IN4320 Machine Learning — Assignment 1

Francisca Mestre - 4930142

February 2019

1

The p-norm function $\|.\|_p$ has the following properties for $p \geq 1$:

- $\|h\|_p \geq 0$,
- $\|h\|_p = 0 \iff u = 0$,
- $\|\alpha h\|_p = |\alpha| \|h\|_p$ for a scalar α,
- $\|h + g\|_p \leq \|h\| + \|g\|$.

It is then easy to verify that for $c \in [0, 1]$ the following holds:

$$\|ch + (1 - c)g\|_p \leq c\|h\|_p + (1 - c)\|g\|_p.$$

Then the p-norm function is convex in its argument for $p \geq 1$ and $\|.\|_1$ is convex.
Any affine function is convex.
The composition $g\big(f(x)\big)$ of functions f and g is convex if f is convex and g is convex and non-decreasing. Therefore, the composition of the 1-norm of an affine function is convex, i.e. $\|x_i^c - m_c\|_1$ and $\|m_+ - m_- + a\|_1$ are convex.
Finally, the objective function consists of a positive-weighted sum of convex terms, with $\lambda \geq 0$, therefore it is also convex.

2

For the one dimensional case $d = 1$ we have $m_+, m_-, a \in \mathbb{R}$. Besides, $N_+ = 1$ and $N_- = 1$, $\lambda = 1$ and m_+ fixed to 0. We can rewrite the objective function in the following way:

$$\begin{aligned} L &= \sum_{i=1}^{N_-} \|x_i^- - m_-\|_1 + \sum_{i=1}^{N_+} \|x_i^+ - m_+\|_1 + \|a - m_-\|_1 \\ &= \|x_1^- - m_-\|_1 + \|x_2^- - m_-\|_1 + \|x_1^+ - m_+\|_1 + \|a - m_-\|_1 \\ &= |x_1^- - m_-| + |x_2^- - m_-| + |x_1^+| + |a - m_-|. \end{aligned} \tag{1}$$

With $x_1^+ = 0$, $x_1^- = 1$ and $x_2^- = 3$ we get

$$L = |1 - m_-| + |3 - m_-| + |a - m_-|. \tag{2}$$

The objective function is piecewise affine and it's behaviour changes for the values of $m_- \in \{1, 3, a\}$, i.e. for the values of m_- that make each of the absolute value terms equal to zero. Take the case with $m_- = a$. The contour $\{(m_-, a)|L(0, m_-, a) = \pi\}$ now becomes

$$|1 - a| + |3 - a| = \pi.$$

For $a < 1$ we get

$$1 - a + 3 - a = \pi \iff 2a = 4 - \pi$$
$$a = \frac{1}{2}(4 - \pi) = 0.4292.$$

The corner point is thus (0.4292, 0.4292). In a similar manner we obtain the remaining corner points of the problem: (1, -0.1416), (1, 2.1416), (3, 1.8584), (3, 4.1416), (3.5708, 3.5708).

3

The term $\|m_+ - m_- + a\|_1$ introduces a penalty in order to force the medians m_+ and m_- in each dimension to be close to each other. The hyperparameter λ controls the strength of the penalty. Because we are working with the lasso we get sparser solutions.
For fixed $m_+ = 2$ and $m_- = 1$ we get that the optimal value of a is -1 as can be seen in Figure 1.

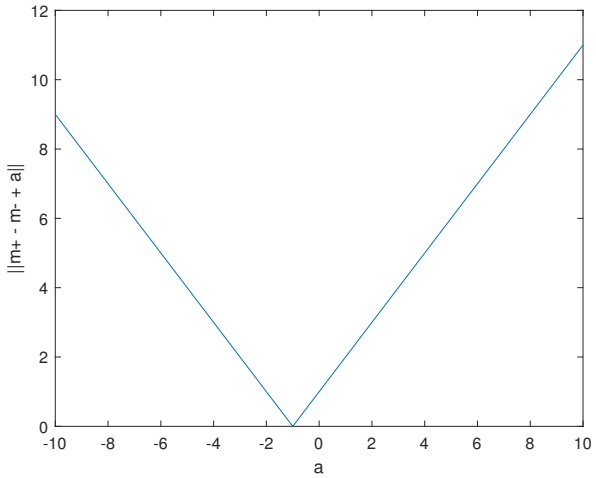


Figure 1: Optimal value of a

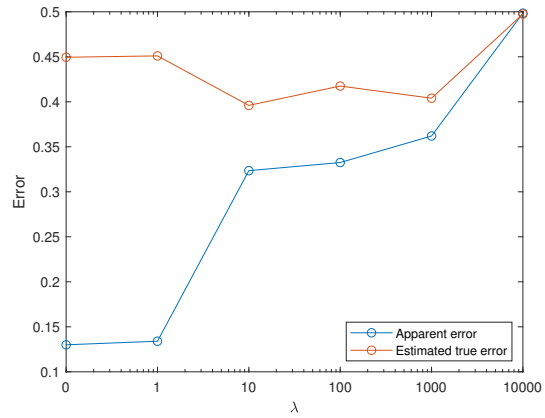
4

In order to solve the optimization problem an optimizer was implemented using MATLAB. The objective function was translated into MATLAB through function handles and the problem is unconstrained with penalties on the weights, thus the objective function will be minimized using the pre-defined function `fminunc`. The optimization variable is a vector where the first N_+ entries correspond to the vector m_+, the following N_- entries correspond to the vector m_- and the last entry is the scalar a.
The optimization routine adopted uses a gradient-based search performed by `fminunc`. This means that starting from some initial point (vector) the algorithm will find it's way to a close local minimum by following the gradient of the objective function backwards. This is because the gradient provides the direction in which the increase in the function value is the steepest on a certain point, and since our aim is to improve (decrease) the function value we follow the opposite direction to that. Since the objective function is convex the algorithm will converge to a global minimum.
The minimum is reached when the size of the gradient is less than 1×10^{-6} which is the default value of the function. As it is an appropriate value it was not modified. The method is illustrated in Figure 2.

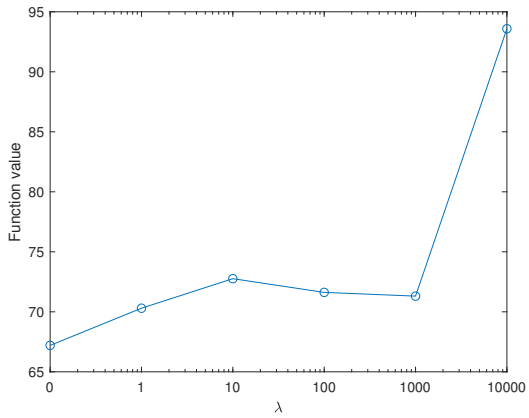
Figure 2: Gradient-based search

5

The following images show the true error and apparent error for different values of regularization as well as for the optimal value of the objective function found during training. These results are presented for different training set sizes of 10 samples and 1000 samples.

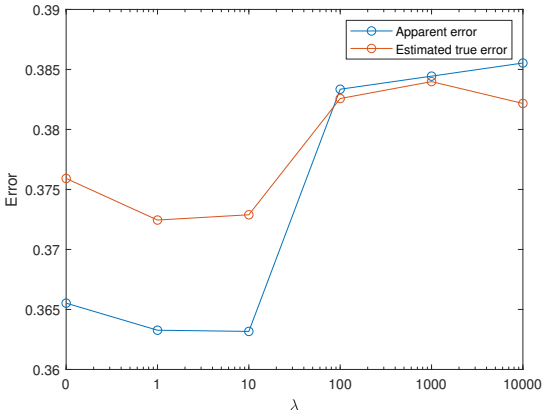


(a) Errors

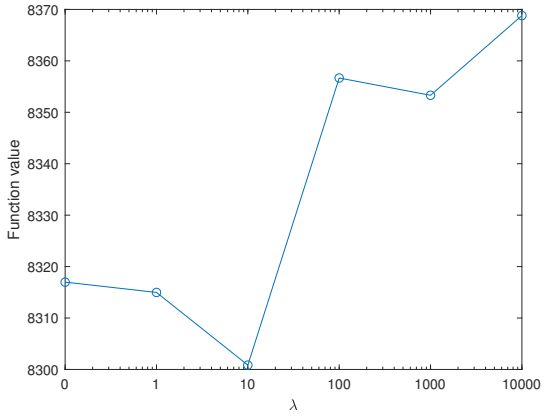


(b) Optimal function value

Figure 3: Training set of 10 samples



(a) Errors

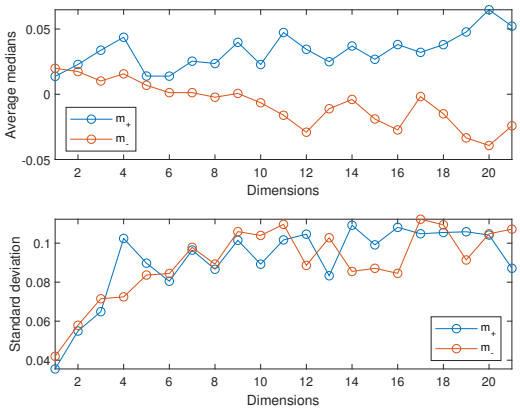


(b) Optimal function value

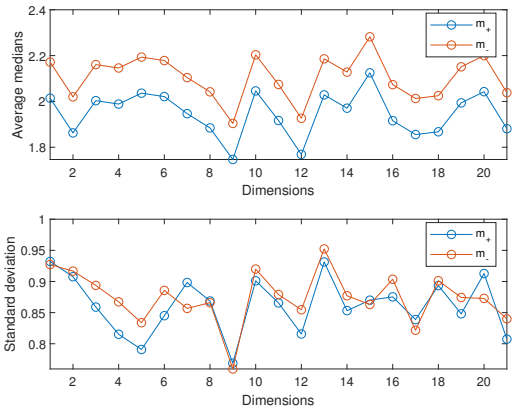
Figure 4: Training set of 1000 samples

6

The next images show the average medians and standard deviations, again for different training set sizes of 10 samples and 1000 samples and for $\lambda = \{0, 10000\}$.

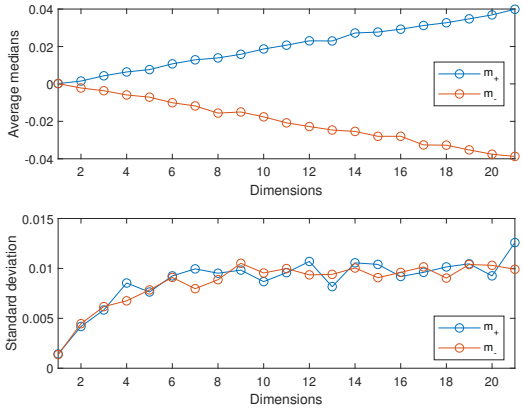


(a) $\lambda = 0$

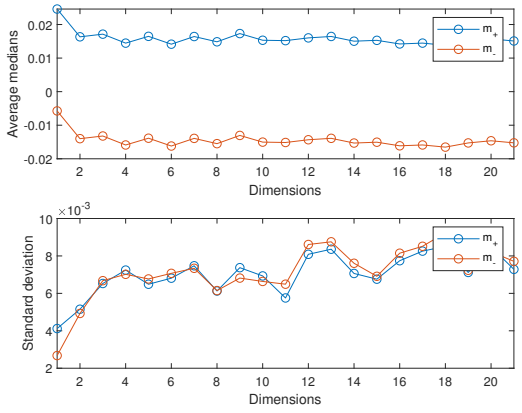


(b) $\lambda = 10000$

Figure 5: Training set of 10 samples



(a) $\lambda = 0$



(b) $\lambda = 10000$

Figure 6: Training set of 1000 samples