OLS

Alexandre Piche
260478404

Philippe Nguyen
260482336

Yash Lakhani
260500612

***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]. Explore dimension reduction algorithms.**

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. Optimization

Given the complexity of the optimization, we explored different strategy

A. Early Stopping

Stop the while loop when we start overfitting the test set.

B. Momentum

The idea is that we would like to take smaller step at the beginning of the optimization since the gradient is huge, but as we move closer to the real solution we might like to take larger step towards the optimal solution.

IV. Lasso and Ridge Regularization

Talk about variance vs bias

Penalizing decrease the variance but increase the bias.

[3]

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

According to the famous and widely accepted Occam's razor principle parsimonous models generalize better than more complex models. Is it therefore legitimate to select feature to increase our prediction power.

A. Ridge or L2-Regularization

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

[4]

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

B. Lasso or L1-Regularization

[1] [4]

V. Dimension Reduction

A. Principal Component Analysis

Given the large dimension of the dataset and that some of the feature are highly correlated we decided to reduced the dimension by using principal component analysis (PCA) algorithm. We noticed that $\%$ of the variance can be explained by the first 3 dimensions. The idea behind the PCA algorithm is trying to reconstruct X, by the minimal set of component. Namely we want to find a W such that

L linear basis vector

X is $K \times N$, where K is the number of feature and N the number of examples.

$$J(W, Z) \quad = \quad \|X - W Z^T\|_F^2 \tag{8}$$

Where W is $K \times L$ orthonormal and Z is $N \times L$ matrix [5] [1]

B. Feature Regularization

It was numerically challenging to apply the gradient to feature that are of multiple order different from each other. We can normalize without changing their span, since it is a linear transformation.

VI. 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

VII. 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

VIII. Complementary Datasets

Huffington post

IX. 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] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[4] 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.

[5] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.