CS 229, Fall 2018

Problem Set #1: Supervised Learning

Due Wednesday, Oct 17 at 11:59 pm on Gradescope

Notes: (1) These questions require thought, but do not require long answers. Please be as concise as possible. (2) If you have a question about this homework, we encourage you to post your question on our Piazza forum, at http://piazza.com/stanford/fall2018/cs229. (3) If you missed the first lecture or are unfamiliar with the collaboration or honor code policy, please read the policy on Handout #1 (available from the course website) before starting work. (4) For the coding problems, you may not use any libraries except those defined in the provided `environment.yml` file. In particular, ML-specific libraries such as `scikit-learn` are not permitted. (5) To account for late days, the due date listed on Gradescope is Oct 20 at 11:59 pm. If you submit after Oct 17, you will begin consuming your late days. If you wish to submit on time, submit before Oct 17 at 11:59 pm.

All students must submit an electronic PDF version of the written questions. We highly recommend typesetting your solutions via LaTeX. If you are scanning your document by cell phone, please check the Piazza forum for recommended scanning apps and best practices. All students must also submit a zip file of their source code to Gradescope, which should be created using the `make_zip.py` script. In order to pass the auto-grader tests, you should make sure to (1) restrict yourself to only using libraries included in the `environment.yml` file, and (2) make sure your code runs without errors using the `run.py` script. Your submission will be evaluated by the auto-grader using a private test set.

Honor code: We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down the solutions independently, and without referring to written notes from the joint session. In other words, each student must understand the solution well enough in order to reconstruct it by him/herself. In addition, each student should write on the problem set the set of people with whom s/he collaborated. Further, because we occasionally reuse problem set questions from previous years, we expect students not to copy, refer to, or look at the solutions in preparing their answers. It is an honor code violation to intentionally refer to a previous year's solutions.

1 [40 points] Linear Classifiers (logistic regression and GDA)

In this problem, we cover two probabilistic linear classifiers we have covered in class so far. First, a discriminative linear classifier: logistic regression. Second, a generative linear classifier: Gaussian discriminant analysis (GDA). Both the algorithms find a linear decision boundary that separates the data into two classes, but make different assumptions. Our goal in this problem is to get a deeper understanding of the similarities and differences (and, strengths and weaknesses) of these two algorithms.

For this problem, we will consider two datasets, provided in the following files:

i) `data/ds1_{train,valid}.csv`

ii) `data/ds2_{train,valid}.csv`

Each file contains m examples, one example $(x^{(i)}, y^{(i)})$ per row. In particular, the i-th row contains columns $x_0^{(i)} \in \mathbb{R}$, $x_1^{(i)} \in \mathbb{R}$, and $y^{(i)} \in \{0, 1\}$. In the subproblems that follow, we will investigate using logistic regression and Gaussian discriminant analysis (GDA) to perform binary classification on these two datasets.

(a) [10 points] In lecture we saw the average empirical loss for logistic regression:

$$\mathbf{J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \Big(y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \Big),}$$

where $y^{(i)} \in \{0, 1\}$, $h_\theta(x) = g(\theta^T x)$ and $g(z) = \frac{1}{1+e^{-z}}$. Find the Hessian H of this function, and show that for any vector z, it holds true that $z^T H z \geq 0$.
Hint: **You may want to start by showing that $\sum_i \sum_j z_i x_i x_j z_j = (x^T z)^2 \geq 0$. Recall also that $g'(z) = g(z)(1 - g(z))$.**
Remark: **This implies that $H \succeq 0$ and hence J is convex.**

The Hessian is defined as

$$H_J(\theta) = \begin{pmatrix} \frac{\partial^2 f}{\partial \theta_1^2}(x) & \frac{\partial^2 f}{\partial \theta_1 \partial \theta_2}(\theta) & \dots & \frac{\partial^2 f}{\partial \theta_1 \partial \theta_n}(\theta) \\ \frac{\partial^2 f}{\partial \theta_2 \partial \theta_1}(\theta) & \frac{\partial^2 f}{\partial \theta_2^2}(\theta) & \dots & \frac{\partial^2 f}{\partial \theta_2 \partial \theta_n}(\theta) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial \theta_n \partial \theta_1}(\theta) & \frac{\partial^2 f}{\partial \theta_n \partial \theta_2}(\theta) & \dots & \frac{\partial^2 f}{\partial \theta_n^2}(\theta) \end{pmatrix}$$

First of all, we start by computing the gradient

$$\begin{aligned} \frac{\partial J}{\partial \theta_k} = -\frac{1}{m} \Bigg[\sum_i y^{(i)} \frac{1}{g(\theta^T x^{(i)})} g(\theta^T x^{(i)}) \Big(1 - g(\theta^T x^{(i)})\Big) x_k^{(i)} \\ - (1 - y^{(i)}) \frac{1}{1 - g(\theta^T x^{(i)})} g(\theta^T x^{(i)}) \Big(1 - g(\theta^T x^{(i)})\Big) x_k^{(i)} \Bigg] = \end{aligned}$$

$$= -\frac{1}{m} \Bigg[\sum_i y^{(i)} \Big(1 - g(\theta^T x^{(i)})\Big) x_k^{(i)} - (1 - y^{(i)}) g(\theta^T x^{(i)}) x_k^{(i)} \Bigg] =$$

$$= -\frac{1}{m} \Bigg[\sum_i \Big(y^{(i)} - g(\theta^T x^{(i)})\Big) x_k^{(i)} \Bigg]$$

Therefore

$$\nabla_\theta J(\theta) = -\frac{1}{m} X^T (Y - g(X\theta))$$

If we now compute the second derivative of $J(\theta)$

$$\frac{\partial^2 J}{\partial \theta_j \partial \theta_k} = -\frac{1}{m} \sum_i \Bigg[- g(\theta^T x^{(i)}) \Big(1 - g(\theta^T x^{(i)})\Big) x_k^{(i)} x_j^{(i)} \Bigg] = \frac{1}{m} \sum_i g(\theta^T x^{(i)}) \Big(1 - g(\theta^T x^{(i)})\Big) x_j^{(i)} x_k^{(i)}$$

As we see, we have something similar in form to $\sum_i a_i x_{ij} x_{ik}$. Because we have two free indexes, this must be a matrix. The term a_i has only the i index and therefore does not mix the j and k, so it can be derived only from a diagonal term.
Thus

$$H_J(\theta) = \frac{1}{m} X^T \text{Diag}\Big(g(X\theta) \odot (1 - g(X\theta))\Big) X$$

We can make a small comprobation by checking the dimensions. The Hessian is a $\mathbb{R}^{n \times n}$ matrix and we have $(n \times m) \times Diag(m \times 1) \times (m \times n) = (n \times m) \times (m \times m) \times (m \times n) = n \times n$.

Now we want to show that $z^T H z \geq 0$. If we write the full expression

$$zHz = \frac{1}{m}\left[\sum_i \sum_j \sum_k g(\theta^T x^{(i)})(1 - g(\theta^T x^{(i)})) x_j^{(i)} x_k^{(i)} z_j z_k\right]$$

Using the fact that $(x^T z) = \sum_i x_i z_i$ and $(x^T z)^2 = \sum_i x_i z_i \sum_j x_j z_j = \sum_i \sum_j x_i x_j z_i z_j$, we can rewrite

$$zHz = \frac{1}{m}\left[\sum_i \sum_j \sum_k g(\theta^T x^{(i)})(1 - g(\theta^T x^{(i)})) x_j^{(i)} x_k^{(i)} z_i z_k\right] =$$
$$= \frac{1}{m}\left[\sum_i g(\theta^T x^{(i)})(1 - g(\theta^T x^{(i)}))[x^T z]^2\right]$$

And because $g(\theta^T x) \in [0, 1]$ and $[x^T z]^2 \geq 0$ we have that $zHz \geq 0$

(b) [5 points] Coding problem. Follow the instructions in `src/p01b_logreg.py` to train a logistic regression classifier using Newton's Method. Starting with $\theta = \vec{0}$, run Newton's Method until the updates to θ are small: Specifically, train until the first iteration k such that $\|\theta_k - \theta_{k-1}\|_1 < \varepsilon$, where $\varepsilon = 10^{-5}$. Make sure to write your model's predictions to the file specified in the code.

(c) [5 points] Recall that in GDA we model the joint distribution of (x, y) by

$$p(y) = \begin{cases} \phi & \text{if } y = 1 \\ 1 - \phi & \text{if } y = 0 \end{cases}$$

$$p(x \mid y = 0) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_0)^T \Sigma^{-1} (x - \mu_0)\right),$$
$$p(x \mid y = 1) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_1)^T \Sigma^{-1} (x - \mu_1)\right).$$

Suppose we have already fit ϕ, μ_0, μ_1, and Σ, and now want to predict y given a new point x. Show that

$$\mathbf{p(y = 1 \mid x; \phi, \mu_0, \mu_1, \Sigma) = \frac{1}{1 + \exp(-(\theta^T x + \theta_0))}},$$

for appropriate $\theta \in \mathbb{R}^n$ and $\theta_0 \in \mathbb{R}$ expressed in terms of $\phi, \Sigma, \mu_0, \mu_1$.

According to Bayes' theorem

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

where $p(x) = p(x|y = 1)p(y = 1) + p(x|y = 0)p(y = 0)$.

We start to compute the required expression

$$\begin{aligned}
p(y = 1 \mid x) &= \frac{p(x \mid y = 1)p(y = 1)}{p(x)} \\
&= \frac{p(x \mid y = 1)p(y = 1)}{p(x \mid y = 1)p(y = 1) + p(x \mid y = 0)p(y = 0)} \\
&= \frac{1}{1 + \frac{p(x \mid y = 0)p(y = 0)}{p(x \mid y = 1)p(y = 1)}} \\
&= \frac{1}{1 + \frac{1-\phi}{\phi} \frac{\exp(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0))}{\exp(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1))}} \\
&= \frac{1}{1 + \frac{1-\phi}{\phi} \exp\Big(\frac{1}{2}\big[(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) - (x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\big]\Big)} \\
&= \frac{1}{1 + \frac{1-\phi}{\phi} \exp\Bigg(\frac{1}{2}\Big[(x^T\Sigma^{-1} - \mu_1^T\Sigma^{-1})(x-\mu_1) \\
&\qquad\qquad - (x^T\Sigma^{-1} - \mu_0^T\Sigma^{-1})(x-\mu_0)\Big]\Bigg)} \\
&= \frac{1}{1 + \frac{1-\phi}{\phi} \exp\Bigg(\frac{1}{2}\Big[x^T\Sigma^{-1}x - \mu_1^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu_1 + \mu_1^T\Sigma^{-1}\mu_1 \\
&\qquad\qquad - x^T\Sigma^{-1}x + \mu_0^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu_0 - \mu_0^T\Sigma^{-1}\mu_0\Big]\Bigg)} \\
&= \frac{1}{1 + \exp\Big(\log\frac{1-\phi}{\phi}\Big) \exp\Bigg(\frac{1}{2}\Big[x^T\Sigma^{-1}x - \mu_1^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu_1 + \mu_1^T\Sigma^{-1}\mu_1 \\
&\qquad\qquad - x^T\Sigma^{-1}x + \mu_0^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu_0 - \mu_0^T\Sigma^{-1}\mu_0\Big]\Bigg)}
\end{aligned}$$

Because the product of a row vector, a matrix and a column is a scalar, then $(a^T\Sigma^{-1}b)^T = (b^T\Sigma^{-1}a)$; and the expression reduces to

$$\begin{aligned}
p(y=1 \mid x) &= \frac{1}{1+\exp\Big(\log\frac{1-\phi}{\phi}\Big)\,\exp\Bigg(\frac{1}{2}\Big[x^T\Sigma^{-1}x - \mu_1^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu_1 + \mu_1^T\Sigma^{-1}\mu_1 \\
&\qquad\qquad - x^T\Sigma^{-1}x + \mu_0^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu_0 - \mu_0^T\Sigma^{-1}\mu_0\Big]\Bigg)} \\
&= \frac{1}{1+\exp\Big(\log\frac{1-\phi}{\phi}\Big)\,\exp\Big(\frac{1}{2}\big[-2\mu_1^T\Sigma^{-1}x + \mu_1^T\Sigma^{-1}\mu_1 + 2\mu_0^T\Sigma^{-1}x - \mu_0^T\Sigma^{-1}\mu_0\big]\Big)} \\
&= \frac{1}{1+\exp\Big(\log\frac{1-\phi}{\phi}\Big)\,\exp\Big(\frac{1}{2}\big[(-2\mu_1^T\Sigma^{-1} + 2\mu_0^T\Sigma^{-1})x + \mu_1^T\Sigma^{-1}\mu_1 - \mu_0^T\Sigma^{-1}\mu_0\big]\Big)} \\
&= \frac{1}{1+\exp\Big(-(\mu_1^T-\mu_0^T)\Sigma^{-1}x + \frac{1}{2}\big[\mu_1^T\Sigma^{-1}\mu_1 - \mu_0^T\Sigma^{-1}\mu_0\big] + \log\frac{1-\phi}{\phi}\Big)} \\
&= \frac{1}{1+\exp\Bigg[-\Big((\mu_1-\mu_0)^T\Sigma^{-1}x - \frac{1}{2}\big[\mu_1^T\Sigma^{-1}\mu_1 - \mu_0^T\Sigma^{-1}\mu_0\big] - \log\frac{1-\phi}{\phi}\Big)\Bigg]}.
\end{aligned}$$

Thus

$$\boxed{\theta^T = \Sigma^{-1}(\mu_1-\mu_0), \quad \theta_0 = -\tfrac{1}{2}\big[\mu_1^T\Sigma^{-1}\mu_1 - \mu_0^T\Sigma^{-1}\mu_0\big] - \log\tfrac{1-\phi}{\phi}}$$

(d) [7 points] For this part only, assume $\mathbf{n=1}$, so $\boldsymbol{\Sigma} = [\boldsymbol{\sigma^2}]$ and $|\boldsymbol{\Sigma}| = \boldsymbol{\sigma^2}$. Given the dataset, we claim the MLEs are

$$\boldsymbol{\phi} = \frac{1}{\mathbf{m}}\sum_{i=1}^{\mathbf{m}} \mathbf{1}\{\mathbf{y}^{(i)}=\mathbf{1}\}, \quad \boldsymbol{\mu_0} = \frac{\sum_{i=1}^{\mathbf{m}}\mathbf{1}\{\mathbf{y}^{(i)}=\mathbf{0}\}\mathbf{x}^{(i)}}{\sum_{i=1}^{\mathbf{m}}\mathbf{1}\{\mathbf{y}^{(i)}=\mathbf{0}\}}, \quad \boldsymbol{\mu_1} = \frac{\sum_{i=1}^{\mathbf{m}}\mathbf{1}\{\mathbf{y}^{(i)}=\mathbf{1}\}\mathbf{x}^{(i)}}{\sum_{i=1}^{\mathbf{m}}\mathbf{1}\{\mathbf{y}^{(i)}=\mathbf{1}\}},$$

$$\boldsymbol{\Sigma} = \frac{\mathbf{1}}{\mathbf{m}}\sum_{i=\mathbf{1}}^{\mathbf{m}}\big(\mathbf{x}^{(i)} - \boldsymbol{\mu}_{\mathbf{y}^{(i)}}\big)\big(\mathbf{x}^{(i)} - \boldsymbol{\mu}_{\mathbf{y}^{(i)}}\big)^{\mathbf{T}}.$$

Starting from the log-likelihood

$$\boldsymbol{\ell}(\boldsymbol{\phi}, \boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \boldsymbol{\Sigma}) = \sum_{i=\mathbf{1}}^{\mathbf{m}} \log \mathbf{p}(\mathbf{x}^{(i)} \mid \mathbf{y}^{(i)}; \boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \boldsymbol{\Sigma}) + \sum_{i=\mathbf{1}}^{\mathbf{m}} \log \mathbf{p}(\mathbf{y}^{(i)}; \boldsymbol{\phi}),$$

prove that the MLEs above indeed maximize $\boldsymbol{\ell}$ (assume at least one positive and one negative example).

If we take into account the expressions (for $n = 1$) of $p(x^{(i)}|y^{(i)}; \mu_0, \mu_1, \Sigma) =$
$= \frac{1}{(\sqrt{2\pi}\sigma)} \exp\left(-\frac{1}{2\sigma^2}(x^{(i)} - \mu_{(y^{(i)})})^2\right)$ and $p(y^{(i)}; \phi) = \phi^{y^{(i)}}(1-\phi)^{1-y^{(i)}}$, we obtain

$$\ell(\phi, \mu_0, \mu_1, \Sigma) = \sum_i^m -\log(2\pi) - \log(\sigma) - \frac{1}{2\sigma^2}(x^{(i)} - \mu_{(y^{(i)})})^2 + y^{(i)}\log(\phi) + (1-y^{(i)})\log(1-\phi)$$

Now we find the maximum with respect each of the parameters. To do so, we derive with respect the parameter we want to maximize and equal the expression to zero. We start with ϕ

$$\partial_\phi \left[\sum_i y^{(i)}\log(\phi) + (1-y^{(i)})\log(\phi)\right] = \frac{\sum_i y^{(i)}}{\phi} - \frac{\sum_i (1-y^{(i)})}{1-\phi} = 0$$

$$(1-\phi)\sum_i y^{(i)} = \phi \sum_i (1-y^{(i)})$$

$$\sum_i y^{(i)} - \cancel{\phi \sum_i y^{(i)}} = m\phi - \cancel{\phi \sum_i y^{(i)}}$$

$$\boxed{\phi = \frac{\sum_i^m y^{(i)}}{m}}$$

In the case of $\mu_{(y^{(i)})}$

$$\partial_{\mu_{(y^{(i)})}} \left[\sum_i^m -\frac{1}{2\sigma^2}\left(x^{(i)} - \mu_{(y^{(i)})}\right)^2\right] = \frac{1}{\sigma^2}\sum_i^m \left(x^{(i)} - \mu_{(y^{(i)})}\right)\partial_{\mu_{(y^{(i)})}}\mu_{(y^{(i)})} = 0.$$

Now, we can distinguish two cases, depending of the two possible values of $\mu_{(y^{(i)})}$. If $\mu_{(y^{(i)})} = \mu_0$

$$\sum_i^m \left(x^{(i)} - \mu_{(y^{(i)})}\right)\partial_{\mu_{(y^{(i)})}}\mu_{(y^{(i)})} = \sum_i^m x^{(i)}\partial_{\mu_{(y^{(i)})}}\mu_{(y^{(i)})} - \sum_i^m \mu_{(y^{(i)})}\partial_{\mu_{(y^{(i)})}}\mu_{(y^{(i)})} = 0.$$

The expression $\partial_{\mu_{(y^{(i)})}}\mu_{(y^{(i)})}$ is only one when $\mathbb{1}\{y^{(i)} = 0\}$ (and 0 otherwise) where $\mathbb{1}$ is the indicator function. Therefore

$$\sum_i^m x^{(i)}\mathbb{1}\{y^{(i)} = 0\} - \mu_0 \sum_i^m \mathbb{1}\{y^{(i)} = 0\} = 0$$

$$\sum_i^m x^{(i)}\mathbb{1}\{y^{(i)} = 0\} = \mu_0 \sum_i^m \mathbb{1}\{y^{(i)} = 0\}$$

$$\sum_i^m x^{(i)}\mathbb{1}\{y^{(i)} = 0\} = \mu_0 \sum_i^m \mathbb{1}\{y^{(i)} = 0\}$$

$$\boxed{\mu_0 = \frac{\sum_i^m x^{(i)}\mathbb{1}\{y^{(i)} = 0\}}{\sum_i^m \mathbb{1}\{y^{(i)} = 0\}}}.$$

And similar with μ_1

$$\boxed{\mu_1 = \frac{\sum_i^m x^{(i)}\mathbb{1}\{y^{(i)} = 1\}}{\sum_i^m \mathbb{1}\{y^{(i)} = 1\}}}.$$

Finally, we just have left the optimal value of the σ parameter,

$$\partial_\sigma \left[-\log(\sigma)m - \frac{1}{2\sigma^2} \sum_i (x^{(i)} - \mu_{(y^{(i)})})^2 \right] = -\frac{m}{\sigma} + \frac{1}{\sigma^3} \sum_i (x^{(i)} - \mu_{(y^{(i)})})^2 = 0.$$

If we assume that $\sigma \neq 0$,

$$-m\sigma^2 + \sum_i (x^{(i)} - \mu_{(y^{(i)})})^2 = 0$$

$$m\sigma^2 = \sum_i (x^{(i)} - \mu_{(y^{(i)})})^2$$

$$\sigma^2 = \frac{1}{m} \sum_i (x^{(i)} - \mu_{(y^{(i)})})^2$$

$$\boxed{\sigma^2 = \Sigma = \frac{1}{m} \sum_i \left(x^{(i)} - \mu_{(y^{(i)})} \right)^2}$$

(e) [3 points] **Coding problem.** In `src/p01e_gda.py`, compute $\phi, \mu_0, \mu_1, \Sigma$, use these to derive θ, and use the resulting GDA model to make predictions on the validation set.

(f) [5 points] For Dataset 1, plot the training data with x_1 on the horizontal axis and x_2 on the vertical axis. Use different symbols for the two classes. On the same figure, plot the decision boundary found by logistic regression in part (b). Make an identical plot with the decision boundary found by GDA in part (e).

(g) [5 points] Repeat part (f) for Dataset 2. On which dataset does GDA seem to perform worse than logistic regression? Why might this be the case?

tra credit points] For the dataset where GDA performed worse, can you find a transformation of the $x^{(i)}$'s such that GDA performs significantly better? What is this transformation?

2 [30 points] Incomplete, Positive-Only Labels

We consider training binary classifiers when we have labels only for a subset of the positive examples. We assume a dataset $\{(\boldsymbol{x}^{(i)}, \boldsymbol{t}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^m$, where $\boldsymbol{t}^{(i)} \in \{\mathbf{0}, \mathbf{1}\}$ is the "true" label and

$$\boldsymbol{y}^{(i)} = \begin{cases} \mathbf{1} & \boldsymbol{x}^{(i)} \textbf{ is labeled} \\ \mathbf{0} & \textbf{otherwise.} \end{cases}$$

All labeled examples are positive, i.e., $p(t^{(i)} = 1 \mid y^{(i)} = 1) = 1$, but unlabeled examples may be positive or negative. Our goal is to construct h such that $h(x^{(i)}) \approx p(t^{(i)} = 1 \mid x^{(i)})$ using only x and y.

Real world example: Proteins database for cross-membrane signaling, etc.

(a) [5 points] Suppose $y^{(i)}$ and $x^{(i)}$ are conditionally independent given $t^{(i)}$:

$$p(y^{(i)} = 1 \mid t^{(i)} = 1, x^{(i)}) = p(y^{(i)} = 1 \mid t^{(i)} = 1).$$

Prove that $p(t^{(i)} = 1 \mid x^{(i)}) = p(y^{(i)} = 1 \mid x^{(i)})/\alpha$ for some constant α.

The law of total probability states that,

If $\{B_n : n = 1, 2, 3...\}$ is a finite or countably infinite set of mutually exclusive and collectively exhaustive events, then for any event A

$$P(A) = \sum_n P(A|B_n)P(B_n).$$

Because the set $\{t^{(i)} = 1, t^{(i)} \neq 1\}$ is a finite set of mutually exclusive and collectively exhaustive events, we have

$$P(y^{(i)} = 1) = P(y^{(i)} = 1|t^{(i)} = 1)P(t^{(i)} = 1) + P(y^{(i)} = 1|t^{(i)} \neq 1)P(t^{(i)} \neq 1)$$

If we now condition to the event $x^{(i)}$,

$$P(y^{(i)} = 1|x^{(i)}) = P(y^{(i)} = 1|t^{(i)} = 1, x^{(i)})P(t^{(i)} = 1|x^{(i)}) + P(y^{(i)} = 1|t^{(i)} \neq 1, x^{(i)})P(t^{(i)} \neq 1|x^{(i)}).$$

We know that $P(t^{(i)} = 1|y^{(i)} = 1) = 1$, which implies that $y^{(i)} = 1 \longrightarrow t^{(i)} = 1$. If we negate the conclusion, then $t^{(i)} \neq 1 \longrightarrow y^{(i)} \neq 1$. Therefore,

$$P(y^{(i)} = 1|t^{(i)} \neq 1, x^{(i)}) = 0.$$

This means that, given the event $t^{(i)} \neq 1$, independently of $x^{(i)}$, the event $y^{(i)} = 1$ cannot occur.
Thus

$$P(y^{(i)} = 1|x^{(i)}) = P(y^{(i)} = 1|t^{(i)} = 1, x^{(i)})P(t^{(i)} = 1|x^{(i)}) + \cancel{P(y^{(i)} = 1|t^{(i)} \neq 1, x^{(i)})P(t^{(i)} \neq 1|x^{(i)})}$$

And finally we get

$$\boxed{P\left(t^{(i)} = 1|x^{(i)}\right) = \frac{P\left(y^{(i)} = 1|x^{(i)}\right)}{P\left(y^{(i)} = 1|t^{(i)} = 1, x^{(i)}\right)}; \quad \alpha := P\left(y^{(i)} = 1|t^{(i)} = 1, x^{(i)}\right)}$$

(b) [5 points] Using a trained classifier h and a held-out validation set V, let $V^+ = \{x^{(i)} \in V \mid y^{(i)} = 1\}$. Assuming $h(x^{(i)}) \approx p(y^{(i)} = 1 \mid x^{(i)})$ and that $p(t^{(i)} = 1 \mid x^{(i)}) \approx 1$ for $x^{(i)} \in V^+$, show that $h(x^{(i)}) \approx \alpha$ for all $x^{(i)} \in V^+$.

From the result we obtained in (a),

$$1 \approx \frac{P(y^{(i)} = 1|x^{(i)})}{\alpha},$$

$$\alpha \approx P(y^{(i)} = 1|x^{(i)}).$$

Thus

$$\boxed{h\left(x^{(i)}\right) \approx \alpha = P\left(y^{(i)} = 1|t^{(i)} = 1, x^{(i)}\right)}$$

(c) [5 points] Coding problem. Consider files `data/ds3_{train,valid,test}.csv` with columns `x1, x2, y, t`. In `src/p02cde_posonly.py`, write logistic regression using `x1` and `x2` as input features, and train using the t-labels. Output predictions on the test set.

(d) [5 points] Coding problem. Re-train the classifier using only the y-labels (still using `x1` and `x2`).

(e) [10 points] Coding problem. Estimate α on the validation set by averaging your classifier's predictions over V^+:

$$\alpha \approx \frac{1}{|V^+|} \sum_{x^{(i)} \in V^+} h(x^{(i)}).$$

Rescale your predictions from part (d) using this α. Using a threshold of 0.5 for $p(t^{(i)} = 1 \mid x^{(i)})$, make three plots with the decision boundaries from parts (c)–(e) on top of the test set. Plot `x1` horizontally and `x2` vertically; use different symbols for positive ($t^{(i)} = 1$) and negative ($t^{(i)} = 0$). In each plot, indicate the separating hyperplane with a red line.

Remark: **Since $p(t \mid x)$ differs from $p(y \mid x)$ by a constant factor, ranking by either is equivalent.**

3 [25 points] Poisson Regression

(a) [5 points] Consider the Poisson distribution parameterized by λ:

$$\mathrm{p(y;\lambda)} = \frac{e^{-\lambda}\,\lambda^{\mathrm{y}}}{\mathrm{y!}}.$$

Show that the Poisson distribution is in the exponential family, and clearly state $b(y)$, η, $T(y)$, and $a(\eta)$.

The general form of the exponential family is,

$$p(y;\eta) = b(y)\exp\Big(\eta^T T(y) - a(\eta)\Big).$$

We see that the Poisson distribution has almost this form, we just need to work on the term λ^y

$$p(y;\lambda) = \frac{e^{-\lambda}\lambda^y}{y!} = \frac{e^{-\lambda}e^{y\log\lambda}}{y!} = \frac{e^{y\log\lambda-\lambda}}{y!}.$$

Therefore

$$\boxed{b(y) = \frac{1}{y!}, \quad T(y) = y, \quad \eta = \log(\lambda), \quad a(\eta) = \lambda = e^{\eta}}$$

(b) [3 points] Consider performing regression using a GLM model with a Poisson response variable. What is the canonical response function for the family? (You may use the fact that a Poisson random variable with parameter λ has mean λ.)

For a GLM, given a new sample x, the output of the model is $E[y|x;\theta] \implies h_\theta(x) = E[y|x;\theta]$. Also, the exponential family has the following property,

$$\boxed{E[y;\eta] = \lambda = \partial_\eta a(\eta) = e^{\eta} = e^{\theta^T x}}$$

(c) [7 points] For training set $\{(x^{(i)}, y^{(i)})\}$, let the log-likelihood be $\log p(y^{(i)} \mid x^{(i)};\theta)$. By differentiating w.r.t. θ_j, derive the stochastic gradient ascent update rule for a GLM with Poisson responses and the canonical link.

We basically need to derive the log-likelihood expression with respect to the parameter θ,

$$\begin{aligned}
\log p(y^{(i)} \mid x^{(i)};\theta) &= \log\left(\frac{e^{y^{(i)}\log\lambda-\lambda}}{y^{(i)}!}\right) \\
&= \log\left(\frac{e^{y^{(i)}\log e^{\theta^T x^{(i)}} - e^{\theta^T x^{(i)}}}}{y^{(i)}!}\right) \\
&= \log\left(\frac{1}{y^{(i)}!}\right) + \log\left(e^{y^{(i)}\log e^{\theta^T x^{(i)}} - e^{\theta^T x^{(i)}}}\right) \\
&= -\log\left(y^{(i)}!\right) + y^{(i)}\theta^T x^{(i)} - e^{\theta^T x^{(i)}}.
\end{aligned}$$

If we now derive with respect to θ,

$$\begin{aligned}\frac{\partial}{\partial\theta_j}\log p(y^{(i)} \mid x^{(i)};\theta) &= y^{(i)}\frac{\partial}{\partial\theta_j}\theta^T x^{(i)} - \frac{\partial}{\partial\theta_j}e^{\theta^T x^{(i)}} \\ &= y^{(i)}x_j^{(i)} - e^{\theta^T x^{(i)}}x_j^{(i)} \\ &= \big(y^{(i)} - e^{\theta^T x^{(i)}}\big)x_j^{(i)} \\ &= \big(y^{(i)} - h_\theta(x^{(i)})\big)x_j^{(i)}.\end{aligned}$$

Thus, the update rule is,

$$\boxed{\theta_{j+1} = \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}}$$

(d) [7 points] Coding problem. Dataset: `data/ds4_{train,valid}.csv`. Implement Poisson regression in `src/p03d_poisson.py` and use gradient ascent to maximize the log-likelihood of θ (with $\eta = \theta^T x$).

4 [15 points] Convexity of Generalized Linear Models

We aim to show that the negative log-likelihood (NLL) loss of a GLM is convex in θ. Assume a scalar natural parameter η and $T(y) = y$, so

$$p(y;\eta) = b(y)\exp(\eta y - a(\eta)).$$

(a) [5 points] Derive an expression for the mean of the distribution. Show that $\mathbb{E}[Y \mid X;\theta]$ can be represented as the gradient of the log-partition a with respect to the natural parameter η. Hint: Start with observing that $\partial_\eta \int p(y;\eta)dy = \int \partial_\eta p(y;\eta)dy$.

If we start from the property $1 = \int_{\mathbb{R}} p(y;\eta)$ and derive the expression with respect the natural parameter,

$$\begin{aligned}\partial_\eta 1 &= \partial_\eta \int_{-\infty}^{\infty} b(y)e^{\eta y - a(\eta)}dy, \\ 0 &= \int_{-\infty}^{\infty} \partial_\eta b(y)e^{\eta y - a(\eta)}dy, \\ 0 &= \int_{-\infty}^{\infty} b(y)\partial_\eta\left(e^{\eta y - a(\eta)}\right)dy, \\ 0 &= \int_{-\infty}^{\infty} b(y)(y - \partial_\eta a(\eta))\left(e^{\eta y - a(\eta)}\right)dy, \\ 0 &= \int_{-\infty}^{\infty} b(y)ye^{\eta y - a(\eta)}dy - \partial_\eta a(\eta)\int_{-\infty}^{\infty} b(y)e^{\eta y - a(\eta)}dy, \\ 0 &= E[Y;\eta] - \partial_\eta a(\eta).\end{aligned}$$

Therefore

$$\boxed{E[Y;\eta] = \partial_\eta a(\eta)}$$

(b) [5 points] Next, derive an expression for the variance of the distribution. In particular, show that $Var(Y|X;\theta)$ can be expressed as the derivative of the mean w.r.t η (i.e., the second derivative of the log-partition function $a(\eta)$ w.r.t the natural parameter η.)

If we derive with respect to the natural parameter the expected value,

$$\begin{aligned}
\partial_\eta \mathbb{E}[Y;\eta] &= \partial_\eta \int_{-\infty}^{\infty} y\, b(y)\, e^{\eta y - a(\eta)}\, dy \\
&= \int_{-\infty}^{\infty} y\, b(y)\, \partial_\eta \Big(e^{\eta y - a(\eta)} \Big)\, dy \\
&= \int_{-\infty}^{\infty} y\, b(y)\, (y - \partial_\eta a(\eta))\, e^{\eta y - a(\eta)}\, dy \\
&= \int_{-\infty}^{\infty} y^2\, b(y)\, e^{\eta y - a(\eta)}\, dy - \partial_\eta a(\eta) \int_{-\infty}^{\infty} y\, b(y)\, e^{\eta y - a(\eta)}\, dy \\
&= \mathbb{E}[Y^2;\eta] - \mathbb{E}^2[Y;\eta] \\
&= \mathrm{Var}(Y;\eta).
\end{aligned}$$

Thus,

$$\boxed{\mathrm{Var}(Y;\eta) = \partial_\eta \mathbb{E}[Y;\eta] = \frac{\partial^2}{\partial \eta^2} a(\eta)}$$

(c) [5 points] Finally, write out the loss function $l(\theta)$, the NLL of the distribution, as a function of θ. Then, calculate the Hessian of the loss w.r.t θ, and show that it is always PSD. This concludes the proof that NLL loss of GLM is convex.

Hint: Use the chain rule of calculus along with the results of the previous parts to simplify your derivations.

First, we need to derive the expression of the loss function

$$\begin{aligned} l(\theta) &= -\log\left(\prod_{i=1}^{m} p(y^{(i)} \mid x^{(i)}; \theta)\right) \\ &= -\sum_{i=1}^{m} \log\, p(y^{(i)} \mid x^{(i)}; \theta) \\ &= -\sum_{i=1}^{m} \left[\log b(y^{(i)}) + \theta^T x^{(i)} y^{(i)} - a(\theta^T x^{(i)})\right]. \end{aligned}$$

Now, we need to derive the loss function with respect the parameter θ

$$\frac{\partial l(\theta)}{\partial \theta_j} = -\sum_{i=1}^{m} x_j^{(i)} y^{(i)} - a(\theta^T x^{(i)}) x_j^{(i)} = \sum_{i=1}^{m} \left[a'(\theta^T x^{(i)}) - y^{(i)}\right] x_j^{(i)}.$$

And finally, the Hessian is

$$H_{jk} = \frac{\partial l^2(\theta)}{\partial \theta_j \partial \theta_k} = \frac{\partial}{\partial \theta_k} \sum_{i=1}^{m} \left[a'(\theta^T x^{(i)}) - y^{(i)}\right] x_j^{(i)} = \sum_{i=1}^{m} a''(\theta^T x^{(i)}) x_j^{(i)} x_k^{(i)}.$$

We just need to show that the Hessian is PSD,

$$z^T H z = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{n} a''(\theta^T x^{(i)}) x_j^{(i)} x_k^{(i)} z_j z_k = \sum_{i=1}^{m} a''(\theta^T x^{(i)}) [(x^{(i)})^T z]^2.$$

Because $a''(\theta^T x^{(i)}) = \mathrm{Var}(Y; \eta)$ and the variance is always positive then,

$$\boxed{z^T H z \geq 0}.$$

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

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

i. **Show that $J(\theta) = (X\theta - y)^T W (X\theta - y)$ for an appropriate matrix W. Specify each element of W.**

From the expression given we see that there are m $w^{(i)}$'s, and $w^{(i)}$ takes effect on the i-th entry of $(X\theta - Y)^T$ and $(X\theta - Y)$. Therefore W is a diagonal matrix which entries are $\frac{1}{2}w^{(i)}$, i.e,

$$\boxed{W = \text{diag}(w^{(1)}, w^{(2)}, ..., w^{(m)})}$$

ii. **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 the value of θ that minimizes $J(\theta)$ is given by $(X^T X)^{-1} X^T Y$. 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 θ that minimizes $J(\theta)$ in closed form as a function of X, W and y.

First, we expand the expression of $J(\theta)$,

$$J(\theta) = (X\theta - Y)^T W(X\theta - Y) = \theta^T X^T WX\theta - \theta^T X^T WY - Y^T WX\theta + Y^T WY,$$

and compute its gradient

$$\begin{aligned}\nabla_\theta J(\theta) &= \nabla_\theta(\theta^T X^T WX\theta - \theta^T X^T WY - Y^T WX\theta + Y^T WY) \\ &= \nabla_\theta(\theta^T X^T WX\theta - \theta^T X^T WY - Y^T WX\theta)\end{aligned}$$

Now, if we take into account that $Y^T WX\theta$ is an scalar, then $Y^T WX\theta = \theta^T X^T W^T Y = \theta^T X^T WY$ (because W is symmetric). Therefore,

$$\begin{aligned}\nabla_\theta J(\theta) &= \nabla_\theta(\theta^T X^T WX\theta - \theta^T X^T WY - Y^T WX\theta) \\ &= \nabla_\theta(\theta^T X^T WX\theta - 2\theta^T X^T WY)\end{aligned}$$

If we define, $g := \theta^T X^T WX\theta - 2\theta^T X^T WY$, then

$$\begin{aligned}dg &= d(\theta^T X^T WX\theta - 2\theta^T X^T WY) \\ &= (d\theta^T X^T WX\theta + \theta^T X^T WX\, d\theta - 2\, d\theta^T X^T WY) \\ &= (\theta^T X^T W^T X + \theta^T X^T W^T X - 2Y^T W^T X)\, d\theta \\ &= 2(X^T WX\theta - X^T WY)^T d\theta \\ &= (\nabla_\theta g)^T d\theta\end{aligned}$$

Thus, the gradient is equal to

$$\nabla_\theta J(\theta) = 2(X^T WX\theta - X^T WY).$$

Finally, the normal equation is,

$$\nabla_\theta J(\theta) = 2(X^T W X\theta - X^T W Y) = 0,$$

$$X^T W X\theta = X^T W Y,$$

$$\boxed{\theta = (X^T W X)^{-1} X^T W Y}.$$

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

The MLE is given by

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

Thus, maximizing $l(\theta)$ is the same as minimizing

$$\frac{1}{2}\sum_i \frac{(y^{(i)} - \theta^T x^{(i)})^2}{(\sigma^{(i)})^2}$$

From this expression we see that setting $w^{(i)} = \frac{1}{(\sigma^{(i)})^2}$, finding the MLE of θ reduces to minimze $J(\theta)$

(b) [10 points] Coding problem. Dataset: `data/ds5_{train,valid,test}.csv` (with 1-D $x^{(i)}$). In `src/p05b_lwr.py`, implement locally weighted linear regression using

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

Train on the train split with $\tau = 0.5$, report MSE on the valid split, and plot predictions (train: blue "x", valid: red "o"). Comment on under/overfitting.

(c) [5 points] Coding problem. In `src/p05c_tau.py`, evaluate various τ values on the validation set, plot as in part (b), report the τ with lowest validation MSE, and finally report the test MSE using that τ.