# cs229 milestone LWR

jballouz

November 2018

## 1 Locally Weighted Regression

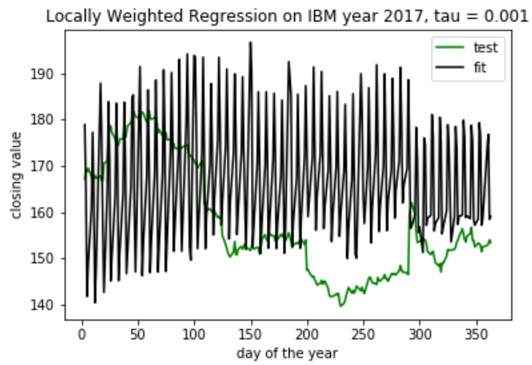
In this section we use locally weighted regression to try and predict the daily closing values of the DOW JONES, the SP 500 index, and the stocks for the company IBM, for the year 2017. Thus, our test set will be comprised of the year 2017. We use as our development set the year 2016 to fine tune which $\tau$ value gives the best fit (the least mean square error MSE). Note that the weight we associated with each data point is $w^{(i)} = exp(\frac{-\|x^{(i)}-x\|_2^2}{365^2\tau^2})$ (this formulation was found to be better suited to avoid singular weight matrices). For the training set, although we could include all the previous years (for the DOW JONES from 1900 to 2015, for the S&P 500 from 1950 to 2015, and for IBM from 1962 to 2015) we found that this does not help our fit at all because all those indices follow a general upward trend. So less training years is actually better which is already an indication that locally weighted regression is not well suited for this time series prediction, and another tool like a recurrent neural network (RNN) would be better suited (more on that later). Thus, for our training set we will use the daily closing values from 2009 to 2015.

### 1.1 IBM

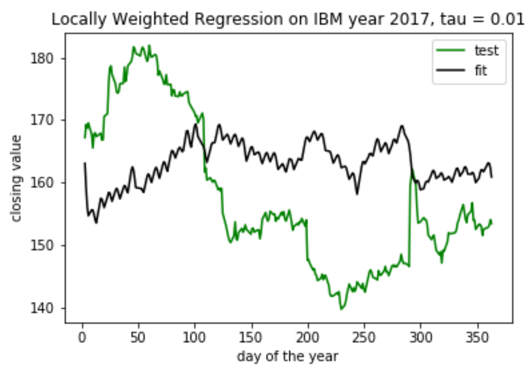
The $\tau$ and the corresponding MSE values are shown below:

| $\tau$ | 0.0006 | 0.0008 | 0.001 | 0.005 | 0.01 | 0.05 | 0.1 |
|---|---|---|---|---|---|---|---|
| MSE | 223.6 | 223.13 | 223.11 | 238.9 | 260.6 | 266.1 | 263.5 |

Thus according to the dev set, it is best to use $\tau = 0.001$, and using this value for $\tau$ the prediction on the test set (daily closing values for the year 2017) is shown below (note MSE = 450.1):




However, from observing the prediction on the dev set, we noticed that $\tau = 0.01$ is better suited (even though its MSE is higher). Using $\tau = 0.01$ the prediction on the test set is shown below (note MSE = 216.4):

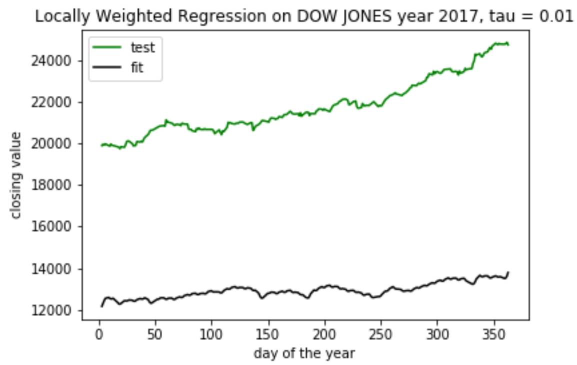



## 1.2 DOW JONES

The $\tau$ and the corresponding MSE values are shown below:

| $\tau$ | 0.0006 | 0.0008 | 0.001 | 0.005 | 0.01 | 0.05 | 0.1 |
|---|---|---|---|---|---|---|---|
| MSE | $26.6 \times 10^6$ | $26.6 \times 10^6$ | $26.6 \times 10^6$ | $25.6 \times 10^6$ | $25.2 \times 10^6$ | $25.2 \times 10^6$ | $25.2 \times 10^6$ |

Thus it is best to use $\tau = 0.01$, and using this value for $\tau$ the prediction on the test set (daily closing values for the year 2017) is shown below (note MSE $= 79 \times 10^6$):

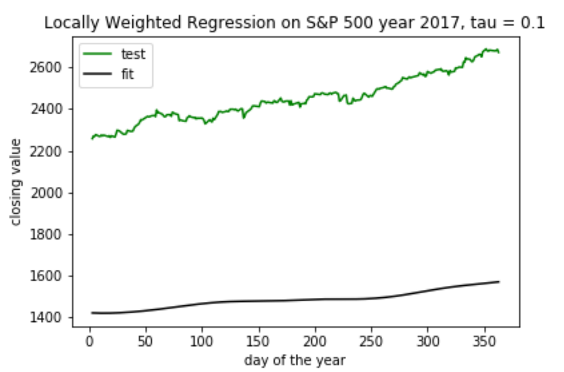



## 1.3 S&P 500

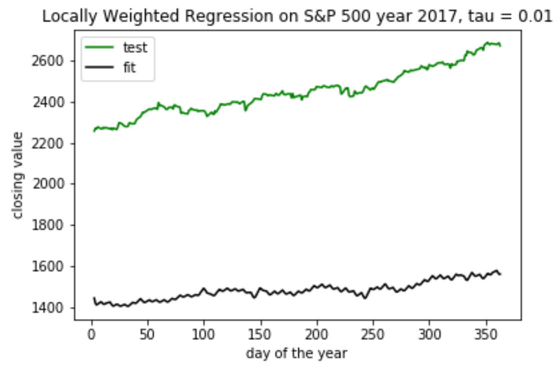
The $\tau$ and the corresponding MSE values are shown below:

| $\tau$ | 0.0006 | 0.0008 | 0.001 | 0.005 | 0.01 | 0.05 | 0.1 |
|---|---|---|---|---|---|---|---|
| MSE | $436.7 \times 10^3$ | $436.3 \times 10^3$ | $436.3 \times 10^3$ | $399.3 \times 10^3$ | $382.4 \times 10^3$ | $379.7 \times 10^3$ | $379.6 \times 10^3$ |

Thus according to the dev set, it is best to use $\tau = 0.1$, and using this value for $\tau$ the prediction on the test set (daily closing values for the year 2017) is shown below (note MSE $= 941 \times 10^3$):




However, from observing the prediction on the dev set, we noticed that $\tau = 0.01$ is better suited (even though its MSE is higher). Using $\tau = 0.01$ the prediction on the test set is shown below (note MSE $= 945 \times 10^3$):




## 2 Neural Networks

The previous sections is evidence that locally weighted regression is obviously not suited for this time series prediction because all three indices show a strong general upward trend. Thus, if the training set is years 2009 to 2015, the prediction for 2017 will be somewhere around the average values of 2009-2015, whereas the actual values for 2017 will be higher than that (that's why we see that our predictions are mostly below the actual values). In order to solve this issue, we turn to recurrent neural networks, which we hope will understand this general upward trend and make better predictions. The network architecture we first plan to try is a many-to-many RNN architecture where $T_x$ = number of training points $\neq T_y$ = number of prediction points and we will include LSTM (long short-term memory) units or GRU (gated recurrent unit) to capture the long term data trend as much as possible.