Project 1. Report
Machine Learning Course

Mikhail Seliugin, Aleksandr Alekseev, Yauheniya Karelskaya
School of Computer and Communication Sciences, EPFL, Switzerland

Abstract—**The first part of the Machine Learning Course at EPFL provided us with the basic ML techniques such as linear, logistic regression. At this paper the problem of classification is considered to predict whether the event was signal (a Higgs boson) or background (something else). Based on logistic regression, this work includes exploratory data analysis, feature processing and estimation of the suggested methods.**

I. Introduction

The aim of the work was to solve classification problem on dataset [1] using the methods highlighted during the first part of the Machine Learning Course.

In Section II the methods we implemented to solve the problem is explained, in Section III main steps of feature preprocessing is considered, Section IV includes experiments designs and results and Section V consists of experiments discussion and our contribution summary.

II. Implemented methods

A. Linear regression

Dataset consist of $(\mathbf{x_n}, y_n)_{n=1}^N$, where $\mathbf{x_n}$ is features with added bias and y_n is an output. The problem is to find a vector $\mathbf{w}$: $\mathbf{x}^T\mathbf{w}$ is the best prediction for output y_n.

To measure whether the prediction is precise loss function L is used. Specifically, in this project, it is mean squared error (MSE). Then the problem could be reformulated as minimizing the loss function that could be examining by gradient calculation. Due to convexity of loss function, point with zero gradient will be a global minimum. This point could be calculated in the close form or by gradient descent.

- **Least squares method** is solution of normal equation:

$$\begin{gathered} L = \frac{1}{2N}(\mathbf{y} - X^T\mathbf{w})^2 \Rightarrow \\ \nabla L = 0 \Leftrightarrow X^T X\mathbf{w} = X^T\mathbf{y} \end{gathered} \tag{1}$$

  To avoid overfitting, **ridge regression** is used:

$$\mathbf{w}^* = arg\min_w L(\mathbf{w}) + \Omega(\mathbf{w}), \tag{2}$$

  where $\Omega(\mathbf{w}) = \lambda\|\mathbf{w}\|_2^2$ is regularizer. In that case the solution also could be derived in closed form.

- **Gradient descent** is iterative method where the minimum is reached step-by-step:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \gamma\nabla L(\mathbf{w}^{(t)}), \tag{3}$$

  where γ is step-size. To make the calculations of gradient faster it is possible to use the fact that if $L_n = \frac{1}{2}(y_n - \mathbf{x}^T\mathbf{w})^2$ and $L = \frac{1}{N}\sum_{n=1}^N L_n$, then $E[\nabla L_n] = \nabla L$ and:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \gamma\nabla L_n(\mathbf{w}^{(t)}), \tag{4}$$

  Thus, iteration by gradient of loss function at one point is less precise, but faster. Due to random selection of n it is called **stochastic gradient descent**.

B. Logistic regression

Linear regression is not considered as an effective way to solve classification problem, i.e. problem with discrete output y. To solve binary classification task, which is figured out in this work, logistic regression is proposed. It uses non-linear function, e.g. sigmoid, to predict a probability that event is related to one class or another.

$$\begin{aligned} \sigma(\eta) &= \frac{e^\eta}{1 + e^\eta} \\ p(1|x) &= \sigma(\mathbf{x}^T\mathbf{w}) \\ p(0|x) &= 1 - \sigma(\mathbf{x}^T\mathbf{w}) \end{aligned} \tag{5}$$

To design the best prediction maximum likelihood estimator is proposed.

$$\begin{gathered} \mathcal{L}(\mathbf{w}) = p(\mathbf{y}, X|\mathbf{w}) = p(X)p(\mathbf{y}|\mathbf{w}) \\ \mathbf{w}_* = arg\max_w[\mathcal{L}(\mathbf{w})], \end{gathered} \tag{6}$$

where $\mathcal{L}$ — likelihood. Features do not depends on weights, so the problem is to find an $arg\max_w p(\mathbf{y}|\mathbf{w})$.

$$\begin{aligned} p(\mathbf{y}|\mathbf{w}) &= \prod_{n=1}^N p(y_n|\mathbf{x_n}, \mathbf{w}) = \\ &= \prod_{n=1}^N \sigma(\mathbf{x}^T\mathbf{w})^{y_n}(1 - \sigma(\mathbf{x}^T\mathbf{w}))^{1-y_n} \\ \mathbf{w}_* &= arg\max_w[\mathcal{L}(\mathbf{w})] = arg\min_w[-\log p(\mathbf{y}|\mathbf{w})] = \\ &= arg\min\left[\frac{1}{N}\sum_{n=1}^N -y_n\mathbf{x_n}^T\mathbf{w} + \log(1 + e^{\mathbf{x_n}^T\mathbf{w}})\right] \end{aligned} \tag{7}$$

Thus, new cost function is introduced. It is convex.

Although there is no closed form solution like on the previous point, the minimum could be reached by gradient descent. Similarly, regularization term could be added to avoid overfitting.

III. Exploratory data analysis and feature processing

A. Removing correlated features

Since in this classification problem we use a linear model (logistic regression), the first step in data processing is to remove correlated features. To do this, we calculate the Pearson correlation between each feature

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}}, \tag{8}$$

and then remove the pairs for those where the correlation is close to 1.

B. Outliers processing

In the proposed dataset, there are a number of features that could take the value -999. It means that it is not measured. In order to make computations easier, but keep their isolation, we replace -999 with the value that is slightly less than the infimum of the other objects distribution (in our case, we chose -5).

C. Normalization

The last step in data preprocessing is normalization. Its necessity is due to different scales and ranges of the features. Thus, there is an imbalance between their influence on the output and the trained model may reveal incorrect dependencies. In this problem, we standardize the features by subtracting the mean and scaling to unit variance

$$z_i = \frac{x_i - \overline{x}}{\sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n}}}. \tag{9}$$

IV. Experiments and results

A. Experimental setup

We split the data into train and validation for our experiments, 90% of data for train and 10% for validation. We use cross-validation on 3 folds for obtaining our experimental metrics. As an optimization method we use the gradient descent. The baseline step $\gamma = 0.05$ and the number of iterations is 1000. We report the mean and standard deviation of accuracy calculated on a cross-validation splits.

B. Experiments

The results are shown in Fig. 1. The numbers on X-axis correspond to the following methods.

0 Logistic regression. As a baseline we used a logistic regression method described in section II. Accuracy: 0.728 ± 0.006.

1 Logistic regression with adaptive step size. The step size γ for gradient descent is decreasing linearly from the baseline value 0.05 at the first iteration to the value 0.001 at the last iteration. Accuracy: 0.738 ± 0.002.

2 Logistic regression with regularization. We add the regularization to logistic regression and perform the grid search for regularization term λ over the values $\{0.001,\ 0.01,\ 0.1,\ 1\}$. Accuracy with the best one $\lambda = 0.01$ is 0.7319 ± 0.0006.

3 Logistic regression with polynomial features. We add the squares of each feature (except the bias) to the dataset and run the logistic regression. Accuracy: 0.749 ± 0.012.

4 Logistic regression with polynomial features, regularization ($\lambda = 0.01$) and step-size adaptation. Accuracy: 0.758 ± 0.002.

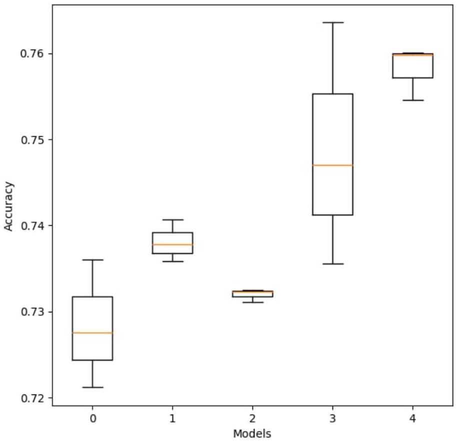

Figure 1. Accuracy

V. Discussion

We have worked with the linear regression and implemented several approaches to boost the performance. All the proposed methods boost the performance over the baseline, but the most significant boost is achieved by applying polynomial features, regularization and step size adaptation. This can be explained by the fact polynomial features enrich the feature space independently from original features. Regularization is a technique to prevent overfitting caused by adding extra features to the data and it works well with excessive models with a lot of polynomial features. Step size adaptation helps to converge on later stages where the model oscillates nearby the minima.

To further increase the model performance, one can try proper feature selection, different step-size adaptation methods and expand the polynomial features.

References

[1] "Ml higgs," https://www.aicrowd.com/challenges/epfl-machine-learning-higgs, 2022.