CSE 446: Machine Learning
Winter 2018

Assignment 3

from
Lukas Nies
University of Washington

02/22/18

Contents

0 Policies

0.1 List of Collaborators

My collaborator was Edith Heiter (discussed Problem 2 and 4). The development of the answers though was completely independent and individually.

0.2 List of Acknowledgments

None.

0.3 Policies

I have read and understood these policies.

1 Problem: Linear Regression on MNIST

1.1 Closed Form Estimator

1. If one runs the Closed Form Estimator with $\lambda = 0$ one encounters trying to invert a singular matrix $(X^T X)$ which is not possible per definition since the determinant is $\det(X^T X) = 0$. The matrix is therefore not invertible. To avoid this we introduce a regularization by adding the term $\lambda \mathbb{1}_d$. This is intuitively clear by considering the data itself: one digit consists of 28×28 pixels where most pixels (at the edges and in the corners) don't carry any information about the digit itself. When calculating $X^T X$ we get the same result: we have more "dimensions" than information for those "dimensions". In mathematical terms: $X^T X$ is underdetermined.

2. For this part a grid search was implemented to search for different values of λ and the threshold to optimize the performance on the development set:

   (a) The best result was found with $\lambda = 101$ and a threshold of 0.4. The grid search ran for λ from 1 to 250, the treshold ran from 0.1 to 1.0.

   (b) The average squared error using the parameters stated above is as follows:
   - Training error = 0.09165
   - Development error = 0.01907
   - Test error = 0.02132

   (c) The misclassification error using the parameters stated above is as follows:
   - Training error = 1.88%
   - Development error = 1.64%
   - Test error = 2.30%

3. Samples with large values (far off the mean of the rest of the data points) have a strong influence on linear polynomial functions fitted through regression. This leads to large misclassification on most of the data points.

1.2 Linear regression using gradient descent

1. The proof is as follows:

$$
\begin{aligned}
\frac{\partial \mathcal{L}_w}{\partial w} &= \frac{\partial}{\partial w}\left(\frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\left(y_n - w^T x_n\right)^2 + \frac{\lambda}{2}\|w\|^2\right) \\
&= \frac{1}{N}\sum_{n=1}^{N}\left(-\frac{2x_n}{2}\right)\left(y_n - w^T x_n\right) + \left(\frac{2\lambda}{2}\mathbf{w}\right) \\
&= -\frac{1}{N}\sum_{n=1}^{N}\left(y_n - \hat{y}_n\right) x_n + \lambda \mathbf{w}
\end{aligned}
$$

2. We can rewrite this as a matrix expression:

$$\frac{\partial \mathcal{L}_w}{\partial w} = -\frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)\,x_n + \lambda\mathbf{w} = -\frac{1}{N}X^T \cdot \left(Y - \hat{Y}\right) + \lambda\mathbf{w}$$

3. Stepsizes $10^{-3} \leq \eta \leq 10^{-2}$ worked well for this problem. For the error rate see figure 1. For generating the plots, $\lambda = 1$ and $\eta = 10^{-2}$ were chosen.

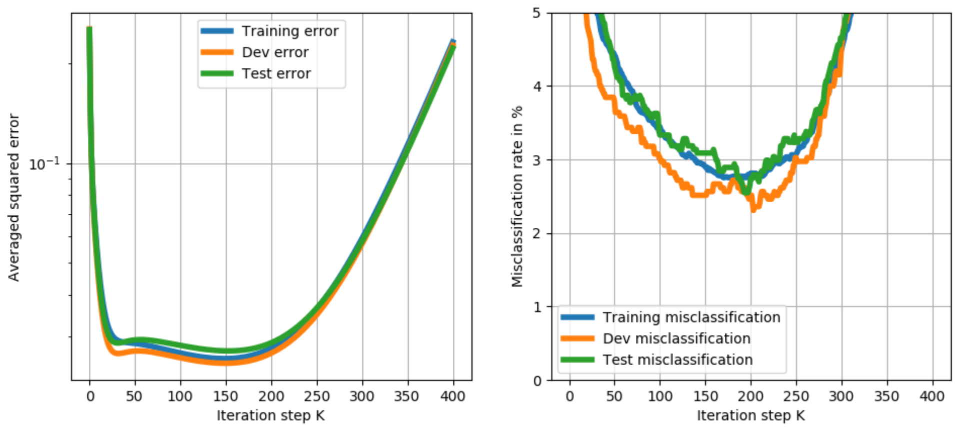

Figure 1: Plot of averaged squared errors (left, note the logarithmic vertical axis) and misclassification loss in percent (right). For generating the plots, $\lambda = 1$ and $\eta = 10^{-2}$ were chosen.

1.3 Linear Regression Using Stochastic Gradient Descent

References