Machine Learning Project 1

Yifei Song, Haoming Lin, Ruiqi Yu
École Polytechnique Fédérale de Lausanne, Switzerland

I. Data Preprocessing

Firstly we separate features into two sets, one containing meaningless values(-999) and the other without. For the first set, only the normalization is applied. For the second, some features are log-scaled before the normalization process since the original data are highly skewed. After that, we applied polynomial feature augmentation described as follows.

Let $X := [x_1, x_2, \ldots, x_D] \in R^{N\times D}$ be the processed features of the second set. The augmented features set is the union of the two following sets:

$$S_0 := \{\mathbf{1}\}$$
$$S_1 := \{x_i^d : 1 \le i \le D;\ 2 \le d \le maxDeg\}$$
$$S_2 := \{x_i x_j : 1 \le i, j \le D\}$$
$$S_3 := \{x_i^2 x_j : 1 \le i, j \le D\}$$

where $maxDeg$ is a hyperparameter to be defined. Note that all the operations are element-wise. $1 \in R^{N\times 1}$ is a vector where every entry is equal to 1.

In the following sections, the data are split into training and test sets for each experiment, containing 80% and 20% of the data respectively. Later, the training set will be split into new training and validation sets by 75% and 25%, respectively.

II. Method

A. Least Square - (Stochastic) Gradient Descent

Least square is one common date-fitting method. The loss function of it with mean square error is:

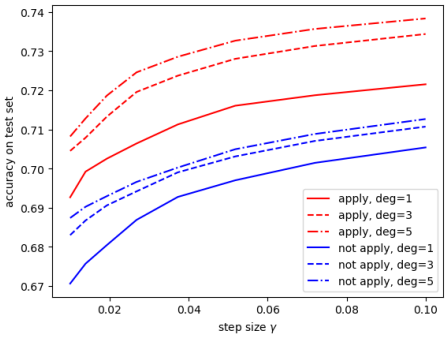
$$\mathcal{L}(\mathbf{w}) = \frac{1}{2N}\sum_{n=1}^{N}\left(y_n - \mathbf{x}_n^\top w\right)^2 = \frac{1}{2N}(\mathbf{y}-\mathbf{X}w)^\top(\mathbf{y}-\mathbf{X}w)$$

We use gradient descent(GD) and then stochastic gradient descent(SGD) to find the optimal w. The update rule of w is:

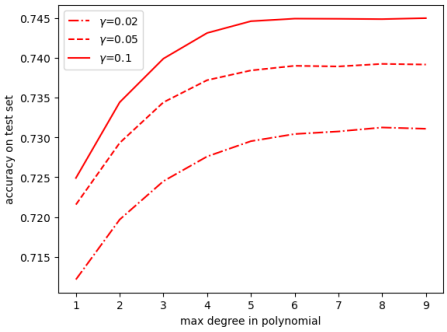
$$w^{t+1} := w^t - \gamma g$$

where g is $-\frac{1}{N}\mathbf{X}^\top(\mathbf{y}-\mathbf{X}w)$ and $-x_n(y_n - x_n^\top w)$ for GD and SGD respectively; $\gamma > 0$ is the learning rate (step size).

The experiment results of this model using gradient descent are shown in Figure 1. In Figure 1(a), models are trained with $max\_iter = 1000$ iterations and the weights are initialized at $\mathbf{0}$. We can notice that the log-scaling on the skewed-distributed features significantly improves the performance of models, regardless of learning rate γ and max degree in polynomial augmentation. It indicates that a relatively large learning rate γ and a high polynomial degree leads to a high accuracy score.



(a) apply log-scale



(b) max degree in polynomial

Figure 1: MSE with Gradient Descent

Degree	5	9	11	13	15
Accuracy	0.7877	0.7985	0.8001	0.79692	0.52068

Table I: Validation Results with different degrees

B. Least Square - Normal Equation

In this part, we implement the least square method by solving the normal equation $\mathbf{X}^\top(\mathbf{y}-\mathbf{X}\mathbf{w}) = \mathbf{0}$, where the term $(\mathbf{y}-\mathbf{X}\mathbf{w})$ is seen as the error and its ideal value is 0.

Table I summarizes the results on the validation sets. The accuracy increases initially with the degree, reaching the maximum at degree 11, and decreases afterward.

C. Ridge Regression

In the ridge regression, the parameters can be explicitly solved by the equation: $w_{rigid} = (\mathbf{X}^\top\mathbf{X} + 2\mathbf{N}\lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$,

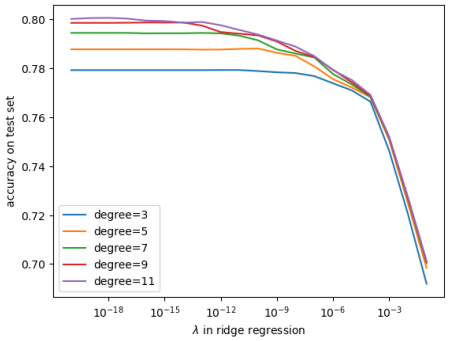


Figure 2: Ridge Regression

where λ is the weight for constraint.

Figure 2 shows the results of ridge regression with different λ. We observed that the model performs better with smaller λ. Since the best model does not outperform the least square with normal equation, the penalty term does not contribute to the performance of models in our setting.

D. (Regularized) Logistic Regression

Finally, we use the logistic regression method to represent the probability of set categories for the binary classification:

$$p(1 \mid \mathbf{x}, \mathbf{w}) = \sigma(\mathbf{x}^\top \mathbf{w} + w_0), p(0 \mid \mathbf{x}, \mathbf{w}) = 1 - \sigma(\mathbf{x}^\top \mathbf{w} + w_0),$$

where 1 and 0 represent the two results in the binary methods. The sigmoid function $\sigma(z) := \frac{e^z}{1+e^z}$ is compatible with the values given by linear functions and maps it into S-curve between 0 and 1.

The loss function used in the logistic regression is logistic loss or cross-entropy loss, which is given by:

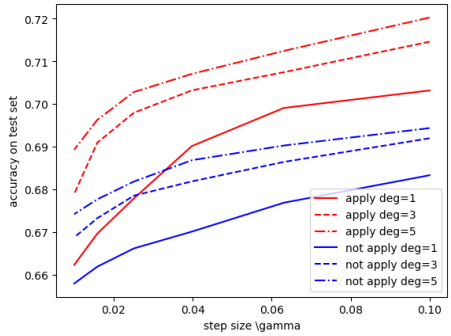
$$\mathcal{L}(\mathbf{w}) = -\frac{1}{N} \sum_{n=1}^{N} y_n \ln \sigma\left(\mathbf{x}_n^\top \mathbf{w}\right) + (1 - y_n) \ln\left[1 - \sigma\left(\mathbf{x}_n^\top \mathbf{w}\right)\right]$$

The gradient of the loss function is:

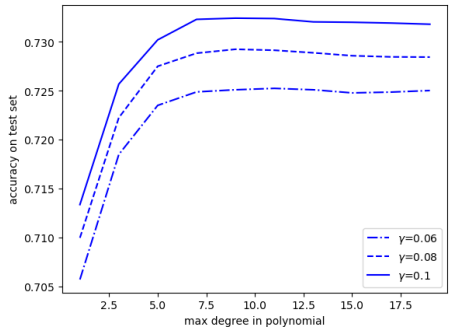
$$\nabla \mathcal{L}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n \left(\sigma\left(\mathbf{x}_n^\top \mathbf{w}\right) - y_n\right) = \frac{1}{N} \mathbf{X}^\top [\sigma(\mathbf{X}\mathbf{w}) - \mathbf{y}]$$

For the regularized logistic regression, a penalty term is added as the regularization to avoid the overfitting problem: $L_R(w) = L(w) + \lambda \|w\|^2$ (λ is the weight for the regularization item). Gradient descent is to find the global optimal weights for the unregularized logistic and regularized regressions.

Figure 3 shows the accuracy of logistic regression with different hyperparameters when the iteration number is 1000. Figure 1(a) once again justifies the use of log-scaling. The accuracy is increased by approximately 0.02 by applying log-scaling. It also shows that the accuracy increases with the learning rate γ. However, we observe that too large ($\gamma \geq 0.01$) learning rate may prevent the model from converging.



(a) apply log-scale



(b) max degree in polynomial

Figure 3: Logistic Regression with Gradient Descent

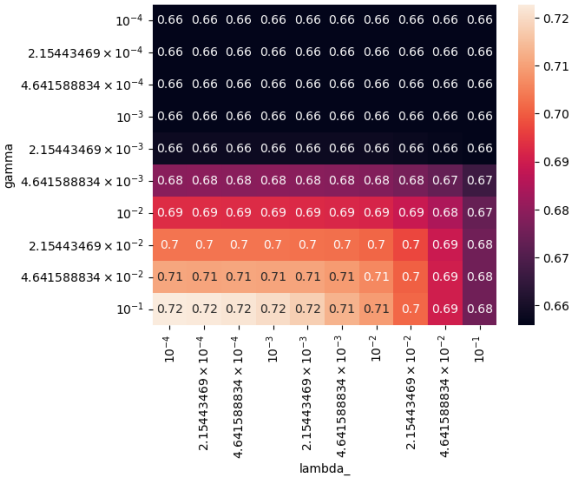


Figure 4: Regularized Logistic Regression

The experiment results with regularized logistic regression are shown in Figure 4. The accuracy is computed for different models after 1000 iterations with λ and γ varying in range $[10^{-4}, 10^{-1}]$. The best accuracy is achieved with $\gamma = 0.01$ and $\lambda = 10^{-4}$. We observed that smaller λ lead to high accuracy, the model is possibly suffering from underfitting.