EP2420 Project 1 - Advanced Project
Comparing Linear Regression, Random Forest Regression, and Neural Network Regression to estimate Service Metrics

Rolf Stadler Forough Shahab

October 25, 2020

Project Objective

In this project, you train regression models that map infrastructure measurements X to predictions of service-level metrics Y. The services we consider are Video-on-demand (VOD) or a Key-Value store (KV), and the service-level metrics are Video Frame Rate or Response Time as experienced by a service client.

Using machine-learning techniques, the problem is to find a function (i.e., to train a model) $M : X \rightarrow \hat{Y}$, such that $\hat{Y}$ closely approximates Y for a given X. If Y has a numeric value, the problem is referred to as a regression problem; if Y is a class label, the problem is called a classification problem. This project considers both of these problems. The machine-learning methods you will use are linear regression, random forest regression and classification, and neural network regression.

To train M, measurement pairs (or observations) of the form (x_t,y_t) are needed. A set of measurement pairs, indexed by time, is also called a trace. In this project a trace based on observations collected once per second from running the services on a testbed during several hours. You can find a description of the infrastructure and the measurements that produced the trace you use in this project in [1].

Project tasks

The project is composed of five tasks. You will be given a data trace for analysis.

Task I - Data Exploration and Pre-processing

1. Describe the data set in terms of the number of samples, the number of features, and the number of types of targets. Choose 10 features uniformly at random. Provide a short description of these features using the linux manual pages for the command sar. Compute the following statistics for each of these features and for and the target: mean, standard deviation, maximum, minimum, 25th percentile, and 90th percentile. Give no more than two digits after the decimal point, for instance 52.3 or 5.23e+01.

2. Pre-process the feature samples and produce six design matrices X as numpy arrays. A design matrix is a matrix whose row vectors represent samples and whose column vectors represent features.

   (a) L^2 Normalization: linearly scale the values of each feature column (or sample row) so that its L^2-norm becomes 1.

   (b) Restriction to Interval: linearly scale the values of each feature column (or sample row) so that they all lie within the interval [0,1].

   (c) Standardization: linearly scale the values of each feature column (or sample row) so that they all have 0 mean and a variance of 1.

Task II - Estimating Service Metrics from Device Statistics

2.1 Evaluate the Accuracy of Service Metric Estimation

1. Model Training - train three models M on the training set using the methods linear regression, random forest regression, and neural network regression.

2. Train and test your models M with the so-called validation-set technique. This technique entails that you split the set of observations into two parts: the *training set* for computing the model M and the *test set* for evaluating the accuracy of M. From the complete set of observations, you select uniformly at random 70% of the observations to form the training set and then assign the remaining 30% to the test set.

3. Accuracy of Model M - compute the *estimation error* of the models M on the test set. We define the estimation error as the *Normalized Mean Absolute Error* $(NMAE) = \frac{1}{\bar{y}}(\frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i|)$, whereby $\hat{y}_i$ is the model estimation for the measured service metric y_i, and $\bar{y}$ is the average of the observations y_i of the test set. Note that $\hat{y}_i = M(y_i)$.

4. Provide this evaluation of accuracy for all three methods. For random forest and neural network, list the hyper-parameters that give the best accuracy for your data trace.

5. As a baseline for M, use a naïve method which relies on Y values only. For each $x \in X$ it predicts a constant value $\bar{y}$ which is the mean of the samples y_i in the training set. Compute $\bar{y}$ for the naïve method for the training set and compute the NMAE for the test set.

6. Choose one model (linear regression, random forest, or neural network) and produce a time series plot that shows both the measurements and the model estimations for the target on the test set. Show also the prediction of the a naïve method (see Figure 1).

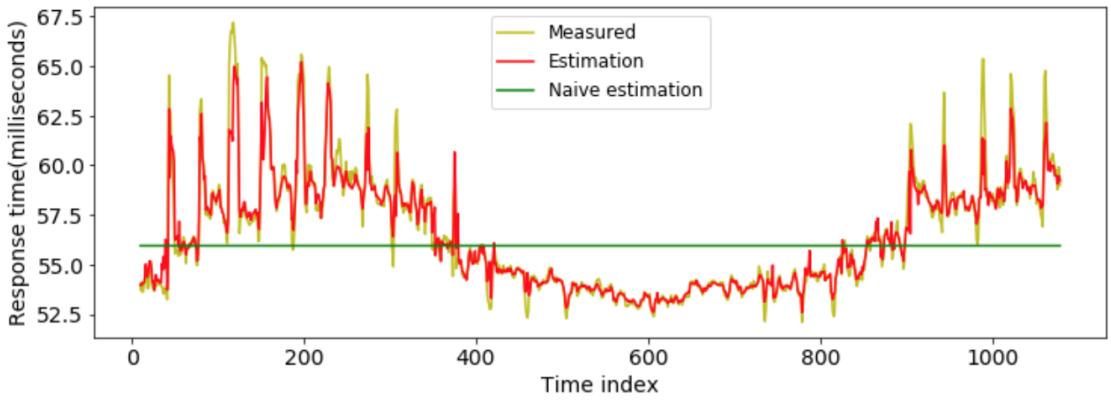

Figure 1: Example: time series plot for KV store trace

7. Produce a density plot and a histogram for the target values on the test set. Set the bin size of the histogram to 1 frame for Video Frame Rate or 1ms for Response Time.

8. Produce a density plot of the estimation errors $y_i - \hat{y}_i$ in the test set.

9. Based on the above results, figures and graphs, discuss and compare the accuracy and computational overhead of estimating the target metric for the regression methods you used. You can measure the computational overhead as the training time.

2.2 Study the Relationship between Estimation Error and the Size of the Training Set

1. From the trace of observations S, create eight training sets $S_1, ..., S_8$ by selecting uniformly at random 25, 50, 100, 200, 400, 800, 1600, and 3200 observations. For each training set S_i, you create a test set T_i, by selecting uniformly at random 1000 samples from S which have not been chosen for S_i. You end up with six test sets $T_1, ..., T_8$.
2. Train a linear model for each training set S_i and compute the $NMAE$ for this model on the corresponding test set T_i, $i = 1, ..., 8$.
3. Perform the above 50 times, so you train and evaluate models for 50 different pairs of training set and test set for a given training set size.
4. Produce a plot that shows $NMAE$ for M (vertical axis) against the size of the training set (horizontal axis). Use error bars or box plots to show the range of the $NMAE$ values for a given set size.
5. Based on the above, discuss the relationship between the estimation error and the size of the training set.

Task III - Studying the Impact of Data Pre-processing and Outlier Removal on the Prediction Accuracy

1. In Task I, you performed six different forms of pre-processing on the collected device measurements X. Together with the unprocessed measurements, you can create seven different design matrices X.
2. Perform a comparative study to find out which of the pre-processing method performs best with each of the regression methods on your data set. Can you achieve a significant accuracy gain through pre-processing? Which pre-processing method is effective for all regression methods?
3. Detect and remove outliers. Take the design matrix whose feature columns are standardized (see above). We call a sample an outlier when one of its components has an absolute value larger than a given threshold T. Compute and plot the number of outliers of your data set in function of T.
   The idea is that once the threshold T is decided upon, all samples with components whose absolute value is larger than T are removed from the data set.
4. Investigate the error of a regressor in function of T for your data set. Select random forest as regression method and evaluate for $T = 10, 20, 30, ...100$.

Task IV - Predicting the Distribution of Target Variables using Histograms

1. The basic idea of this task is to discretize the target space Y and use a histogram estimator for predicting $P(Y|X)$. This means that each $x \in X$ is mapped onto a histogram on Y.
2. In the case of Y representing the Video Frame Rate, the y values are integers, i.e., 15 Frames/sec. Consider the histogram on the interval $y \in [0.5, 30.5]$ with a bin size of 1. This results in 30 bins, with mid points 1, 2, .., 30.
3. In the case of Y representing the Response Time, consider the histogram on the interval $y \in [y_{min}, y_{max}]$ whereby y_{min} is the minimum y value in the training set and y_{max} is the maximum y value. Divide this interval into 20 bins of equal size.
4. Consider each bin of the histogram as a separate class and use a random forest classifier to predict the density for each class.
5. To evaluate the accuracy of the method, compute the NMAE between the expectation of the predicted value and the measured value over the test set. Compare the result with those from Tasks II and III.

6. For illustration purposes, chose two x-samples from the test set and draw the two predicted histograms. For both histograms indicate the measured y-values.

7. For this task choose the data set X that has been the outcome of pre-processing and outlier removal. For pre-processing chose the method that gives the best results in Task III. For outlier removal, chose a threshold that keeps 99% of the samples in the data set.

Task V - Predicting Percentiles of Target Metrics

1. The goal of this task is to use the histogram estimator from Task IV to predict the 20^{th}, 50^{th}, and 95^{th} percentile values of the target Y.

2. Given an instance of a histogram described in Task III (for Video Frame Rate or Response Time), describe how you compute the above given percentile values.

3. Compute the 20^{th}, 50^{th}, and 95^{th} percentile values of the target Y on the training set used in Task IV.

4. Consider the x samples in your data set that belong to the first hour (3600 second) of the experiment. Produce a time series plot that shows the predicted 20^{th}, 50^{th}, and 95^{th} percentile values of the target Y, together with the measured values.

5. To evaluate the accuracy of the predicted percentile values, you compute $\frac{1}{n}\sum_{t=1}^{n} \mathbb{1}\{y^{(t)} \leq a_{perc}(x^{(t)})\}$, for $perc = 0.2$, 0.5, and 0.95. In the formula, $(y^{(t)}, x^{(t)}), t = 1...n$ are the samples of the test set. $a_{perc}(x^{(t)})$ is the predicted percentile value for $x^{(t)}$. You can obtain this value from the histogram for $x^{(t)}$. $\mathbb{1}\{Q\}$ denotes the indicator function, which takes the value 1 if Q is true, and the value 0 otherwise. The summation is over the test set. The formula gives an estimation for *perc*. (The Clivenko-Cantelli Theorem gives the foundation for the estimation method.)

References

[1] F. S. Samani, H. Zhang, and R. Stadler, "Efficient learning on high-dimensional operational data," in *2019 15th International Conference on Network and Service Management (CNSM)*, IEEE, 2019.