Online Learning for Linear Regression and Quantile Linear Regression

PARC Term Paper
M.Sc. in Data Science

Shiuli Subhra Ghosh, MDS202035

Advisor:
Dr. Sushma Kumari

Academic year:
2021

Abstract: Exponential increase in the volume of data incurs some storage problems which is going to be more severe in near future. Online learning is one of the best solutions for solving this problem. Compared to "traditional" ML solutions, online learning is a fundamentally different approach, one that embraces the fact that learning environments can (and do) change from second to second. So, linear models can be extremely fast and powerful in terms of training when we will use this method. We can also implement online learning for different loss functions with or without taking consideration of some regularization techniques. This work for Online Linear Regression will be extended to understand Quantile Regression. The heteroscedastic Normal or asymmetric Pareto distributed, noisy training data make the estimation via Linear Regression less efficient, i.e. we need more data to get stable results and, in addition, large outliers can have a huge impact on the fitted coefficients. In this asymmetric setting, the median or different quantiles give additional insights. In fact for Online Learning, each data streaming points might be useful and produce better estimate than a normal Linear Regression model. Comparison of online methods for Linear and Quantile Regression might give us some beautiful insights for symmetric and asymmetric targets.

Key-words: Online Learning, Linear Regression, Quantile Linear Regression, Ordinary Least Square, Ridge Regression, Elastic-Net Regression, Lasso Regression, Passive Aggressive Algorithms

1. Data Sets

1. First we will test these methods in two simple simulated data sets. One for heteroscedastic Normal distributed target and one for asymmetric Pareto distributed target.
2. mtcars
3. Old Faithful Geyser

2. Introduction

In this term paper we will discuss about the importance of the Online Learning Methods for Linear and Quantile Linear Regressions with the help of some examples. We will also discuss about the difference between Linear Regression and Quantile Linear Regression. In statistics, linear regression is a linear approach for modelling the relationship between a scalar response and one or more explanatory variables (also known as dependent and independent variables). It can be multivariate where multiple correlated dependent variables are predicted, rather than a single scalar variable. But here, as our main focus is to discuss the importance of the Online Learning techniques for Regression Analysis we will restrict our discussion for the uni-variate case for better explainability and visualization.

2.1. Data Set Description

In the First Section of this term paper, we have mentioned about the Data Sets. Now, I will explain the data sets in more details.

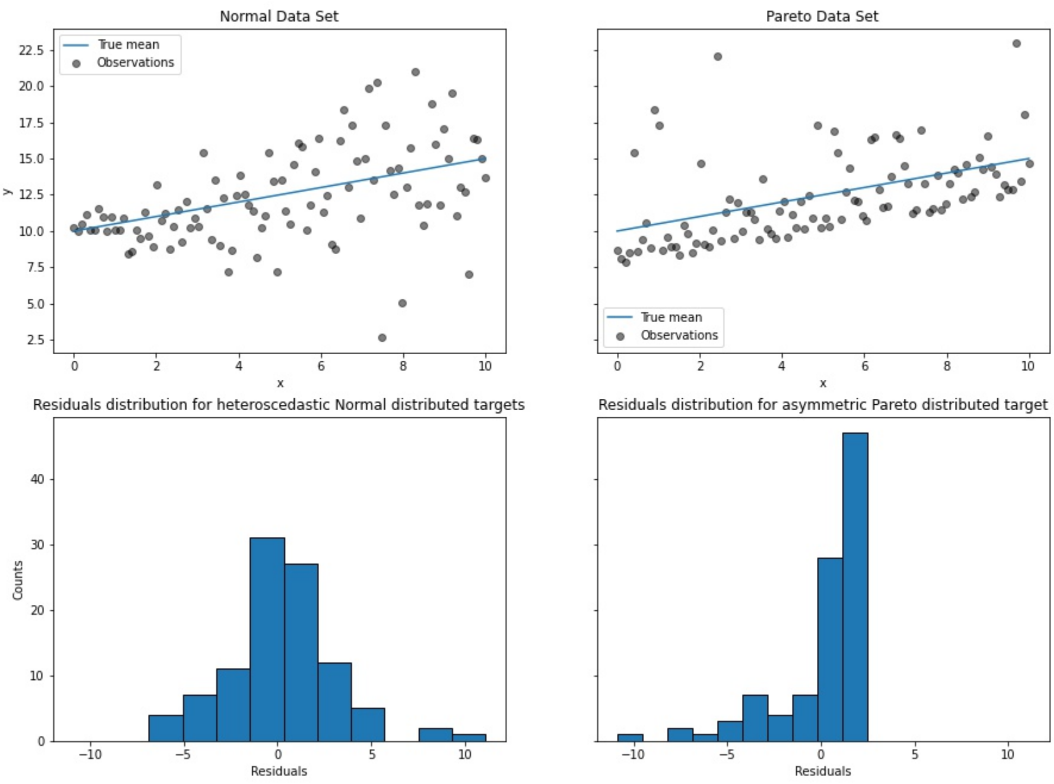
1. **Normal Data Set:**
   This is a synthetic data set which we generated by adding heteroscedastic normal noise with the linear equation $y = 10 + 0.5x$

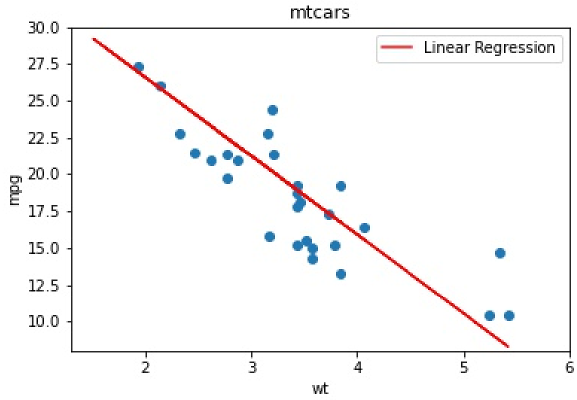
   ```
   y_normal = y_true_mean + rng.normal(loc=0, scale=0.5 + 0.5 * x, size=x.shape[0])
   ```

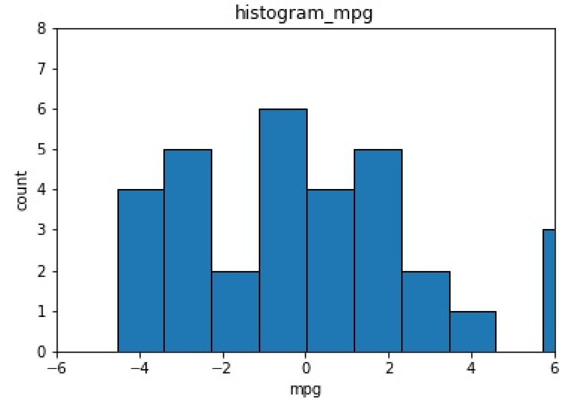
2. **Pareto Data Set:**
   This is a synthetic data set which we generated by adding asymmetric Pareto noise noise with the linear equation $y = 10 + 0.5x$

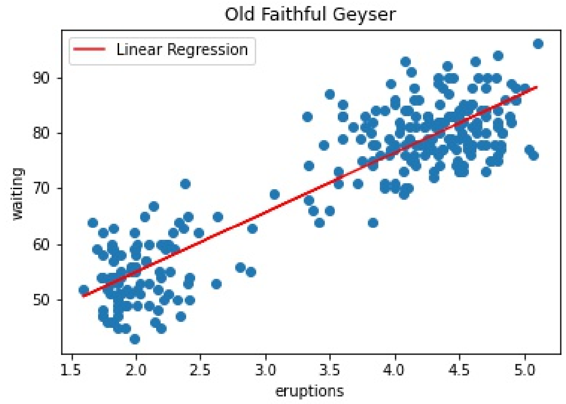
   ```
   a = 5
   y_pareto = y_true_mean + 10 * (rng.pareto(a, size=x.shape[0]) - 1 / (a - 1))
   ```

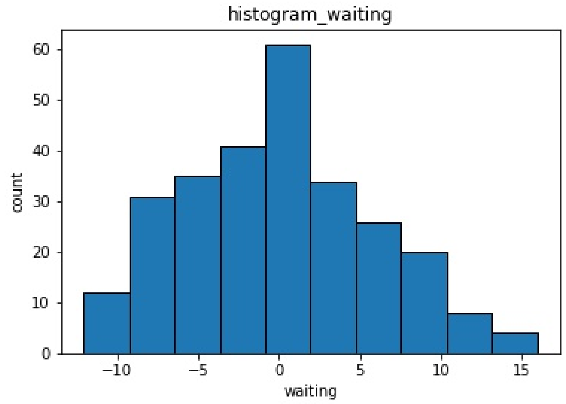
3. **mtcars Data Set:**
   The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models). Here we will the linear model to find the correlation between weight ('wt') and miles per gallon ('mpg') attributes. It should be essentially have negative slope as if the weight of the car is more i.e if the car is bigger then it should consume more petrol.
4. **Faithful Data Set:**
   Old Faithful is a cone geyser in Yellowstone National Park in Wyoming, United States. It is a highly predictable geothermal feature and has erupted every 44 minutes to two hours since 2000. Eruptions can shoot 3,700 to 8,400 US gallons (14,000 to 32,000 L) of boiling water to a height of 106 to 185 feet (32 to 56 m) lasting from $1\frac{1}{2}$ to 5 minutes. The average height of an eruption is 145 feet (44 m). Intervals between eruptions can range from 60 to 110 minutes, averaging 66.5 minutes in 1939, slowly increasing to an average of 90 minutes apart today, which may be the result of earthquakes affecting subterranean water levels.











3. Problem Statement

If there is a huge number of data points and the data is streamed at times, then the entire linear regression model needs to be updated again and again when one new data point will come.

4. Methodology

In this term paper, we will implement Online Learning Algorithms from scratch or we might use some existing packages.

- There are several python libraries (Vowpal Wabbit[5], River etc.[3]) which already implement online learning for Linear Regression and classification.
- We will try to reconstruct the Online Learning Algorithms for different Regression techniques (Linear and Quantile).[4]
- We will also reconstruct the Online Learning method for Quantile Regression. [1]
- Then we will compare our Online methods with the classical ones for the 4 data sets mentioned earlier.

4.1. Linear Regression

Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y).
Considering x has k features and n data samples,

$$X = \begin{pmatrix} 1 & x_1^1 & \dots & x_1^k \\ 1 & x_2^1 & \dots & x_2^k \\ 1 & x_3^1 & \dots & x_3^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n^1 & \dots & x_n^k \end{pmatrix} \qquad \theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_k \end{pmatrix}$$

Then the hypothesis for Linear Regression is,

$$h_\theta(x) = \hat{y} = X\hat{\theta} \tag{1}$$

Where, $\hat{\theta}$ and $\hat{y}$ are the estimates.
With simple linear regression when we have a single input, we can use statistics to estimate the coefficients (w) and intercept/bias (b). Different techniques can be used to prepare or train the linear regression equation from data, the most common of which is called Ordinary Least Squares (OLS). Let the OLS loss function is $J(\theta)$.

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(h_\theta(x_i) - y_i\right)^2 \tag{2}$$

The Ordinary Least Squares procedure seeks to minimize the sum of the squared residuals. This means that given a regression line through the data we calculate the distance from each data point to the regression line, square it, and sum all of the squared errors together. This is the quantity that ordinary least squares seeks to minimize. When there are one or more inputs we can use a process of optimizing the values of the coefficients by iteratively minimizing the error of the model on your training data. For minimizing the Loss function, we can use the **Normal Equation**($\hat{\theta} = (X^TX)^{-1}X^Ty$), but this is computationally expensive. Hence, we use **Gradient Descent Algorithms** for the optimization of this convex loss function.

4.1.1 Gradient Descent Algorithm

As we have said earlier, we follow iterative approach.

1. Initially we guess a line.
2. Then we calculate the error over all the data points.
3. Then we try to reduce the OLS loss function.
4. Then we keep adjusting the line

Stopping criterion is, when we find the best fit line or the incremental progress is less than a small value i.e there will not be any significant change .

4.2. Online Linear Regression Algorithm

The goal of Online Linear Regression is to minimize the square of a linear function in an online setting. It is basically equivalent to stochastic gradient descent, but we are not randomly selecting any data point from the training data set. We will update our model as the new data points will be added in out training data set.So, in SGD, we find out the gradient of the cost function of a single example at each iteration instead of the sum of the gradient of the cost function of all the examples. Stochastic gradient descent is an optimization method for unconstrained optimization problems. In contrast to (batch) gradient descent, SGD approximates the true gradient of $E(w, b)$ by considering a single training example at a time. The algorithm iterates over the training examples and for each example updates the model parameters according to the update rule given by

$$w \longleftarrow w - \eta\left[\alpha\frac{\partial R(w)}{\partial w} + \frac{\partial L(w^Tx_i + b, y_i)}{\partial w}\right] \tag{3}$$

where η is the learning rate which controls the step-size in the parameter space. The intercept b is updated similarly but without regularization.
The advantages of Stochastic Gradient Descent are:

1. Efficiency
2. Ease of Implementation

4.3. Quantile Linear Regression

Quantile regression can predict non-trivial conditional quantiles. Unlike regular linear regression which uses the method of least squares to calculate the conditional mean of the target across different values of the features, quantile regression estimates the conditional median of the target. Quantile regression is an extension of linear regression that is used when the conditions of linear regression are not met (i.e., linearity, homoscedasticity, independence, or normality). For quantile regression, it is not limited to just finding the median, but any quantile (percentage) can be calculated for a particular value in the features variables. For example, if we were to find the 25th quantile for the duration of a delivery, that would mean that there is a 25% chance the actual delivery time is below the prediction, while there is a 75% chance that the delivery time is above.
The quantile regression model equation for τ^{th} quantile is,

$$Q_\tau(y_i) = \beta_0(\tau) + \beta_1(\tau)x_{i1} + \cdots + \beta_p(\tau)x_{ip} \quad i = 1, 2, \ldots, n \tag{4}$$

Where, the given data is $(x_i, y_i)\;\; i = 1, 2, \ldots, n$.
This means that instead of being constants, the beta coefficients are now functions with a dependency on the quantile. Finding the values for these betas at a particular quantile value involves almost the same process as it does for regular linear quantization, except now we have to reduce the median absolute deviation as the loss function. let $q(x)$ be the τ^{th} conditional quantile of y given x,

$$MAD \;= \frac{1}{n}\sum_{i=1}^{n} \rho_\tau(y_i - q(x_i)) \tag{5}$$

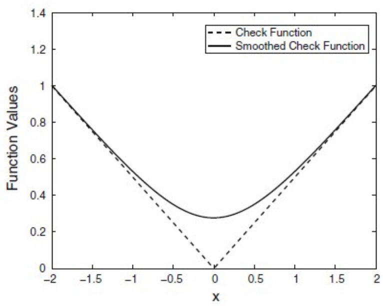
Compared to least square regression, quantile regression is robust to outliers in observations, and can give a more complete view of the relationship between predictor and response. [2]

4.4. Online Quantile Linear Regression Algorithm

The gradient based optimization methods usually converge quickly to a local optimum. However, we notice that the objective function used by quantile regression is not differentiable at the origin which makes the gradient based methods not directly applicable. So, we use an approximate the loss function by a smooth function, based on which gradient descent algorithm can be applied. This is called GDS-QReg model. [6]
The function $\rho_\tau(x)$ which is the main check funcion employed by the original quantile regression model is not differentiable at origin. So, we have used the following smooth function for approximating the check function.

$$S_{\tau,\alpha}(x) = \tau x + \alpha \log(1 + \exp^{-\frac{x}{\alpha}}) \tag{6}$$

where $\alpha > 0$ is called as the smooth parameter. [6]



The gradient vector of the smooth loss function is,

$$\nabla\Phi_\alpha(w) = \frac{1}{n}\sum_{i=1}^{n}\left[\tau - \frac{1}{1 + exp\left(\frac{y_i - x_i^T w}{\alpha}\right)}\right] x_i \tag{7}$$

and this enables us to minimize $\Phi_\alpha(w)$ by gradient descent, resulting the gradient descent smooth quantile regression (GDS-QReg) algorithm.

4.4.1 Gradient Descent Algorithm for Quantile Linear Regression

1. Initialize $w^{[0]}$, and set n = 0
2. Increase n by 1. Compute the gradient vector of $\phi_\alpha(w)$ at $w^{[n-1]}$ by eqn.(7)
3. If the vector $\nabla\Phi(w^{[n-1]})$ is close to 0 vector, stop.
4. Choose a step size $\eta_n \in \mathbf{R}$ and update the estimated vector w as

$$w^{[n]} = w^{[n-1]} - \eta_n \nabla\Phi_\alpha(w^{[n-1]}) \tag{8}$$

and go to step 1. [6]

5. Comparison of Linear Regression, Quantile Linear Regression for both online and offline methods

5.1. Normal Data Set

In normal data set, as we mentioned earlier that the noise is distributed normally. So, mean and median should coincide which we will verify from our analysis.

5.1.1 Linear Regression

There are total 100 data points in the data set. So, while training in online mode, we have incorporated semi-online learning. Like, we have trained the model using traditional linear regression up to first 50 data points then we have updated our model for the rest 50 data points using online learning. The results for both models are comparable. Coefficients are similar,Mean Square Errors are also close.

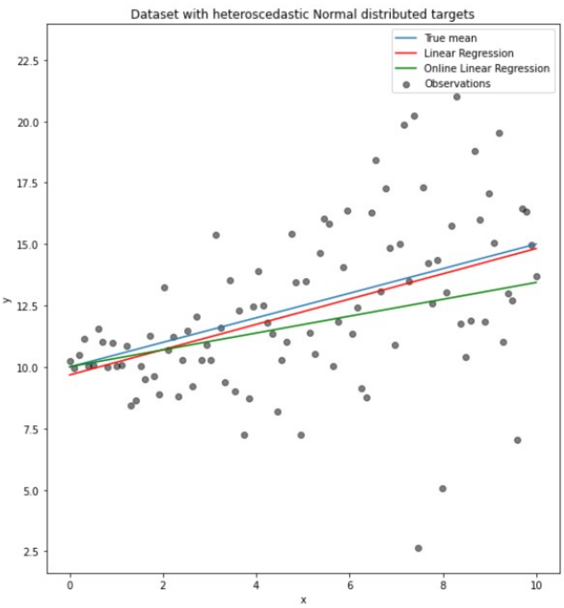
Data Set	Weight	Weight_Online	Bias	Bias_Online
Normal noise	0.514377	0.342886	9.675209	10.009603

Table 1: Coefficient Data Frame for Normal Data

Data Set	MSE	MSE_Online
Normal noise	2.896111	2.985142

Table 2: Mean Square Errors for Normal Data

So, Online Linear Regression can be used instead of traditional Linear Regression. When the data set will contain more data points, online learning reduces the time complexity of training. For time series data, if the data is obtained in each specified intervals, this online learning method is very helpful as we don't need to train the entire model after receiving a new data point.



5.1.2 Quantile Linear Regression

Here, We will find the estimates from 5% to 95% quantiles. We will also compare the Mean Squared Error and Mean Absolute Errors for each quantiles.

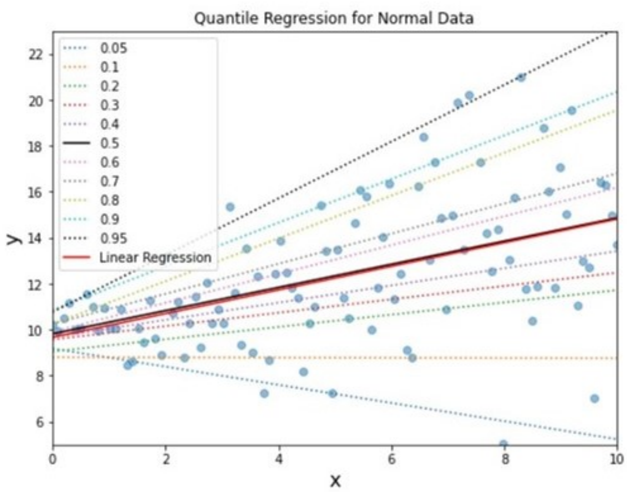
Quantiles	Weight	Bias	Weight Online	Bias Online
0.05	9.182897	-0.394532	8.546921	-0.079947
0.10	8.803792	-0.004115	8.803782	-0.004130
0.20	9.051236	0.267287	9.262631	0.138225
0.30	9.574339	0.291115	9.832591	0.158419
0.40	9.676854	0.374668	9.676919	0.375087
0.50	9.821716	0.504138	9.841111	0.467141
0.60	9.910968	0.627872	9.754289	0.698114
0.70	10.248351	0.654820	10.021099	0.701503
0.80	10.248358	0.930664	10.248377	0.930534
0.90	10.856092	0.949410	10.856100	0.949204
0.95	10.769471	1.235259	11.009324	0.928364

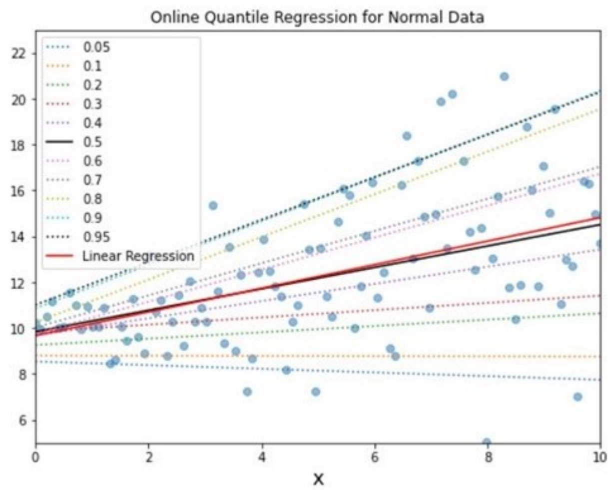
Table 3: Coefficient Data Frame for different Quantiles

Quantiles	MAE	MSE	MAE_Online	MSE_Online
0.05	5.142231	40.780760	4.316580	28.199660
0.10	3.781359	22.671435	3.781429	22.672166
0.20	2.647569	12.363957	2.927552	14.849764
0.30	2.329236	10.292757	2.533569	12.096871
0.40	2.163666	9.039065	2.163249	9.035072
0.50	2.092697	8.397434	2.097202	8.411367
0.60	2.171494	9.142158	2.220367	9.670002
0.70	2.304180	10.181682	2.309248	10.327454
0.80	3.064768	16.907584	3.064390	16.903314
0.90	3.564262	21.259482	3.563456	21.251118
0.95	4.726851	34.883042	3.604264	21.432105

Table 4: MAE and MSE Data Frame for different Quantiles

Here, we can see that at 50% quantile both MAE and MSE attains minimum. So, here the noise is normally distributed and the mean and median coincides. Graphically,





5.2. Pareto Data Set

In normal data set, as we mentioned earlier that the noise is distributed following pareto distribution. So, mean and median should not coincide which we will verify from our analysis.

5.2.1 Linear Regression

There are total 100 data points in the data set. So, while training in online mode, we have incorporated semi-online learning. Like, we have trained the model using traditional linear regression up to first 50 data points then we have updated our model for the rest 50 data points using online learning. The results for both models are comparable. Coefficients are similar,Mean Square Errors are also close.

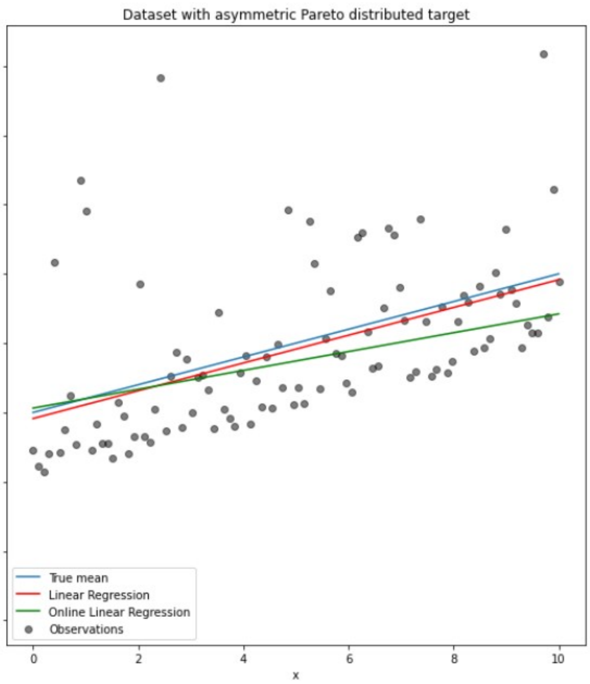
Data Set	Weight	Weight_Online	Bias	Bias_Online
Pareto Noise	0.500700	0.339790	9.780902	10.157114

Table 5: Coefficient DataFrame for Pareto Data

Data Set	MSE	MSE_Online
Pareto Noise	2.546756	2.624798

Table 6: Mean Square Errors for Pareto Data

After training in online mode, we can see that MSE is very close to the Linear Regression model. So, definitely we can use this online learning model for obtaining efficient training as we discussed for Normal Data Set. But, in the next section we will show that Quantile Linear Regression is working better because the noise is not normal.



From graph it is clear that majority number of points are lying under the MSE Estimate. So, lets check it using Quantile Regression.

5.2.2 Quantile Linear Regression

Here, We will find the estimates from 5% to 95% quantiles. We will also compare the Mean Squared Error and Mean Absolute Errors for each quantiles.

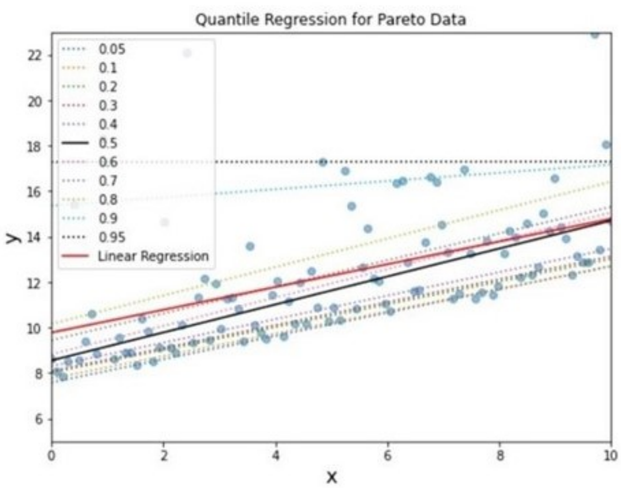
Quantiles	Weight	Bias	Weight Online	Bias Online
0.05	7.583786	0.512362	7.583786	0.512362
0.10	7.759982	0.493403	7.759982	0.493403
0.20	8.015095	0.500707	8.015095	0.500707
0.30	8.092188	0.504584	8.092188	0.504584
0.40	8.303639	0.516666	8.303639	0.516666
0.50	8.553998	0.615352	8.553998	0.615352
0.60	8.824240	0.623867	8.824240	0.623867
0.70	9.441583	0.587189	9.441583	0.587189
0.80	10.159515	0.625649	10.159515	0.625649
0.90	15.357886	0.181337	15.357886	0.181337
0.95	17.281067	0.002392	17.281067	0.002392

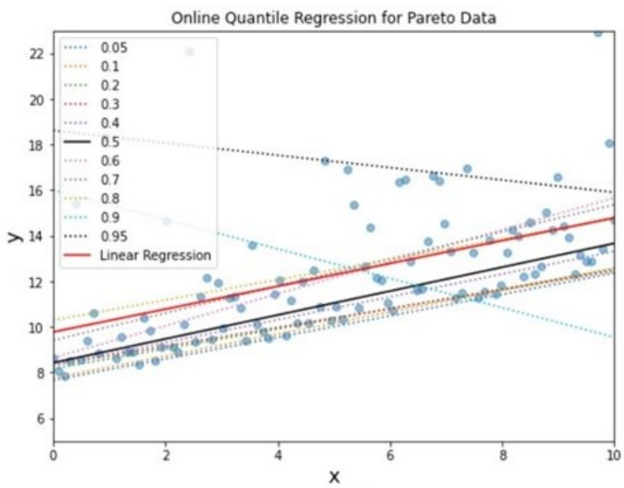
Table 7: Coefficient Data Frame for different Quantiles

Quantiles	MAE	MSE	MAE_Online	MSE_Online
0.05	2.148932	11.061632	2.148932	11.061632
0.10	2.080609	10.719347	2.080609	10.719347
0.20	1.883737	9.603931	1.883737	9.603931
0.30	1.835647	9.272651	1.835647	9.272651
0.40	1.745789	8.440961	1.745789	8.440961
0.50	1.670429	7.024976	1.670429	7.024976
0.60	1.699223	6.731102	1.699223	6.731102
0.70	1.824780	6.558235	1.824780	6.558235
0.80	2.299910	7.625421	2.299910	7.625421
0.90	4.372168	23.194800	4.372168	23.194800
0.95	5.254020	33.683355	5.254020	33.683355

Table 8: MAE and MSE Data Frame for different Quantiles

Due to the asymmetry of the distribution of the noise, we observe that the true mean and estimated conditional median are different. We also observe that each quantile model has different parameters to better fit the desired quantile. Note that ideally, all quantiles would be parallel in this case, which would become more visible with more data points or less extreme quantiles, e.g. 10% and 90%.





5.3. mtcars Data Set

Now, we will use our Online Learning Algorithms on two real data set and check if it is working fine for both Linear and Quantile Regression.

5.3.1 Linear Regression

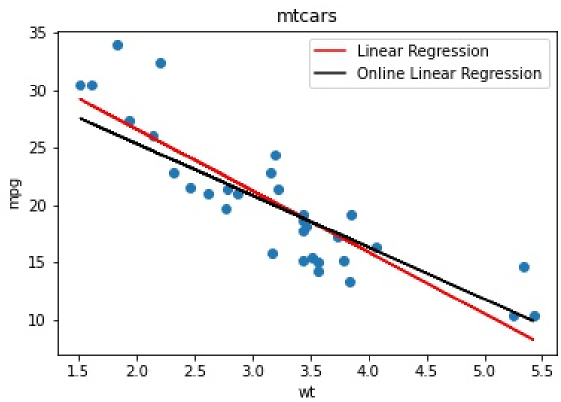
This mtcars data set is a very small data set containing only 32 data points. But out of many variables here we have chosen one feature 'wt'(weight) and modelled 'mpg' (miles per gallon) as a function of weight. While training in Semi-Online mode we have trained the first 11 points using Linear Regression and then applying Online Learning for the rest data points. The results are comparable.

Data Set	Weight	Weight_Online	Bias	Bias_Online
mtcars	-5.344472	-4.507312	37.285126	34.363265

Table 9: Coefficient DataFrame for Pareto Data

Data Set	MSE	MSE_Online
mtcars	2.949163	3.065905

Table 10: Mean Square Errors for Pareto Data



Now, we will explore the Quantile Linear Regression and try to draw some interesting inferences from the data.

5.3.2 Quantile Linear Regression

Here, We will find the estimates from 5% to 95% quantiles. We will also compare the Mean Squared Error and Mean Absolute Errors for each quantiles.

From MSE and MAE it is evident that the mean and the median coincides. Online Learning is also working fine producing similar results to the traditional method.

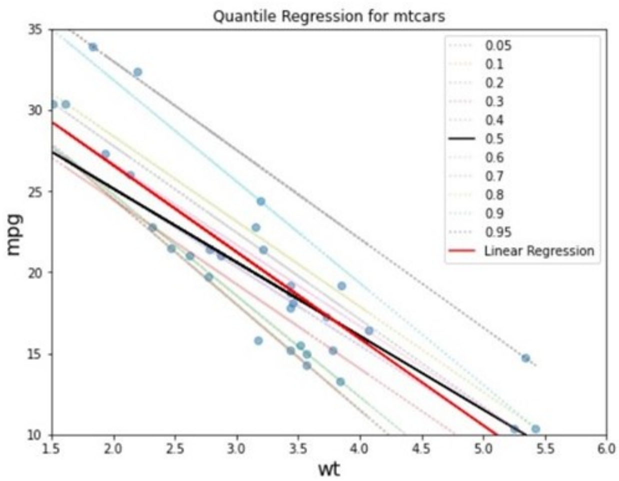
Now, I will explain what does it mean by 5% and 95% quantile. As we have aready discussed in the theory that whenever we are predicting 5% quantile, we can say that there is a probability that 5% of the entire data set will lie below that line. So, here we can say that it is the lower bound. Specifically, for a car weighing at least 1.5 ton to at most 4 ton should result into at most 27 mpg to at least 10 mpg. Predicting 95% quantile retrieves, a car which weighing at least 2.5 ton to at most 6 ton, should result into at most 35 mpg to at least 10 mpg. We can say this with 95% confidence. Figure below, is the reference for the explanation.

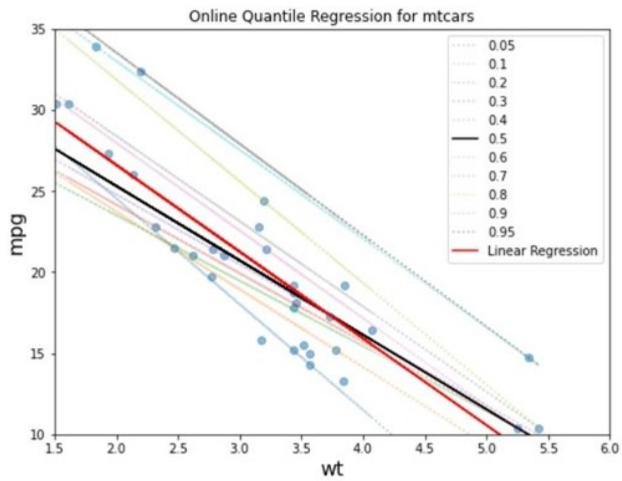
Quantiles	Weight	Bias	Weight Online	Bias Online
0.05	37.561538	-6.515837	37.561489	-6.515990
0.10	37.509794	-6.494845	33.309016	-4.792580
0.20	37.276798	-6.239999	31.559318	-4.031799
0.30	34.876712	-5.205479	32.618159	-4.233136
0.40	34.891171	-4.852940	33.441261	-4.328493
0.50	34.232244	-4.539476	34.474367	-4.586298
0.60	36.734407	-5.016078	38.778028	-5.406320
0.70	38.497403	-5.351886	38.879690	-5.251316
0.80	38.879916	-5.250722	44.390825	-6.267436
0.90	44.391048	-6.266786	43.937500	-5.470412
0.95	43.937608	-5.470086	44.781502	-5.628143

Table 11: Coefficient Data Frame for different Quantiles

Quantiles	MAE	MSE	MAE_Online	MSE_Online
0.05	3.561309	22.165330	3.561816	22.169439
0.10	3.547893	22.010086	2.821803	13.822411
0.20	3.123265	17.790365	2.571117	12.553519
0.30	2.701224	12.561946	2.425579	11.034460
0.40	2.359816	9.581914	2.340055	9.985762
0.50	2.321542	9.512949	2.321925	9.368721
0.60	2.391582	9.053420	2.657293	10.375340
0.70	2.607877	10.109960	2.948550	12.293868
0.80	2.949586	12.302067	4.550174	26.596540
0.90	4.551687	26.614572	6.278439	47.739662
0.95	6.279385	47.754056	6.583816	52.117727

Table 12: MAE and MSE Data Frame for different Quantiles





One more observation, The lower quantiles are very dense compared to the higher quantiles. We can also infer that, there are very few cars which can obtain higher mpg even if there weight is less.

5.4. Old Faithful Geyser Data Set

Finally, we will discuss about the implementation of our online and traditional Linear and Quantile Linear Regression models on Old Faithful Geyser data set.

5.4.1 Linear Regression

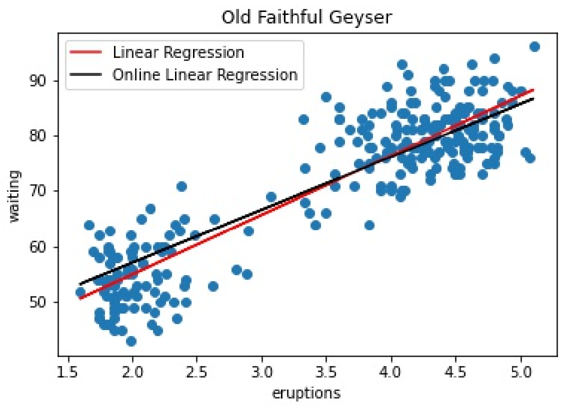
The faithful data set contains 272 observations on 2 variables. This data set describes waiting time between eruptions and the duration of the eruption. According to the scatter plot, when the duration of the eruption is more, the waiting time for the next eruption is also more and thereby we can see a linear relationship which we will model using Linear Regression. Both Online and traditional method is working fine which we can see from the coefficient Data Frame as well as from Mean Squared Error Data Frame.

Data Set	Weight	Weight_Online	Bias	Bias_Online
faithful	10.729641	9.535176	33.474397	37.981623

Table 13: Coefficient DataFrame for Pareto Data

Data Set	MSE	MSE_Online
faithful	5.892227	6.056944

Table 14: Mean Square Errors for Pareto Data



But here we wish to analyze the data with a particular confidence. Some one might ask, what will be the waiting time when the eruption lasts for 5 minutes? To answer this question we need to model it with Quantile Regression.

5.4.2 Quantile Linear Regression

Here, We will find the estimates from 5% to 95% quantiles. We will also compare the Mean Squared Error and Mean Absolute Errors for each quantiles.This faithful data is normally distributed as we have seen from the histogram. So, the mean and median should coincide. So, estimates for 50% quantile should be approximately equal to OLS estimates.

If we consider the weight and bias for each quantiles and put the corresponding values of x we can get the prediction of y from 5% to 95% quantiles.

When, x = 5, the estimates of y are given in Table 16. So, from here we can say that when the duration of the eruption is 5 minutes then the duration will range from 75.64 minutes to 96.75 minutes.

Quantiles	Weight	Bias	Weight Online	Bias Online
0.05	24.577418	10.653409	31.523487	8.823535
0.10	26.906245	10.416669	30.020932	9.700250
0.20	28.143721	10.697674	31.853563	9.816833
0.30	29.828331	10.804322	33.435492	9.758823
0.40	31.547896	10.743062	36.890907	9.441688
0.50	34.729160	10.416669	37.140207	9.729967
0.60	34.883717	10.697677	41.128161	8.941265
0.70	37.094082	10.452959	40.839812	9.499671
0.80	37.818180	10.909092	42.352789	9.712393
0.90	39.836324	11.272736	45.833975	9.368102
0.95	43.567455	10.706637	47.614234	9.828308

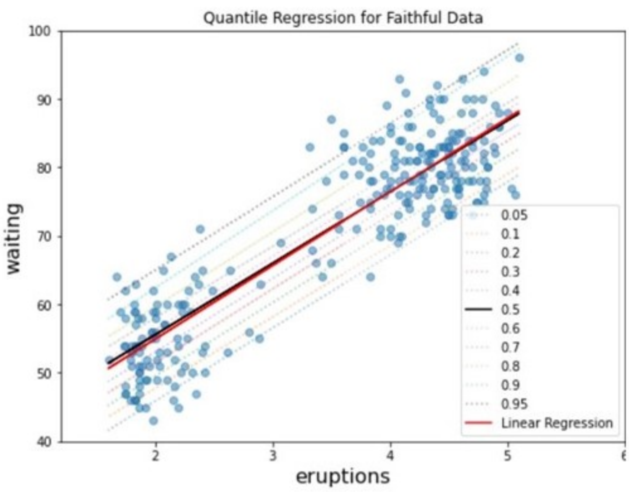
Table 15: Coefficient Data Frame for different Quantiles

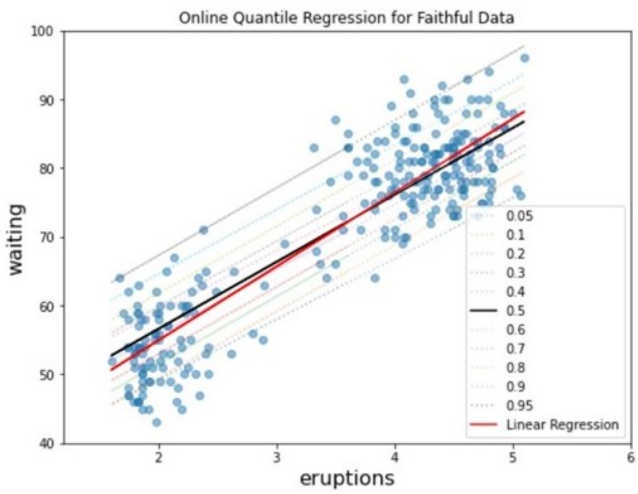
Quantiles	0.05	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	0.95
y_hat	75.64	78.52	80.93	82.22	84.09	85.79	85.83	88.33	90.91	92.67	96.75

Table 16: Values of Quantile estimates when x = 5

Quantiles	MAE	MSE	MAE_Online	MSE_Online
0.05	9.296250	118.683929	8.974184	113.376768
0.10	8.015430	93.516939	7.602203	85.708227
0.20	6.453767	64.336876	6.176390	58.883129
0.30	5.448242	46.187835	5.557794	47.671632
0.40	5.008874	38.251816	4.959638	38.028281
0.50	4.768497	34.872099	4.850150	36.047527
0.60	4.863125	36.404033	5.155619	40.875411
0.70	5.282830	41.865000	5.518918	46.140864
0.80	6.450388	59.457720	6.735531	64.475118
0.90	8.761401	103.264743	8.337043	95.049093
0.95	10.254480	134.975683	11.153196	156.688752

Table 17: MAE and MSE Data Frame for different Quantiles





6. Conclusion

From the above experiments we have seen that,

1. For some particular cases Quantile Linear Regression works better than the Linear Regression.
2. Online Learning with partial offline training with Linear Regression is useful and reduced time complexity of the model.

7. Appendix

The code for this paper is hosted in a github repository.
https://github.com/bonnya15/Online_learning_for_Linear_and_Quantile_Regression/tree/master/code

References

[1] Ting Hu, Dao-Hong Xiang, and Ding-Xuan Zhou. Online learning for quantile regression and support vector regression. *Journal of Statistical Planning and Inference*, 142(12):3107–3122, 2012.

[2] Roger Koenker. *Frontmatter*. Econometric Society Monographs. Cambridge University Press, 2005.

[3] Jacob Montiel, Max Halford, Saulo Martiello Mastelini, Geoffrey Bolmier, Raphael Sourty, Robin Vaysse, Adil Zouitine, Heitor Murilo Gomes, Jesse Read, Talel Abdessalem, and Albert Bifet. River: machine learning for streaming data in python, 2020.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[5] Kilian Q. Weinberger, Anirban Dasgupta, Josh Attenberg, John Langford, and Alexander J. Smola. Feature hashing for large scale multitask learning. *CoRR*, abs/0902.2206, 2009.

[6] Songfeng Zheng. Gradient descent algorithms for quantile regression with smooth approximation. *International Journal of Machine Learning and Cybernetics*, 2(191), 2011.