CMPE 482 - Final Project - Bike Sharing Prediction

İsmet Burak Kadron

May 28, 2015

1 Problem Statement

Our aim is to predict bicycle usage in Washington D.C's bike sharing system using previous data. Features are date, season, working day, holiday, weather indicator, temperature, apparent temperature (function of humidity and temperature), humidity. We have three usage outputs. First output indicates usage of casual (non-registered) users, second output indicates usage of registered users. The last one is total usage, sum of aforementioned outputs.

2 Proposed Solution

In the proposal, we proposed an Multilayer Perceptron (MLP) based solution as these are used for time-series data such as finance data. I tried to implement an MLP with 2 hidden layers but I encountered a bug that I couldn't fix. When I couldn't solve the problem, I switched to another MLP variant, Extreme Learning Machines. Extreme Learning Machines are similar to MLP with a single hidden layer but instead of backpropagation, weights of second layer are found analytically.

The idea is, instead of assigning all weights randomly and apply stochastic gradient descent, we assign input weights randomly and leave them as they are. Since hidden weights multiplied by hidden units should give us the outputs, we solve this problem using least square solution and Moore-Penrose inverse. We also regularize the inverse using Tikhonov regularization, so that overfitting does not happen. [HZS04],[HZDZ12]

$$h_i = f(\sum_j (w_{ij}x_j) + b_i);$$

$$H = f(WX + B); \qquad \text{(Obtaining hidden unit values in matrix form.)}$$

$$Y = HV; \qquad \text{(Formula for obtaining output in matrix form. Y and H are known.)}$$

$$V = (H^T H + \Gamma^T \Gamma)^{-1} V^T Y \qquad \Gamma = \alpha I \qquad (\alpha \text{ is the regularization constant.})$$

As you can see by the formulas, V is found using least square solution. Psuedo inverse is regularized by Tikhonov regularization. The problem with this approach is all training data is trained at once with a single pass, so it needs a quite amount of memory according to the hyperparameters and size of training data.

3 Results

Our score was based on Root Mean Squared Logarithmic Error (RMSLE). It is calculated as

$$\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i + 1) - log(a_i + 1))^2}$$

Where ϵ is RMSLE value, n is total number of data points, p_i is our prediction, a_i is actual value for i'th data point.

As I understand, the ratio between prediction and actual values is more important rather than simple difference, so I trained the training data using logarithms of outputs and it performed better than my previous tries on the test set. It performed better and since logarithm lower bounded the outputs by 0, my predictions didn't have much fluctuations.

We use half of the data as training and other half as validation. Number of hidden units is 750 and regularization constant is 0.1. We train ELM in 9 trials and take the average of test results for submission.

Our final score on Kaggle is 0.99909 with team name Kadron. It is performing better than Mean Value Benchmark which has score 1.58456.

4 Future Work

In the future, we can compare these results with results obtained by SVM's and Decision Trees. This may also help us in a voting scheme by different trainers.

We haven't modified features except to scale discrete features between 0-n. We can use feature extraction methods like PCA to obtain new features and use them in training.

Bibliography

[HZDZ12] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(2):513–529, 2012.

[HZS04] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 985–990. IEEE, 2004.