**Abstract**

One of the most basic techniques in Machine Learning is Linear Regression. In this project, we therefore implemented a closed-form algorithm as well as a gradient-descent algorithm to perfect our understanding of basic Machine Learning concepts.

**Introduction**

The trends behind popular content on the internet are often extremely hard to predict. In this project, we hence decided to build a machine learning model that aims at predicting the “popularity score” of comments on the social network “Reddit”. We used different features of a dataset of reddit comments (N = 12000) to build a model that uses Linear Regression to predict their “popularity scores”.

**Dataset**

**Results**

1) In this first section of the analysis, all tests were done with only the first 3 features (by including a bias term and by excluding the text features). Before comparing the runtime, stability and performance of the closed form and gradient descent algorithms, we explored how to optimize the hyperparameters of the gradient descent with 3 features. In this part of the analysis, the parameter vector w0 initialized to 0 for all features.

We first compared how the convergence of the gradient descent is influenced by the β parameter (figure 1). By setting the initial learning rate (η0) to 1e-5 and the convergence threshold (ε) to 1e-4, we observed both how the mean square error (MSE) (fig.1a) and the speed of convergence (fig.1b) is affected by β. We can observe that there is a tradeoff between the reduction of MSE and the number of iterations, where higher β will converge faster, but with higher MSE, and vice-versa. We determined that a β of 1/10 offers a strong tradeoff between these two variables, and hence, this value was selected for further analysis with 3 features. Next, we investigated which value for η0 is be optimal for 3 features (figure 2). We figured out that, when setting ε = η0, higher values of η0 tend to reduce both the final MSE reached on the training and the validation datasets (fig.2a) and the number of iterations required to reach convergence to a large degree (fig.2a). Although in principle smaller learning rates would eventually reach better convergence than larger learning rates since the linear regression has a convex error function with a single global minimum, the runtime required with decreasing learning rates grows extremely fast (exponentially?). Hence, a η0 of 1e-4 was selected, since learning rates above this value tended to diverge. An ε of 1e-4 was selected since smaller values would tend to extend greatly the runtime of the algorithm.

Then, we compared the performances of the closed-form and gradient descent algorithms. The runtime of the closed-form algorithm was much faster with a runtime of approximately 2.72 ms, whereas the runtime of the gradient-descent with the previously optimized hyperparameters was of approximately 816.05 ms. We next investigated the stability of learning for the two different algorithms by evaluating how the learned weights vary depending the training samples. The training set was separated into 10 different sub-samples of 1000 examples, and each algorithm was run over each of these sub-samples. We then computed the standard deviation of the learned weights across the different subsamples. The mean standard deviation was of 0.122 for the closed-form algorithm and 0.052 for the gradient descent. Hence, the gradient descent demonstrated clearly a more stable learning with only half the variability of the closed-form algorithm. Finally, the performance of the two algorithms was compared according to their MSE between the training and validation datasets. The closed-form had a MSE of 1.08468 and 1.02032 on the training and validation sets respectively, whereas the gradient-descent obtained an MSE of 1.08639 and 1.02310. The two algorithms thus had a very similar performance, although the closed-form performed slightly better on both the training and validation sets.

2) Using the closed-form approach, by using 3 features, we obtain a Mean-Squared Error (MSE) of 1.08468307 for the training set, and 1.02032668 for the validation set. By using 63 features, we obtain an MSE of 1.06142257 for the training set, and 0.98696264 for the validation set. By using the 163 features, we obtain an MSE of 1.04871848 for the training set, and 0.99454456 for the validation set.

->overfitting with 163 word features; get better performance on training set but worse on validation set

Using the gradient descent, with hyperparameters epsilon=1e-4 and learning rate= 1e-4, MSE with 3 features is 1.0863914454752952 on training set and 1.0231009744676112. With 63 features and an alpha=1e-5, it is 1.081090406594019 for training and 1.0129705833879499 for validation. With 163 features, it is 1.075314229923179 for training and 1.0073406707196304for validation.

\*we should consider that although the closed form tend to overfit with 163 features, the gradient descent doesn’t necessarily

3)

**Discussion and Conclusion**

**Statement of Contribution**