Newton vs Natural gradient for least squares

Yaroslav Bulatov

April 12, 2017

Suppose we have n datapoints, l input features and k output features. We can make a linear predictor

$$Y = WX$$

Where W has shape k, l, X has shape l, n and Y has shape k, n. Given set of desired labels $\hat{Y}$, we introduce error matrix $e = \hat{Y} - Y$ and denote our least squares loss as follows

$$J = \frac{1}{2}\text{tr}(e'e)$$

To derive the gradient of loss, use technique of differentials (Magnus, Nuedecker) – www.janmagnus.nl/misc/mdc2007-3rdedition

$$dJ = \text{tr}(e'de) = -\text{tr}(Xe'dW)$$

From which it follows that gradient G is written as

$$G = -eX'$$

Our gradient descent update on matrix W with learning rate α can be written as follows

$$W^* = W - \alpha G$$

$$W^* = W + eX'$$

1 Newton

To precondition gradient descent step with inverse Hessian, we write in in vectorized version. Use lower case version to refer to vectorized versions $\text{vec}(W) = w$, $\text{vec}(G) = g$

Our preconditioned gradient descent step is

$$w^* = w - \alpha H^{-1} g$$

To obtain H we apply differential operation to dJ and get

$$d^2J = \text{tr}(XX'dW'dW)$$

From this, we can extract the Hessian as follows (following Theorem 1 in 10.6 of Magnus/Nuedecker)

$$H = (XX') \otimes I_k$$

Here I_k refers to identity matrix of size k and $\otimes$ is Kronecker product. Note that when $k = 1$, this reduces to XX'

We can use connection between kronecker product and vectorization to write preconditioning step as follows.

$$H^{-1}g = \text{vec}(G(XX')^{-1})$$

This lets us write preconditioned gradient update in matrix form as

$$W^* = W - \alpha G(XX')^{-1}$$

Note, that k is arbitrary

2 Natural Gradient

For natural gradient we take original dataset and sample new labels from the predictive distribution. Let Y and X refer to this resampled dataset here.

A gradient from a single example g_i can be written as

$$g_i = -e_i x_i'$$

Here e_i corresponds to prediction error on example i To compute covariance matrix

$$C = \frac{1}{n}\sum_i g_i' g_i = \frac{1}{n}\sum_i e_i x_i' x_i e_i'$$

Suppose k is 1 (ie, we have a single predictor per datapoint). Then we can write C as follows:

$$C = \frac{1}{n}\hat{X}\hat{X}'$$

Where $\hat{X}$ is the data matrix where each column is weighed by corresponding error. More precisely

$$\hat{X} = X\text{diag}(e)$$

since k is 1, our e matrix has dimension $1, n$ and we can turn it into diagonal matrix of dimension n, n by arranging the errrors on the diagonal.

3 Conclusion

Hessian is a covariance matrix of data, whereas fisher is a covariance matrix of data weighted by errors of datapoints.