Solutions Template

1 Written Questions [20 pts]

Write your solutions to the HW4 Written Questions in this template. You may either enter your solutions into the .tex file directly, or print out the pdf version and write your solutions by hand in the boxes provided. When you have completed all of the written questions, you should upload your solutions to Gradescope in pdf format.

1.1 Binary Logistic Regression [12 pts]

1. **[2 Points]** In logistic regression, our goal is to learn a set of parameters by maximizing the conditional log likelihood of the data. Assume you are given a dataset with N training examples and M features. Write down a formula for the *negative* conditional log likelihood of the training data in terms of the design matrix $\mathbf{X}$, the labels $\mathbf{y}$, and the parameter vector $\boldsymbol{\theta}$. This will be your objective function $J(\boldsymbol{\theta})$ for gradient descent. (Recall that ith row of the design matrix $\mathbf{X}$ contains the features $\mathbf{x}^{(i)}$ of the ith training example. The ith entry in the vector $\mathbf{y}$ is the label $y^{(i)}$ of the ith training example. Here we assume that each feature vector $\mathbf{x}^{(i)}$ contains a bias *feature*, e.g. $x_1^{(i)} = 1 \; \forall i \in \{1, \ldots, N\}$. As such, the bias parameter is folded into our parameter vector $\boldsymbol{\theta}$.)

$$J(\theta) = -\sum_{i=1}^{N} \log P_\theta(y^{(i)} \mid X^{(i)})$$

$$= -\sum_{i=1}^{N} \log\left[\sigma(\theta^T \vec{x}^T)^{y^{(i)}} \cdot (1-\sigma(\theta^T \vec{x}^T))^{1-y^{(i)}}\right]$$

where $\sigma(u) = \frac{1}{1+\exp(-u)}$

2. **[3 Points]** Derive the partial derivative of the objective function with respect to the mth parameter. That is, derive $\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_m}$, where $J(\boldsymbol{\theta})$ is the objective that you provided above. Please show all derivatives can be written in a finite sum form. Show all steps of the derivation.

$$\frac{\partial J(\theta)}{\partial \theta_m} = -\sum_{i=1}^{N} \frac{\partial}{\partial \theta_m}\left[\log \sigma(\theta^T \vec{x}^{(i)})^{y^{(i)}}\right] + \frac{\partial}{\partial \theta_m}\left[(1-\sigma(\theta^T \vec{x}^T))^{1-y^{(i)}}\right]$$

$$= -\sum_{i=1}^{N} \frac{\partial}{\partial \theta_m}\left[y^{(i)}(\log \sigma(\theta^T \vec{x}^T)\right] + \frac{\partial}{\partial \theta_m}\left[(1-y^{(i)}) \log(1-\sigma(\theta^T \vec{x}^T))\right]$$

$$= -\sum_{i=1}^{N}\left[\frac{y^{(i)}}{\sigma(\theta^T \vec{x}^T)} \cdot \frac{\partial}{\partial \theta_m}(\sigma(\theta^T \vec{x}^T)) + \frac{1-y^{(i)}}{(1-\sigma(\theta^T \vec{x}))} \cdot \frac{\partial}{\partial \theta_m}(1-\sigma(\theta^T \vec{x}^T))\right]$$

$$= -\sum_{i=1}^{N}\left[\frac{y^{(i)}}{\sigma} \cdot \sigma^2 \cdot \frac{1-\sigma}{\sigma} \cdot x_m^{(i)} - \frac{1-y^{(i)}}{1-\sigma} \cdot \sigma^2 \frac{1-\sigma}{\sigma} \cdot x_m^{(i)}\right]$$

$$= -\sum_{i=1}^{N}\left[(y^{(i)} - \sigma(\vec{\theta}^T \vec{x}^T)) \cdot x_m^{(i)}\right], \text{ where } \sigma(u) = \frac{1}{1+\exp(-u)}$$

3. **[2 Points]** Write gradient descent update rules for logistic regression for arbitrary θ_m.

$\theta_m = \theta_m + \eta \sum_{i=1}^{N} [(y^{(i)} - \sigma(\vec{\theta}^T \vec{x}^T)) \cdot x_m^{(i)}]$, where $\sigma(u) = \frac{1}{1+\exp(-u)}$

4. **[2 Points]** Write down the stochastic gradient descent update for an arbitrary θ_m using the i^{th} training example with features $\mathbf{x}^{(i)}$ and output label $y^{(i)}$.

$\theta_m = \theta_m + \eta (y^{(i)} - \sigma(\vec{\theta}^T \vec{x}^T)) x_m^{(i)}$ where $\sigma(u) = \frac{1}{1+\exp(-u)}$

5. **[3 Points]** If you train logistic regression for infinite iterations without ℓ_1 or ℓ_2 regularization, the weights can go to infinity. What is an intuitive explanation for this phenomenon? How does regularization help correct the problem?

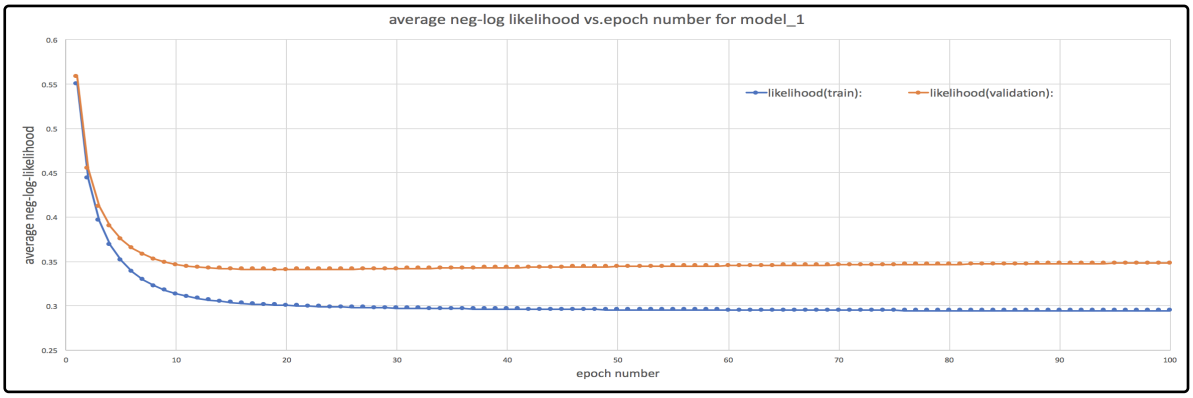
Overfitting

It trades off between fitting the data and keeping the model simple

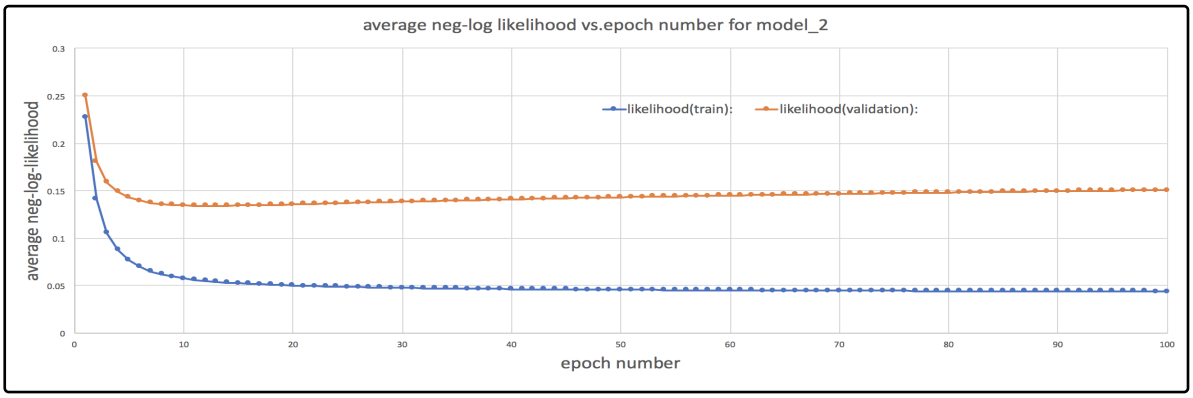
1.2 Empirical Questions [8 pts]

The following questions should be completed as you work through the programming portion of this assignment (Section 2).

1. **Plots [2 Points]** Using the data in the largedata folder in the handout, for each model, make a plot that shows the average negative log likelihood for the training and validation data sets after each of 100 epochs. The y-axis should show the negative log likelihood and the x-axis should show the number of epochs.



2. **Plots [2 Points]** For *Model 2*, make a plot as in the previous question.



3. **Explanation of Experiment [2 Points]** Write a few sentences explaining the output of the above experiment. In particular do the training and validation log likelihood curves look the same or different? Why?

For both model, the average look likelihood tends to be lower for training data than that for validation data, which indicates the trained logistic regression model is doing better on the trained data. This is due to that the model is trained on training data and so that it give better prediction results on the data that is used for training the model.

4. **Results [2 Points]** Make a table with your train and test error for the large data set (found in the largedata folder in the handout) for each of the 2 models after running for 10 epochs.

	Train Error	Test Error
Model 1	0.141035	0.146899
Model 2	0.018433	0.055495

Table 1.1: "Large Data" Results

5. **Collaboration Questions** After you have completed all other components of this assignment, report your answers to the collaboration policy questions detailed in the Academic Integrity Policies found here. (a) Did you receive any help whatsoever from anyone in solving this assignment? Is so, include full details. (b) Did you give any help whatsoever to anyone in solving this assignment? Is so, include full details? (c) Did you find or come across code that implements any part of this assignment ? If so, include full details.

No.