Data 100: Discussion 6

l_1 Loss, Log Transformations, and Regression

Chris Pyles

February 28, 2019

Announcements

- HW 4 due Monday Mar 2 at 11:59 PM
- Lab 4 due Monday Mar 2 at 11:59 PM
- Keep an eye out for a midterm logistics and review materials post to come out this weekend.

l_1 Loss

l_1 Loss

Recall from HW 1 that the minimizing $\hat{\theta}$ of the l_2 loss is

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} (x_i - \theta)^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

This shows that the average squared loss for the constant estimator is the sample variance.

l_1 Loss

Today's discussion focuses on the l_1 loss function:

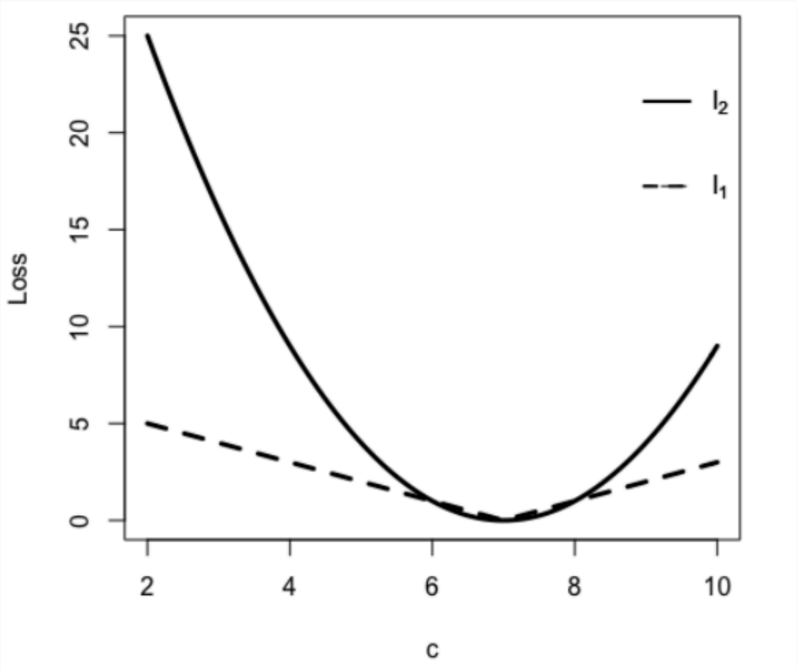
$$l_1(\theta, y) = |y - \theta|$$

We can compute the average loss as

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} |x_i - \theta|$$

l_1 Loss

Why l_1 loss? Because it is more robust to outliers. Notice in the plot below that the loss rises more steeply in l_2 the further away we get from their minima.



l_1 Loss

Should we always use l_1? What about l_2? What might be an advantage of l_2 over l_1?

l_1 Loss

Should we always use l_1? What about l_2? What might be an advantage of l_2 over l_1?

It depends on your data. l_2 is an easily differentiable function (since there's no absolute value) but l_1 is more robust to outliers.

l_1 Loss

How do we find the optimizing θ when loss is not easily optimizable? Use a numerical analysis method: *gradient descent*.

Gradient Descent Algorithm

$\theta^{(0)}$ = some vector
for τ in 0...convergence:
$$\theta^{(\tau)} = \theta^{(\tau-1)} + \rho(\tau - 1)\left(\nabla_\theta \mathbf{L}(\theta) \mid_{\theta^{(\tau-1)}}\right)$$

$\rho(\tau)$ is the *learning rate* or *step size* of the algorithm, usually a constant or $\frac{1}{\tau+1}$.

l_1 Loss

Our gradient descent algorithm requires computing $\mathbf{L}(\theta)$ at each iteration which very expensive in terms of space and time. An alternative method is *stochastic gradient descent*, in which we compute $\nabla_\theta l(x, \theta)$ for some random chosen sample point x at each iteration:

Stochastic Gradient Descent Algorithm

$\theta^{(0)}$ = some vector
for τ in 0...convergence:
    $\mathcal{B}$ = random subset of indices
    $\theta^{(\tau)} = \theta^{(\tau-1)} + \rho(\tau-1)\left(\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \nabla_\theta \mathbf{L}_i(\theta) \mid_{\theta^{(\tau-1)}}\right)$

where $\mathbf{L}_i(\theta) = \mathbf{L}(\theta, x_i, y_i)$.

Log Transformations

Log Transformations

Consider plotting two data points, 1 and 1000, on a paper number line with $\Delta x = 1$cm (that is, a change of 1 in x is represented by a 1cm change on the number line). How long would the number line need to be to hold both data points?

Log Transformations

$$\begin{aligned}\Delta x &= 1000 - 1 \\ &= 999 \\ &= 999\frac{1\text{cm}}{1} \\ &= 999\text{cm} \\ &= 9.99\text{m}\end{aligned}$$

So, you would need a 10m long piece of paper to plot both points!

Log Transformations

If, however, you were to plot 1 and $\log_{10} 1000 = 3$, you would only need 3cm.

Log scaling makes better use of space on axes, and can be used to "linearize" relationships between variables. The trade-off of log scaling is that we distort the axis scale.

Logs also capture relative shocks to values: consider the relative impact of $5,000 on someone making $20,000 per year as opposed to $30,000,000. A logarithmic transformation captures the comparative scale better than the raw values.

Worksheet