# Locally weighted linear regression

Wednesday, August 12, 2020 2:29 PM

5. **[25 points] Locally weighted linear regression**

(a) [10 points] Consider a linear regression problem in which we want to "weight" different training examples differently. Specifically, suppose we want to minimize

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m} w^{(i)}\left(\theta^T x^{(i)} - y^{(i)}\right)^2.$$

In class, we worked out what happens for the case where all the weights (the $w^{(i)}$'s) are the same. In this problem, we will generalize some of those ideas to the weighted setting.

i. [2 points] Show that $J(\theta)$ can also be written

$$J(\theta) = (X\theta - y)^T W (X\theta - y)$$

for an appropriate matrix $W$, and where $X$ and $y$ are as defined in class. Clearly specify the value of each element of the matrix $W$.

ii. [4 points] If all the $w^{(i)}$'s equal 1, then we saw in class that the normal equation is

$$X^T X\theta = X^T y,$$

and that the value of $\theta$ that minimizes $J(\theta)$ is given by $(X^TX)^{-1}X^Ty$. By finding the derivative $\nabla_\theta J(\theta)$ and setting that to zero, generalize the normal equation to this weighted setting, and give the new value of $\theta$ that minimizes $J(\theta)$ in closed form as a function of $X$, $W$ and $y$.

iii. [4 points] Suppose we have a dataset $\{(x^{(i)}, y^{(i)}); i = 1\ldots, m\}$ of $m$ independent examples, but we model the $y^{(i)}$'s as drawn from conditional distributions with different levels of variance $(\sigma^{(i)})^2$. Specifically, assume the model

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma^{(i)}}\exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2(\sigma^{(i)})^2}\right)$$

That is, each $y^{(i)}$ is drawn from a Gaussian distribution with mean $\theta^T x^{(i)}$ and variance $(\sigma^{(i)})^2$ (where the $\sigma^{(i)}$'s are fixed, known, constants). Show that finding the maximum likelihood estimate of $\theta$ reduces to solving a weighted linear regression problem. State clearly what the $w^{(i)}$'s are in terms of the $\sigma^{(i)}$'s.

**Answer:**

$$(i)\quad X = \begin{array}{c c} & \begin{array}{ccc} x_1 & \cdots & x_n \end{array} \\ \begin{array}{c} 1 \\ \vdots \\ m \end{array} & \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \end{array}$$

$$W = \begin{bmatrix} \frac{w^{(1)}}{2} & & 0 \\ & \frac{w^{(2)}}{2} & \\ 0 & & \ddots \quad \frac{w^{(m)}}{2} \end{bmatrix}$$

st, $$W(X\theta - y) = \begin{bmatrix} \frac{w^{(1)}}{2}(\theta^T x^{(1)} - y^{(1)}) \\ \vdots \\ \frac{w^{(m)}}{2}(\theta^T x^{(m)} - y^{(m)}) \end{bmatrix}$$

(ii)
$$\nabla_\theta J(\theta) = \nabla_\theta (X\theta - y)^T W (X\theta - y)$$
$$= \nabla_\theta (\theta^T X^T - y^T)(WX\theta - Wy)$$
$$= \nabla_\theta (\theta^T X^T W X\theta - \theta^T X^T W y - y^T W X\theta + y^T W y)$$
$$= \nabla_\theta (\theta^T X^T W X\theta - 2y^T W X\theta + y^T W y)$$
$$= 2X^T W X\theta - 2X^T W^T y = 0$$

$$X^T W X \theta = X^T W^T y$$

$$\theta = (X^T W X)^{-1} X^T W y$$

(ii) $$l(\theta) = \sum_{i=1}^{m} -\log\sqrt{2\pi} - \log\sigma^{(i)} - \frac{1}{2(\sigma^{(i)})^2}(y^{(i)} - \theta^T x^{(i)})^2)$$

Maximizing $l(\theta)$ is equivalent to minimizing $J\theta = \sum_{i=1}^{m} \frac{1}{2(\sigma^{(i)})^2}(y^{(i)} - \theta^T x^{(i)})^2$

so, $w^{(i)} = \frac{1}{(\sigma^{(i)})^2}$

(b) [10 points] **Coding problem.** We will now consider the following dataset (the formatting matches that of Datasets 1-4, except $x^{(i)}$ is 1-dimensional):

`data/ds5_{train,valid,test}.csv`

In `src/p05b_lwr.py`, implement locally weighted linear regression using the normal equations you derived in Part (a) and using

$$w^{(i)} = \exp\left(-\frac{\|x^{(i)} - x\|_2^2}{2\tau^2}\right).$$

Train your model on the `train` split using $\tau = 0.5$, then run your model on the `valid` split and report the mean squared error (MSE). Finally plot your model's predictions on the validation set (plot the training set with blue 'x' markers and the validation set with a red 'o' markers). Does the model seem to be under- or overfitting?

**Answer:**

(c) [5 points] **Coding problem.** We will now tune the hyperparameter $\tau$. In `src/p05c_tau.py`, find the MSE value of your model on the validation set for each of the values of $\tau$ specified in the code. For each $\tau$, plot your model's predictions on the validation set in the format described in part (b). Report the value of $\tau$ which achieves the lowest MSE on the `valid` split, and finally report the MSE on the `test` split using this $\tau$-value.

**Answer:**