Solutions

Problem 1

1. For the samples (x_i, y_i), where x_i represents the population of the ith city and y_i represents the corresponding profit of a food truck in that city, write the closed-form solution for the linear regression problem that predicts profit based on population. Then, calculate this solution in Python using the data provided in `hw2data1.txt` and visualize the result.

We consider a simple linear model:

$$h_\theta(x) = \theta_0 + \theta_1 x.$$

Our dataset has m examples $\{(x^{(i)}, y^{(i)})\}_{i=1}^m$. We set up:

$$X = \begin{bmatrix} 1 & x^{(1)} \\ 1 & x^{(2)} \\ \vdots & \vdots \\ 1 & x^{(m)} \end{bmatrix}^\top, \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}, \quad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}^\top.$$

The cost function to minimize is the sum of squared errors:

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

The closed-form solution for θ is given by the *Normal Equation*:

$$\theta = \left(XX^\top\right)^{-1} Xy.$$

2. Implement Stochastic Gradient Descent (SGD) to optimize the linear regression parameters and visualize the result.

Our model and cost function remain:

$$h_\theta(x) = \theta_0 + \theta_1 x, \quad J(\theta) = \frac{1}{2m} \sum_{i=1}^m \Big(h_\theta(x^{(i)}) - y^{(i)}\Big)^2.$$

In *Stochastic Gradient Descent* (SGD), we process each training example (or a small batch) one at a time. The parameter update for a single example $(x^{(i)}, y^{(i)})$ is:

$$\theta_j := \theta_j - \alpha\big(h_\theta(x^{(i)}) - y^{(i)}\big)x_j^{(i)},$$

with $x_0^{(i)} = 1$.

3. Implement Stochastic Gradient Descent to optimize the corresponding regularized polynomial regression problem and visualize the result.

In polynomial regression, we introduce higher-order terms of the original feature x. For instance, if we want a cubic polynomial, our hypothesis looks like:

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4.$$

More generally, for a polynomial of degree d, we define:

$$X_{\text{poly}} = \begin{bmatrix} 1 & x^{(1)} & (x^{(1)})^2 & \cdots & (x^{(1)})^d \\ 1 & x^{(2)} & (x^{(2)})^2 & \cdots & (x^{(2)})^d \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x^{(m)} & (x^{(m)})^2 & \cdots & (x^{(m)})^d \end{bmatrix}.$$

To prevent overfitting in polynomial regression, we often apply regularization. As an example, we take an L_2 regularizer on all higher-order terms:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \Big(h_\theta(x^{(i)}) - y^{(i)} \Big)^2 + \frac{\lambda}{2m} \sum_{j=1}^{d} \theta_j^2.$$

Problem 2

1. The logit function is defined as:

$$\operatorname{logit}(p) = \ln\left(\frac{p}{1-p},\right)$$

where $p \in (0, 1)$. The sigmoid function (logistic function) is given by:

$$\sigma(x) = \frac{1}{1+e^{-x}},$$

where $x \in \mathbb{R}$.

Prove that the sigmoid function and the logit function are inverse functions of each other, i.e.,

$$\sigma(\operatorname{logit}(p)) = p, \text{ and } \operatorname{logit}(\sigma(x)) = x.$$

Proof. 1. Prove $\sigma(\operatorname{logit}(p)) = p$:

$$\begin{aligned}
\sigma(\operatorname{logit}(p)) &= \frac{1}{1+e^{-\operatorname{logit}(p)}} \tag{1}\\
&= \frac{1}{1+e^{-\ln\left(\frac{p}{1-p}\right)}} \tag{2}\\
&= \frac{1}{1+\frac{1-p}{p}} \tag{3}\\
&= \frac{1}{1+\frac{1-p}{p}} \tag{4}\\
&= \frac{1}{\frac{p+(1-p)}{p}} \tag{5}\\
&= p. \tag{6}
\end{aligned}$$

2. Prove $\operatorname{logit}(\sigma(x)) = x$:

$$\begin{aligned}
\operatorname{logit}(\sigma(x)) &= \ln\left(\frac{\sigma(x)}{1-\sigma(x)}\right) \tag{7}\\
&= \ln\left(\frac{\frac{1}{1+e^{-x}}}{1-\frac{1}{1+e^{-x}}}\right) \tag{8}\\
&= \ln\left(\frac{\frac{1}{1+e^{-x}}}{\frac{e^{-x}}{1+e^{-x}}}\right) \tag{9}\\
&= \ln\left(\frac{1}{e^{-x}}\right) \tag{10}\\
&= x \tag{11}
\end{aligned}$$

□

2. In **probit regression**, we assume a binary outcome $Y \in \{0, 1\}$ with the probability model:

$$P(Y = 1 \mid X) = \Phi(X\beta) \tag{12}$$

where $\Phi(\cdot)$ is the cumulative distribution function (CDF) of the standard normal distribution, $X \in \mathbb{R}^d$ is a vector of covariates, and β is the coefficients of the probit regression.

Now, given a dataset $\{(x_i, y_i)\}_{i=1}^n$, please: 1) write the log-likelihood function in MLE, 2) compute the gradient of the log-likelihood function.

Solution: Given a dataset $\{(x_i, y_i)\}_{i=1}^n$, the likelihood function is:

$$\begin{aligned} L(\beta) &= \prod_{i=1}^n P(Y_i = y_i \mid X_i) \\ &= \prod_{i=1}^n \Phi(X_i\beta)^{y_i}(1 - \Phi(X_i\beta))^{1-y_i} \end{aligned}$$

Then, we have

$$\ell(\beta) = \sum_{i=1}^n y_i \ln \Phi(X_i\beta) + (1 - y_i) \ln(1 - \Phi(X_i\beta)),$$

The derivative of the normal CDF is the standard normal PDF:

$$\frac{d}{dx}\Phi(x) = \phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}.$$

Thus, differentiating the log-likelihood function:

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^n \left[\frac{y_i}{\Phi(X_i\beta)} - \frac{1 - y_i}{1 - \Phi(X_i\beta)} \right] \phi(X_i\beta) X_i^\top.$$

Then, we can use numerical methods to solve the MLE.