OLS

Alexandre Piche
260478404
Department of Mathematics and Statistics
McGill University
Montreal, Quebec
Email: alexandre.piche@mail.mcgill.ca

Philippe Nguyen
260482336
Springfield, USA
Email: homer@thesimpsons.com

Yash Lakhani
Starfleet Academy
San Francisco, California 96678–2391
Telephone: (800) 555–1212
Fax: (888) 555–1212

***Abstract*—Implementation of linear regression using the closed form and the gradient descent solutions. Incorporate the ridge regularization from scratch and used the lasso implementation from scikit-learn [1].**

I. Introduction

Website popularity prediction is important because ...

Simple tools like OLS have a surprising power, particularly when couple with regularization techniques such as the lasso or ridge.

II. Implementation of OLS

$$Y = X\beta + \epsilon \tag{1}$$

A. Closed Form

With the traditional assumption of $X^T\epsilon = 0$ [2], i.e. that the error is uncorrelated with the matrix X, it is easy to solve for the weights, the resulting equation is given by

$$\begin{aligned} Y &= X\beta + \epsilon && (2)\\ X^TY &= X^TX\beta + X^T\epsilon && (3)\\ \hat{\beta} &= (X^TX)^{-1}X^TY && (4) \end{aligned}$$

B. Gradient Descent

It is computationally inefficient to invert large matrices such as the one provided for this exercise. It is more efficient to minimize the sum of squares $SSR(\beta) = \sum_{i=1}^{n}(Y - X\beta)^2$. We need to take the derivative to

$$\frac{\partial SSR(\beta)}{\partial \beta} = -2X^T(Y - X\beta) \tag{5}$$

cite Joelle's slides lecture 2

while $\epsilon > 0.1$ and $i <$ max_iterations **do**
  hypothesis $\leftarrow X^T\beta$
  loss $\leftarrow$ hypothesis - Y
  gradient $\leftarrow 2\ X^T$ loss
  $\beta_{\text{new}} \leftarrow \beta - \frac{\alpha * \text{gradient}}{n}$
  $\epsilon \leftarrow \beta_{\text{new}} - \beta$
  $i \leftarrow i + 1$
  $\beta_{\text{new}} \leftarrow \beta$
end while

III. Lasso and Ridge Regularization

Talk about variance vs bias

Penalizing decrease the variance but increase the bias. **[?]**

To be able to generalize well to new data, we want to avoid over fitting. To do so we will penalize extreme weights for our β

Talk about Occam's razor

A. Ridge or L2-Regularization

$$\hat{\beta}^{\text{ridge}} = \overset{\text{argmin}}{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{6}$$

[3]

The gradient will then be

$$\frac{\partial SSR(\hat{\beta}^{\text{ridge}})}{\partial \hat{\beta}^{\text{ridge}}} = -2X^T(Y - X\beta) + 2\lambda\|\beta\| \tag{7}$$

We can then add the following condition to our gradient descent algorithm

if Regularization = 'Ridge' **then**
  loss $+= 2 * \lambda\|\beta\|$
end if

B. Lasso or L1-Regularization

[1] [3]

IV. Cross-Validation

k-fold validation

complete randomization of the fold, by a random variable

A. Hyperparameters Optimization

Feature selection using the lasso function from [1]

Trying to avoid overfitting to be able to generalize to new examples.

we want to optimize the learning rate and the penalty rate

V. Results

Also talk about the α parameter for the gradient descent.

Talk about the mean squared error (MSE) obtain when we varied the alpha of the lasso

VI. Complementary Datasets

VII. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[2] R. Davidson and J. G. MacKinnon, *Econometric theory and methods*. Oxford University Press New York, 2004, vol. 5.

[3] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.