資工系100062236林修安

Machine Learning Assignment 1 Report

- Implementation

+ Linear Regressor

- `train(X, y)`:

Take the partial derivatives of the objective $emp(\theta;\chi)=\sum_{t=1}^{N}\left(r^{(t)}-w^{T}\begin{bmatrix}1\\x^{(t)}\end{bmatrix}\right)^{2}$ respectively to w_0 and w_1 to obtain

$$A=\begin{pmatrix} N & \sum_{t=1}^{N}x^{(t)} \\ \sum_{t=1}^{N}x^{(t)} & \sum_{t=1}^{N}(x^{(t)})^{2}\end{pmatrix} \quad \text{and} \quad y=\begin{pmatrix}\sum_{t=1}^{N}r^{(t)} \\ \sum_{t=1}^{N}r^{(t)}x^{(t)}\end{pmatrix}$$

so we can solve w by $w=A^{-1}y$.

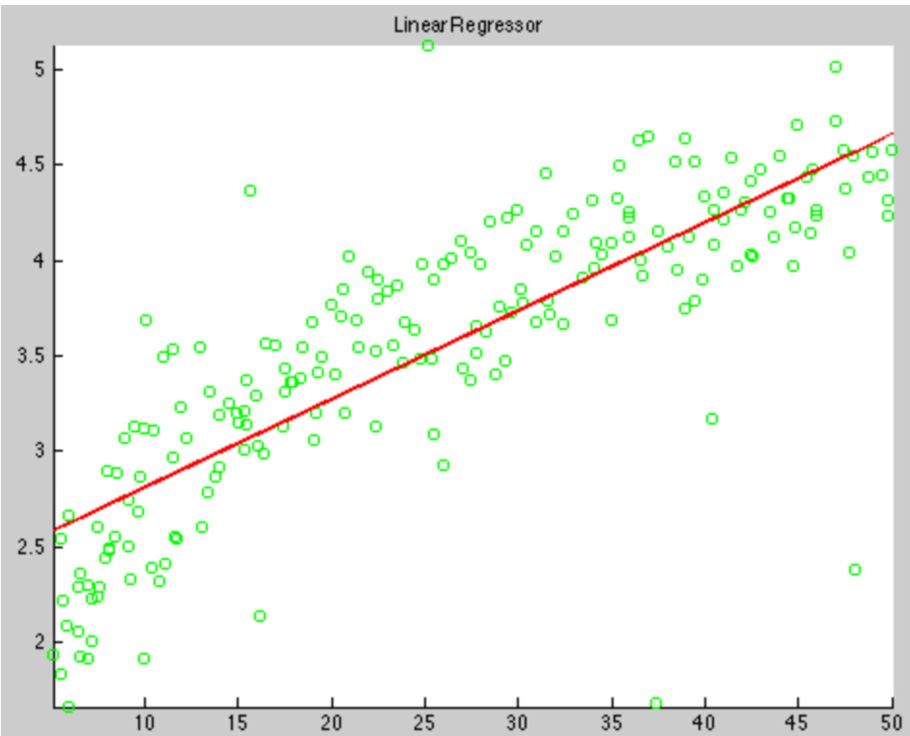
Set `A = [size(y,1) sum(X);sum(X) X'*X]` and `Y = [sum(y);X'*y]`, then by `theta = A\Y`, where `theta` is the parametric vector w.

- `predict(X)` :

Obtain the predicted values by

$w^{T}\begin{bmatrix}1\\x^{(t)}\end{bmatrix} \longrightarrow$ `obj.w'*[ones(1,length(X));X'].`

- Result:



+ Locally Weighted Linear Regressor

- `train(X, y)`:
Simply save the original X and y.

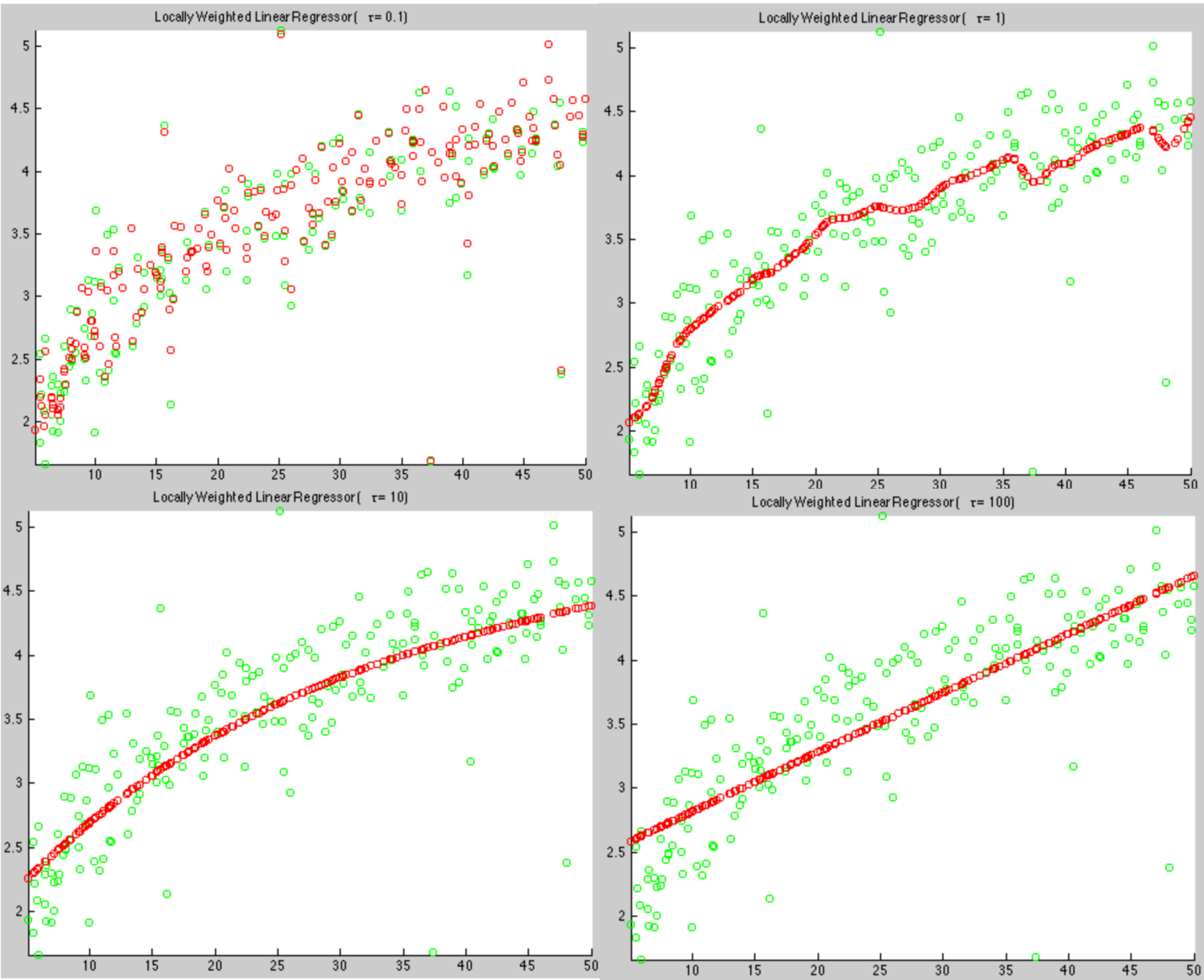
- `CalculateWeight(~, X, tau, newX)`:
Calculate the weight vector of every instance of the new input X.
`weight = exp(-((X-newX).^2)/(2*(tau^2)));`

- `predict(X, cfg)`:
Take the partial derivatives of the objective $\sum_{t=1}^{N} l^{(t)}(r^{(t)} - w^T \begin{bmatrix} 1 \\ x^{(t)} \end{bmatrix})^2$ respectively to w_0 and w_1 to obtain

$$A = \begin{pmatrix} \sum_{t=1}^{N} l^{(t)} & \sum_{t=1}^{N} l^{(t)} x^{(t)} \\ \sum_{t=1}^{N} l^{(t)} x^{(t)} & \sum_{t=1}^{N} l^{(t)} (x^{(t)})^2 \end{pmatrix} \quad \text{and} \quad Y = \begin{pmatrix} \sum_{t=1}^{N} l^{(t)} r^{(t)} \\ \sum_{t=1}^{N} l^{(t)} x^{(t)} r^{(t)} \end{pmatrix}$$

so we can solve w by $w = A^{-1} y$.

- Results:



- Code Execution

Call `main` in the command window, and it outputs 5 plots as the above.

- Discussion

$$l(x^{(t)}; x') = \exp(-\frac{(x^{(t)} - x')^2}{2\tau^2}) = \frac{1}{e^{\frac{(x^{(t)} - x')^2}{2\tau^2}}}$$

$\frac{1}{e^{\frac{(x^{(t)} - x')^2}{2\tau^2}}}$ approaches to 1 as $\tau \to \infty$, making local weights become 1.

So $\sum_{t=1}^{N} l^{(t)}(r^{(t)} - w^T \begin{bmatrix} 1 \\ x^{(t)} \end{bmatrix})^2$ actually becomes $\sum_{t=1}^{N} (r^{(t)} - w^T \begin{bmatrix} 1 \\ x^{(t)} \end{bmatrix})^2$, the objective function of linear regression, resulting the line to be the same as linear regression.

$\frac{1}{e^{\frac{(x^{(t)} - x')^2}{2\tau^2}}}$ approaches to 0 as $\tau \to 0$, making local weights become 0.

So local weight influences the result line only when $x^{(t)} - x' \to 0$, meaning that only instance extremely close to the original data has local weight. The predicted values are influenced only by data really close to the instance.