}$ with $\lambda = 1$, we first illustrate the contour of the regularization parameter.

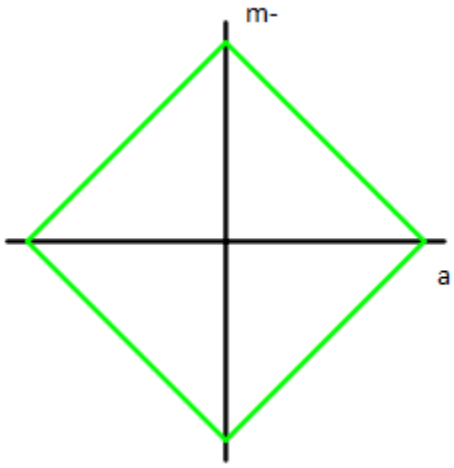


Figure 1: Contour of the regularization term $\lambda = 1$ and m_+ fixed to 0.

With the given observations the objective function reduces to

$$L(0, m_-, a) = \sum_{i=1}^{2} ||x_i^- - m_-||_1 + || - m_- + a||_1 = \pi$$

$$L(0, m_-, a) = ||1 - m_-||_1 + ||3 - m_-||_1 + || - m_- + a||_1 = \pi$$

Now we can insert the corner points, at which one of the variables $\{m_-, a\}$ will be guaranteed zero, which is the strength of using this specific regularization parameter. Corners of a diamond shape such as in Figure 1 are prone to match up first with an objective function.

Taking corner point (a,0) and inserting it into the reduced objective function yields

$$L(0, 0, a) = ||1||_1 + ||3||_1 + ||a||_1 = \pi$$

$$L(0, 0, a) = 4 + a = \pi$$

$$a = \pi - 4$$

Thus this corner point is $(\pi - 4, 0)$. The remaining three corner points can be computed in a similar fashion and can be viewed in table 1.

		Corner points
1	(a,0)	$(\pi - 4, 0)$
2	(-a,0)	$(4 - \pi, 0)$
3	$(0,m_-)$	$(0, \frac{\pi-4}{3})$
4	$(0,-m_-)$	$(0, \frac{4-\pi}{3})$

Table 1: Corner points of contour Figure 1

3 Question 3

In a general context a regularization term is added to the cost function to lower the complexity of the model and hence avoiding overfitting (model with high variance). An overfitted model performs well on training data but fails to generalize. A regularization term penalizes outliers in the data, adding a small amount of bias but decreasing the variance. As a result, the optimal value of the objective function will be higher with an added regularization term, but the model improves. For this regularization term, a is added to ensure that for increasing lambda's, the medians are not pulled towards each other (which would have need to happen to compensate for the larger lambda value). This specific regularization term tries to enforce distance between the two medians.

For sparse solutions, one or multiple coefficients will be zero. This will not happen for this particular regularization term. Making m_+ or m_- zero would counter productive with respect to minimizing the objective function since you subtract the medians from data values. Since a is a scalar and m_+ and m_- are vectors, a is too restricted to become zero.

The optimal value of a can be visualized by plotting a as a function of $||m_+ - m_- + a||_1$ with fixed m_+ and m_-.

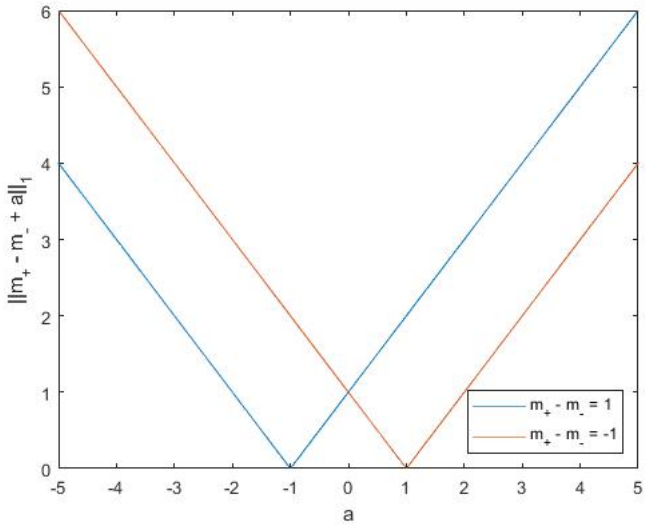


Figure 2: Parameter a as a function of $||m_+ - m_- + a||_1$ with fixed m_+ and m_-, optimal values of a are those for which $a = -m_+ + m_-$.

Figure 2 shows that the optimal value of $a = -m_+ + m_-$, it is the point for which the function $||m_+ - m_- + a||_1$ is zero.

4 Question 4

To minimize the objective function the Gradient Descent (GD) optimization strategy was chosen (illustrated in Figure 3). GD is easy to implement (convenient for debugging), computational efficient and has a stable convergence. Since the objective function is convex, all local minima are also global minima, so in this case gradient descent can converge to the global solution. Whereas for non convex functions it is dependent on the initial conditions used if a global minima is found.

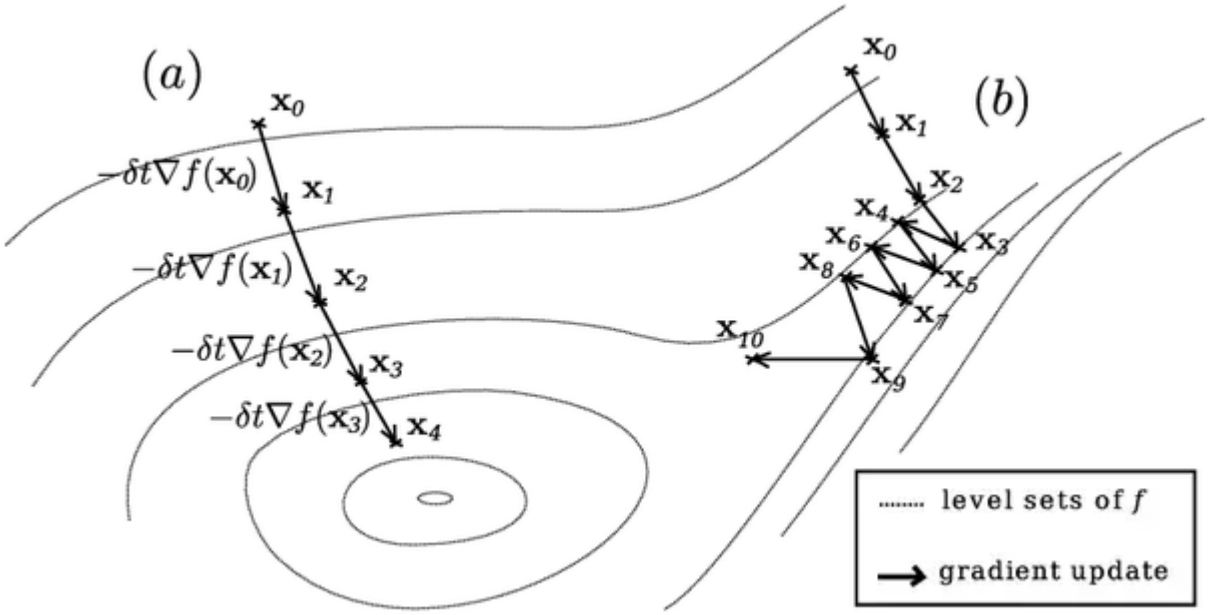


Figure 3: Iterative minimization process of the GD algorithm. Each iteration GD looks for the steepest slope, which is the vector orthogonal to the level curve.

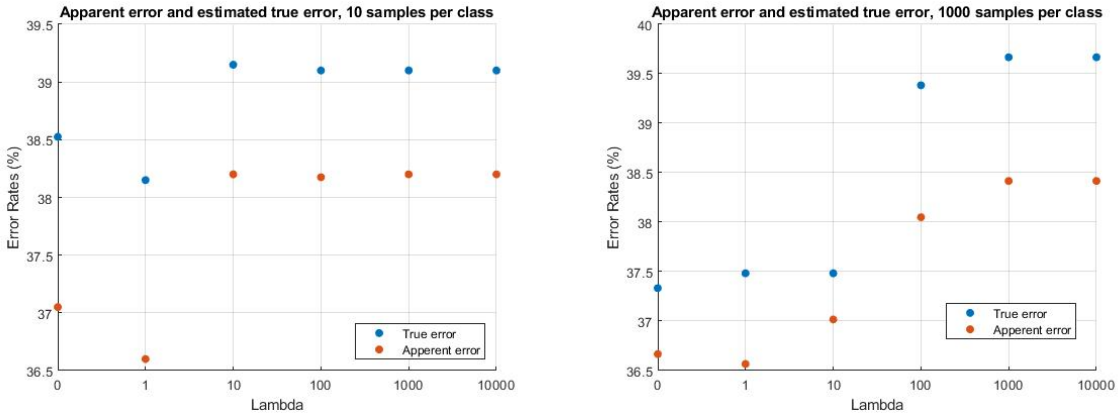
Through an iterative procedure the minimum is searched by taking steps proportional to the negative of the gradient. Note that the negative gradient at a point is orthogonal to the level set. A minima is reached when the GD does not decrease the cost-function anymore (or just constantly by a small amount). The number of iterations needed for convergence is difficult to estimate beforehand.

Regarding the NmC, one of the properties of a geometric median is that it has minimal distance to all other points in the set. Hence it is logical for our objective function to minimize the sum of distances with $\{m_+, m_-, a\}$. The optimal parameters $\{\hat{m}_+, \hat{m}_-\}$ will each be a [21x1] vector which can be used to classify new data. The reason for including a as an optimization parameter was explained in Question 3.

5 Question 5

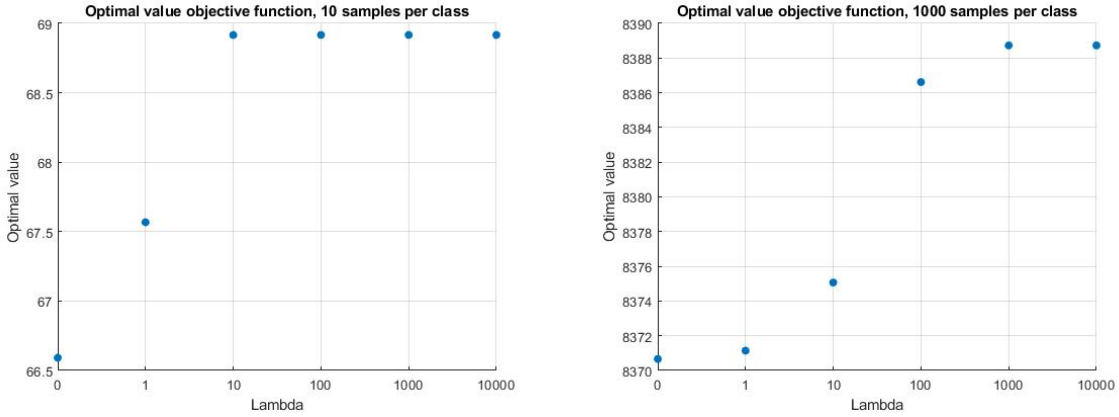
In this section the classification results for training set sizes of 10 and 1000 samples per class will be discussed.

The effect of different sizes of lambda's on the classification problem were tested and are illustrated in Figure 4 and Figure 5, which denote the errors and optimal objective values found in training time respectively. Both instances show a higher true error compared to the apparent error, which makes sense since the apparent error is generated through the same data which the classifier has been trained on.



(a) (b)

Figure 4: Apparent- and true errors vs. lambda's.



(a) (b)

Figure 5: Optimal values of objective function found in training time vs. lambda's

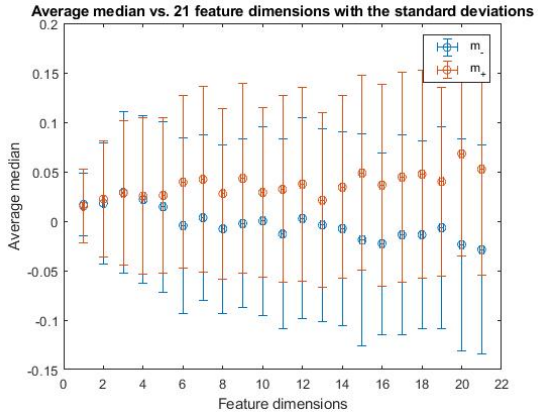
5.1 10 samples per class

Since the data contains 21 features, taking 10 samples to train the classifier requires multiple iterations to stabilize the model. As expected, increasing lambda will first decrease the error, helping generalize the model. Increasing lambda further will increase the error, under fitting the model. Figure 5a shows a fast change in optimal value when the regularization constraint is introduced. Increasing the size of the contour/lambda (Figure 1) does not effect the optimal value after $\lambda = 10$, thus the error stays relatively constant.

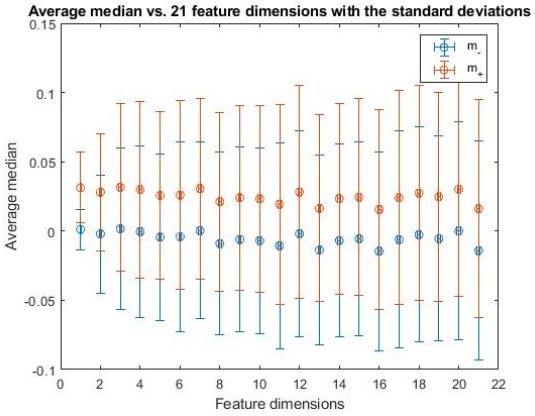
5.2 1000 samples per class

Taking 1000 samples per class for 21 feature is quite a lot of data to train the classifier so it already generates a fairly stable model (3 iterations was sufficient). Adding a regularization term will only add to the bias. Figure 5b and Figure 5a shows that the objective function and errors are less effected by an increase in lambda compared to the 10-class case.

6 Question 6

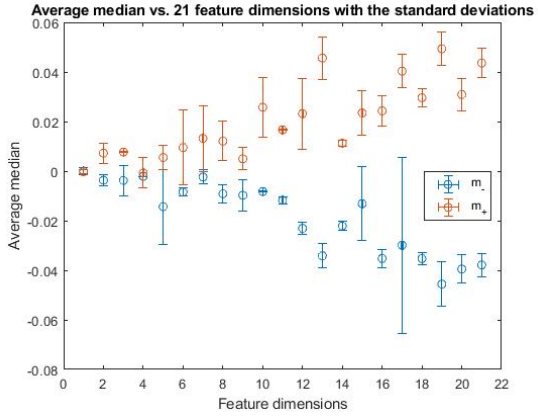


(a) Training set sizes of 10, lambda equals 0.

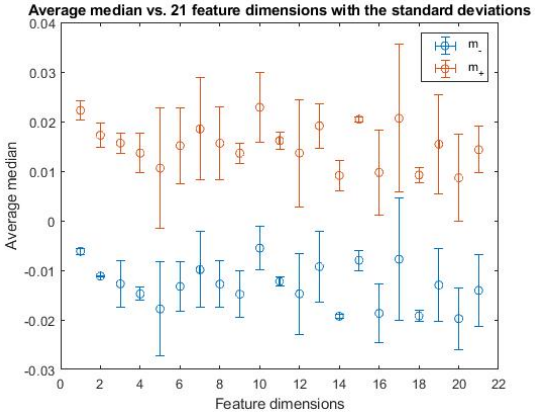


(b) Training set sizes of 10, lambda equals 10000.

Figure 6: Average medians vs. 21 feature dimensions.



(a) Training set sizes of 1000, lambda equals 0.



(b) Training set sizes of 1000, lambda equals 10000.

Figure 7: Average medians vs. 21 feature dimensions.

Figure 6 and Figure 7 reveal the effect of lambda on the median of different training sample sizes. Taking a small sample size will give a larger variance compared to the higher sample size. A large variance means more overlap of the two medians, which makes classification less consistent. Increasing lambda too far as in Figure 7a and Figure 7b destroys the general structure of the medians, under fitting the model. Lambda in general does decreases the variance, which helps the consistency of the model.