Locally Weighted & Logistic Regression

Jerry Peng

1 Locally Weighted Regression

Locally Weighted Regression (LWR), also known as Local Regression or LOESS/LOWESS (Locally Estimated Scatterplot Smoothing), is a type of regression that focuses on fitting simple models to localized subsets of the data to form a regression model. Unlike traditional linear regression that fits a global model to the data, LWR fits multiple models, each one tailored to a specific region of the data.

Weighted Least Squares in LWR In LWR, the key is to fit a linear model to the data, but with a twist: each point in the dataset is weighted based on its distance to the point where the prediction is to be made. The closer a point is to the prediction point, the more influence it has on the model. This process is repeated for each point where a prediction is needed. The mathematical formulation of LWR typically involves solving a weighted least squares problem. For a given prediction point x, the local linear model is:

$$\hat{y}(x) = \beta_0 + \beta_1 x$$

The goal is to find β_0 and β_1 that minimize the weighted sum of squared residuals:

$$\min_{\beta_0,\beta_1} \sum_{i=1}^{n} w_i(x)(y_i - (\beta_0 + \beta_1 x))^2$$

Weight Function The weight $w_i(x)$ are calculated using a kernel function. A common choice is the Gaussian kernel:

$$w_i(x) = \exp(-\frac{(x - x_i)^2}{2\lambda^2})$$

where

- $w_i(x)$ is the weight for the i-th data point when making a prediction at point x
- λ is a bandwidth parameter that determines the width of the neighborhood around x. A smaller λ results in a narrower, more localized weight distribution.

Example for LWR Here is an example figure for LWR. The synthetic data is generated using the function $y = \sin(x)$, with a random noise added to every fifth value. Mathematically, this can be expressed as:

$$y_i = \sin(x_i) + \epsilon_i$$

The figure 1 shows how the LWR fit the data and make predictions.

2 Logistic Regression

Logistic Regression is a type of generalized linear model used for predicting the likelihood of a certain phenomenon. Unlike linear regression analysis, it does not predict numerical output but rather predicts the probability between two or more categories. The most common use is for binary classification problems, where the output has only two categories.

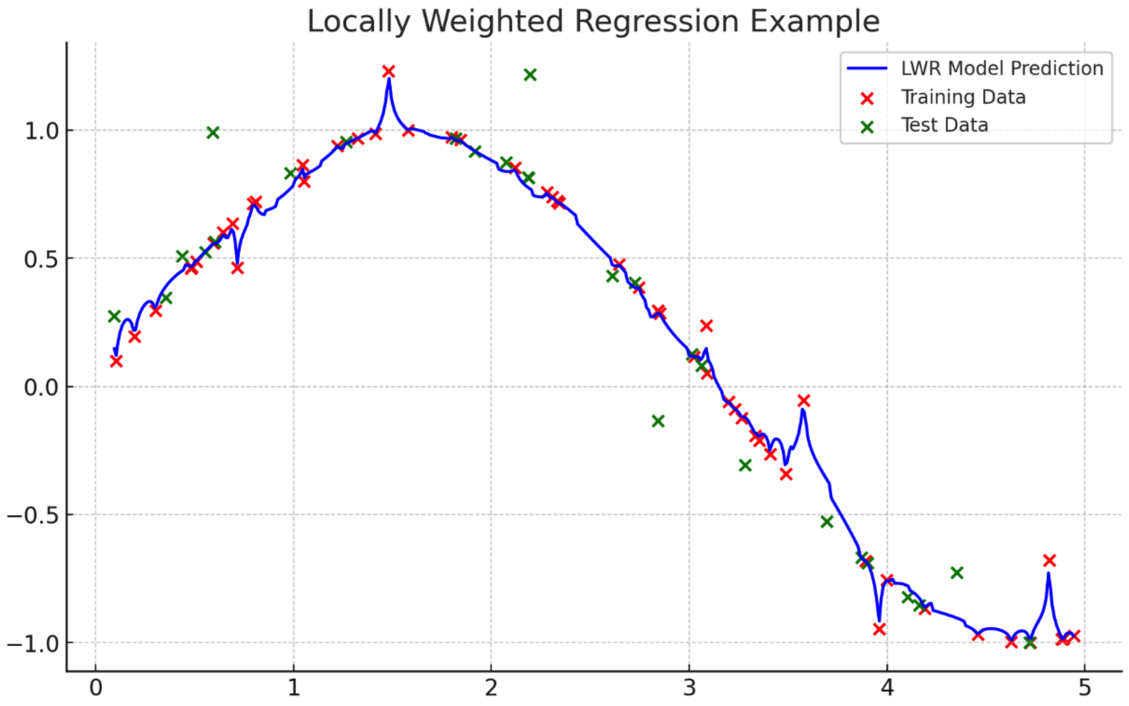

Figure 1: Locally Weighted Regression Example

Output Logistic Regression predicts a probability, which describes the likelihood of a certain category occurring. For example, it can predict the probability of an email being spam.

$$y_i = \begin{cases} 1 & \text{if } X_i\theta > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Sigmoid Function Logistic Regression uses the Sigmoid function to constrain its predicted values between 0 and 1. The formula for the Sigmoid function is:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{2}$$

z is the input or the raw predicted value.

Training In Logistic Regression, our goal is to find a parameter vector θ such that for positive samples ($y = 1$), $X\theta$ is positive and as large as possible, and for negative samples ($y = 0$), $X\theta$ is negative and as small as possible. This can also be understood as follows:

- **Positive samples** ($y = 1$): For positive samples, we want $\sigma(z)$to be as close to 1 as possible. This means that $X\theta$ should be a large positive number, as the output of the Sigmoid function approaches 1 when its input increases.
- **Negative samples** ($y = 0$): For negative samples, we want $\sigma(z)$to be as close to 0 as possible. This means that $X\theta$ should be a large negative number, as the output of the Sigmoid function approaches 0 when its input decreases.

The formula to achieve this can be represented as follows:

$$\arg\max_{\theta} \prod_i \sigma(y_i = 1)P_\theta(y_i|X_i) + \sigma(y_i = 0)(1 - P_\theta(y_i|X_i)) \tag{3}$$

By aggregating all y_i and X_i into datasets y and X, the likelihood formula mentioned earlier can be rewritten as:

$$L_\theta(y|X) = \prod_{y_i=1} P_\theta(y_i|X_i) \prod_{y_i=0} (1 - P_\theta(y_i|X_i)) \tag{4}$$

Optimization We optimize Logistic Regression using **Gradient Ascent**. The optimization algorithm is as follows:

1. First, we take the logarithm of equation 4. The reasons for taking the logarithm include the following:

   (a) **Simplification of calculations**: Taking the logarithm of the likelihood function transforms products into sums, making the calculation of gradients and optimization simpler and more efficient.

   (b) **Numerical stability**: Using the likelihood function directly could lead to numerical issues like overflow or underflow, especially when the probability values are very close to 0 or 1. By using the logarithm, we can avoid these numerical problems.

   After taking the logarithm, the likelihood function becomes:

$$l_\theta(y|X) = \sum_{i=1}^{n}[y_i \log(P_\theta(y_i|X_i)) + (1-y_i)\log(1-P_\theta(y_i|X_i))] \tag{5}$$

2. To prevent overfitting, a regularizer term is subtracted from the likelihood function. L2 regularization is commonly used. Note that we are using the Gradient Ascent algorithm here, so we subtract the regularizer term. The updated likelihood formula is as follows:

$$J_\theta(y|X) = l_\theta(y|X) - \lambda \sum_{j=1}^{p} \|\theta_j\|_2^2 \tag{6}$$

   Here, λ is the regularization coefficient.

3. Next, let's calculate the gradient (detailed mathematical formulas are provided below).

   (a) The gradient of likelihood:

$$\frac{\partial l_\theta(y|X)}{\partial \theta_j} = \sum_{i=1}^{n}(y_i - P_\theta(y_i|X_i))x_{ij} \tag{7}$$

   (b) The gradient of L2 regularization:

$$\frac{\partial}{\partial \theta_j}(\lambda \theta_j^2) = 2\lambda\theta_j \tag{8}$$

   Therefore, the gradient of the regularized likelihood function is:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^{n}(y_i - P_\theta(y_i|X_i))x_{ij} - 2\lambda\theta_j \tag{9}$$

4. After computing the gradient, we need to update the value of θ. Since it's a gradient ascent algorithm, the weight update formula is:

$$\theta_j := \theta_j + \alpha \frac{\partial J(\theta)}{\partial \theta_j} \tag{10}$$

   α is Learning Rate

5. Repeat steps 3 and 4 until the likelihood function converges or a predetermined number of iterations is reached.

Gradients Let's delve deeper into how to compute the gradients for both the likelihood and the regularization term. First, consider that the prediction function for Logistic Regression is:

$$P_\theta(y_i|X_i) = \frac{1}{1+e^{-\theta^T X_i}} \tag{11}$$

Calculate the gradient of likelihood:

$$\frac{\partial l_\theta(y|X)}{\partial \theta_j} = \sum_{i=1}^{n} \left(y_i \frac{1}{P_\theta(y_i|X_i)} - (1-y_i) \frac{1}{1-P_\theta(y_i|X_i)} \right) \frac{\partial P_\theta(y_i|X_i)}{\partial \theta_j} \tag{12}$$

where $\frac{\partial P_\theta(y_i|X_i)}{\partial \theta_j}$ is the partial derivative of sigmoid, and it can be represented by:

$$\frac{\partial P_\theta(y_i|X_i)}{\partial \theta_j} = P_\theta(y_i|X_i)(1-P_\theta(y_i|X_i))x_{ij} \tag{13}$$

Hence, the gradient of likelihood is:

$$\begin{aligned} \frac{\partial l_\theta(y|X)}{\partial \theta_j} &= \sum_{i=1}^{n} \left(y_i \frac{1}{P_\theta(y_i|X_i)} - (1-y_i) \frac{1}{1-P_\theta(y_i|X_i)} \right) \frac{\partial P_\theta(y_i|X_i)}{\partial \theta_j} \\ &= \sum_{i=1}^{n} (y_i - P_\theta(y_i|X_i))x_{ij} \end{aligned} \tag{14}$$

Now we calculate the gradient of regularization:

$$\frac{\partial}{\partial \theta_j}(\lambda \theta_j^2) = 2\lambda\theta_j \tag{15}$$

substract them, and we get our final gradient expression 9

References