XCS229i Problem Set 2

Due Monday, 10 August 2020.

I formed study session with Ketki Ambekar.

Guidelines

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 Slack channel, at `http://xcs229i-scpd.slack.com/`
3. Familiarize yourself with the collaboration and honor code policy before starting work.
4. For the coding problems, you may not use any libraries except those defined in the provided started code. In particular, ML-specific libraries such as `scikit-learn` are not permitted.

Submission Instructions

Written Submission: All students must submit an electronic PDF version of the written questions. We highly recommend typesetting your solutions via LaTeX, though it is not required. If you choose to hand write your responses, please make sure they are well organized and legible when scanned. The source LaTeXfor all problem sets is available on GitHub.

Coding Submission: All students must also submit a zip file of their source code. Create a submission using the following bash command:

```
zip -j ps2_submission.zip linearclass/src/gda.py linearclass/src/logreg.py poisson/src/poisson.py
```

If you are **NOT** able to successfully zip your code using the following bash command or do **NOT** have the zip command line tool on your machine, please run the following python script to zip your code as an alternative:

```
python zip_submission.py
```

You should make sure to (1) restrict yourself to only using libraries included in the starter code, and (2) make sure your code runs without errors. Your submission will be evaluated by the auto-grader using a private test set and will be used for verifying the outputs reported in the writeup.

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. **[25 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, along with starter codes provided in the following files:

- `linearclass/src/ds1_train,valid.csv`
- `linearclass/src/ds2_train,valid.csv`
- `linearclass/src/logreg.py`
- `linearclass/src/gda.py`

Each file contains n 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) **[6 point(s) Written]**

In lecture we saw the average empirical loss for logistic regression:

$$J(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right),$$

where $y^{(i)} \in \{0, 1\}$, $h_\theta(x) = g(\theta^T x)$ and $g(z) = 1/(1 + e^{-z})$.

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 is one of the standard ways of showing that the matrix H is positive semi-definite, written "$H \succeq 0$." This implies that J is convex, and has no local minima other than the global one. If you have some other way of showing $H \succeq 0$, you're also welcome to use your method instead of the one above.

BEGIN PROOF HERE

Since $g'(z) = g(z)(1 - g(z))$ and $h(x) = g(\theta^T x)$, it follows that $\partial h(x)/\partial\theta_k = h(x)(1 - h(x))x_k$. Letting $h_\theta(x^{(i)}) = g(\theta^T x^{(i)}) = 1/(1 + \exp(-\theta^T x^{(i)})$, we have

$$\begin{aligned}
&\frac{\partial \log h_\theta(x^{(i)})}{\partial\theta_k} \\
&\quad = \frac{\partial \log g(\theta^T x^{(i)})}{\partial\theta_k} \\
&\quad = \frac{1}{g(\theta^T x^{(i)})} \frac{\partial g(\theta^T x^{(i)})}{\partial\theta_k} \\
&\quad = \frac{1}{g(\theta^T x^{(i)})} g(\theta^T x^{(i)})(1 - g(\theta^T x^{(i)})) \frac{\partial(\theta^T x^{(i)})}{\partial\theta_k} \\
&\quad = (1 - g(\theta^T x^{(i)}))x_k^{(i)}
\end{aligned}$$

$$\begin{aligned}
&\frac{\partial \log(1 - h_\theta(x^{(i)}))}{\partial\theta_k} \\
&\quad = \frac{\partial \log(1 - g(\theta^T x^{(i)}))}{\partial\theta_k} \\
&\quad = \frac{-1}{1 - g(\theta^T x^{(i)})} \frac{\partial g(\theta^T x^{(i)})}{\partial\theta_k} \\
&\quad = \frac{-1}{1 - g(\theta^T x^{(i)})} g(\theta^T x^{(i)})(1 - g(\theta^T x^{(i)})) \frac{\partial(\theta^T x^{(i)})}{\partial\theta_k} \\
&\quad = -g(\theta^T x^{(i)})x_k^{(i)}
\end{aligned}$$

Substituting into our equation for $J(\theta)$, we have

$$\frac{\partial J(\theta)}{\partial \theta_k}$$

$$\begin{aligned}
&= \frac{\partial}{\partial \theta_k} \frac{-1}{n} \sum_{i=1}^{n} \Big(y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \Big) \\
&= \frac{-1}{n} \sum_{i=1}^{n} \left(y^{(i)} \frac{\partial}{\partial \theta_k} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \frac{\partial}{\partial \theta_k} \log(1 - h_\theta(x^{(i)})) \right) \\
&= \frac{-1}{n} \sum_{i=1}^{n} \Big(y^{(i)} (1 - g(\theta^T x^{(i)})) x_k^{(i)} + (1 - y^{(i)}) (-g(\theta^T x^{(i)}) x_k^{(i)}) \Big) \\
\\
&= \frac{-1}{n} \sum_{i=1}^{n} \Big(y^{(i)} x_k^{(i)} - y^{(i)} g(\theta^T x^{(i)}) x_k^{(i)} + y^{(i)} g(\theta^T x^{(i)}) x_k^{(i)} - g(\theta^T x^{(i)}) x_k^{(i)} \Big) \\
&= \frac{-1}{n} \sum_{i=1}^{n} \Big(y^{(i)} x_k^{(i)} - g(\theta^T x^{(i)}) x_k^{(i)} \Big) \\
&= \frac{-1}{n} \sum_{i=1}^{n} \Big(y^{(i)} - g(\theta^T x^{(i)}) \Big) x_k^{(i)}
\end{aligned}$$

Consequently, the (k, l) entry of the Hessian is given by

$$H_{kl}$$

$$\begin{aligned}
&= \frac{\partial^2 J(\theta)}{\partial \theta_k \partial \theta_l} \\
&= \frac{\partial}{\partial \theta_l} \frac{\partial J(\theta)}{\partial \theta_k} \\
&= \frac{\partial}{\partial \theta_l} \frac{-1}{n} \sum_{i=1}^{n} \Big(y^{(i)} - g(\theta^T x^{(i)}) \Big) x_k^{(i)} \\
&= \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial \theta_l} g(\theta^T x^{(i)}) x_k^{(i)} \\
&= \frac{1}{n} \sum_{i=1}^{n} g(\theta^T x^{(i)}) (1 - g(\theta^T x^{(i)})) x_k^{(i)} \frac{\partial}{\partial \theta_l} (\theta^T x^{(i)}) \\
&= \frac{1}{n} \sum_{i=1}^{n} g(\theta^T x^{(i)}) (1 - g(\theta^T x^{(i)})) x_k^{(i)} x_l^{(i)}
\end{aligned}$$

Using the fact that $X_{ij} = x_i x_j$ if and only if $X = xx^T$, we have

$$\begin{aligned} H & \\ &= \sum_k \sum_l H_{kl} \\ &= \sum_k \sum_l \frac{1}{n} \sum_{i=1}^{n} g(\theta^T x^{(i)})(1 - g(\theta^T x^{(i)})) x_k^{(i)} x_l^{(i)} \end{aligned}$$

To prove that H is positive semi-definite, show $z^T H z \geq 0$ for all $z \in \mathbb{R}^d$.

$$\begin{aligned} z^T H z & \\ &= \sum_k \sum_l \frac{1}{n} \sum_{i=1}^{n} g(\theta^T x^{(i)})(1 - g(\theta^T x^{(i)})) z_k x_k^{(i)} x_l^{(i)} z_l \\ &= \frac{1}{n} \sum_{i=1}^{n} g(\theta^T x^{(i)})(1 - g(\theta^T x^{(i)})) \sum_k \sum_l z_k x_k^{(i)} x_l^{(i)} z_l \\ &= \frac{1}{n} \sum_{i=1}^{n} g(\theta^T x^{(i)})(1 - g(\theta^T x^{(i)}))(x^T z)^2 \end{aligned}$$

$g(\theta^T x^{(i)})$ is sigmoid with value between 0 and 1; and $(x^T z)^2 \geq 0$; therefore, $H \succeq 0$.

END PROOF

(b) **[2 point(s) Coding]** Follow the instructions in `linearclass/src/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 $\|\theta_k - \theta_{k-1}\|_1 < \epsilon$, where $\epsilon = 1 \times 10^{-5}$. Make sure to write your model's predicted probabilities on the validation set to the file specified in the code.

To verify a correct implementation, consider creating a plot of the **validation data** with x_1 on the horizontal axis and x_2 on the vertical axis. To visualize the two classes, use a different symbol for examples $x^{(i)}$ with $y^{(i)} = 0$ than for those with $y^{(i)} = 1$. On the same figure, plot the decision boundary found by logistic regression (i.e, line corresponding to $p(y|x) = 0.5$).

Your plot should look similar to the following:

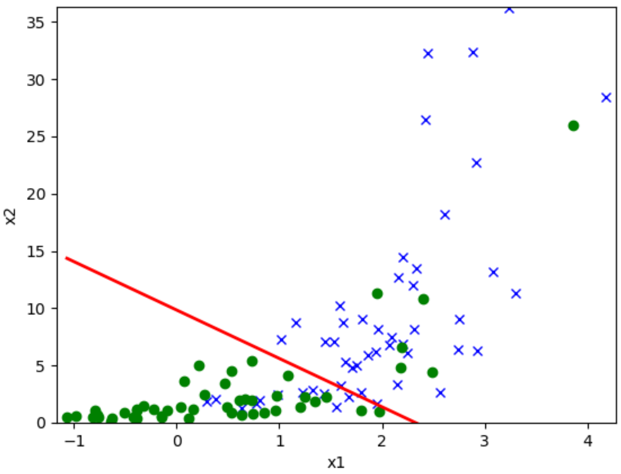


Figure 1: Separating hyperplane for logistic regression on Dataset 1

(c) [**4 point(s) Written**] Recall that in GDA we model the joint distribution of (x, y) by the following equations:

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

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

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

where ϕ, μ_0, μ_1, and Σ are the parameters of our model.

Suppose we have already fit ϕ, μ_0, μ_1, and Σ, and now want to predict y given a new point x. To show that GDA results in a classifier that has a linear decision boundary, show the posterior distribution can be written as

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

where $\theta \in \mathbb{R}^d$ and $\theta_0 \in \mathbb{R}$ are appropriate functions of ϕ, Σ, μ_0, and μ_1.

BEGIN PROOF HERE

For shorthand, we let $\mathcal{H} = \{\phi, \Sigma, \mu_0, \mu_1\}$ denote the parameters for the problem. Since the given formulae are conditioned on y, use Bayes rule to get:

$$
\begin{aligned}
&p(y=1|x;\mathcal{H})\\
&\qquad= \frac{p(x|y=1;\mathcal{H})p(y=1;\mathcal{H})}{p(x;\mathcal{H})}\\
&\qquad= \frac{p(x|y=1;\mathcal{H})p(y=1;\mathcal{H})}{p(x|y=1;\mathcal{H})p(y=1;\mathcal{H})+p(x|y=0;\mathcal{H})p(y=0;\mathcal{H})}\\
&\\
&\qquad= \frac{\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)\phi}{\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)\phi+\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right)(1-\phi)}\\
&\qquad= \frac{\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)\phi}{\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)\phi+\exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right)(1-\phi)}\\
&\qquad= \frac{1}{1+\frac{\exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right)(1-\phi)}{\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)\phi}}\\
&\qquad= \frac{1}{1+\frac{\exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right)\exp(\log(1-\phi))}{\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)\exp(\log(\phi))}}\\
&\qquad= \frac{1}{1+\exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)+\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)+\log((1-\phi)/\phi)\right)}\\
&\qquad= \frac{1}{1+\exp\left(-\frac{1}{2}(x^T-\mu_0^T)\Sigma^{-1}(x-\mu_0)+\frac{1}{2}(x^T-\mu_1^T)\Sigma^{-1}(x-\mu_1)+\log((1-\phi)/\phi)\right)}\\
&\qquad= \frac{1}{1+\exp\left(D+\log((1-\phi)/\phi)\right)}
\end{aligned}
$$

$$
\begin{aligned}
D &= -\frac{1}{2}(x^T-\mu_0^T)\Sigma^{-1}(x-\mu_0)+\frac{1}{2}(x^T-\mu_1^T)\Sigma^{-1}(x-\mu_1)\\
&= -\frac{1}{2}(x^T\Sigma^{-1}x-x^T\Sigma^{-1}\mu_0-\mu_0^T\Sigma^{-1}x+\mu_0^T\Sigma^{-1}\mu_0)+\frac{1}{2}(x^T\Sigma^{-1}x-x^T\Sigma^{-1}\mu_1-\mu_1^T\Sigma^{-1}x+\mu_1^T\Sigma^{-1}\mu_1)\\
&= -\frac{1}{2}(x^T\Sigma^{-1}\mu_1-x^T\Sigma^{-1}\mu_0+\mu_1^T\Sigma^{-1}x-\mu_0^T\Sigma^{-1}x+\mu_0^T\Sigma^{-1}\mu_0-\mu_1^T\Sigma^{-1}\mu_1)\\
&= -\frac{1}{2}(x^T\Sigma^{-1}(\mu_1-\mu_0)+(\mu_1-\mu_0)^T\Sigma^{-1}x+\mu_0^T\Sigma^{-1}\mu_0-\mu_1^T\Sigma^{-1}\mu_1)\\
&= -\frac{1}{2}(2(\mu_1-\mu_0)^T\Sigma^{-1}x+\mu_0^T\Sigma^{-1}\mu_0-\mu_1^T\Sigma^{-1}\mu_1)\\
&= -(\mu_1-\mu_0)^T\Sigma^{-1}x+\frac{1}{2}(\mu_1^T\Sigma^{-1}\mu_1-\mu_0^T\Sigma^{-1}\mu_0)
\end{aligned}
$$

$$
\begin{aligned}
&p(y=1|x;\mathcal{H})\\
&\qquad = \frac{1}{1+\exp\big(-(\mu_1-\mu_0)^T\Sigma^{-1}x+\frac{1}{2}(\mu_1^T\Sigma^{-1}\mu_1-\mu_0^T\Sigma^{-1}\mu_0)+\log((1-\phi)/\phi)\big)}\\
&\qquad = \frac{1}{1+\exp\big(-\big((\mu_1-\mu_0)^T\Sigma^{-1}x+\frac{1}{2}(\mu_0^T\Sigma^{-1}\mu_0-\mu_1^T\Sigma^{-1}\mu_1)-\log((1-\phi)/\phi)\big)\big)}
\end{aligned}
$$

$$
\begin{aligned}
\theta^T &= (\mu_1-\mu_0)^T\Sigma^{-1}\\
\theta_0 &= \frac{1}{2}(\mu_0^T\Sigma^{-1}\mu_0-\mu_1^T\Sigma^{-1}\mu_1)-\log((1-\phi)/\phi)
\end{aligned}
$$

END PROOF

(d) **[5 point(s) Written]** Given the dataset, we claim that the maximum likelihood estimates of the parameters are given by

$$
\begin{aligned}
\phi &= \frac{1}{n}\sum_{i=1}^{n} 1\{y^{(i)}=1\}\\
\mu_0 &= \frac{\sum_{i=1}^{n} 1\{y^{(i)}=0\}x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)}=0\}}\\
\mu_1 &= \frac{\sum_{i=1}^{n} 1\{y^{(i)}=1\}x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)}=1\}}\\
\Sigma &= \frac{1}{n}\sum_{i=1}^{n}(x^{(i)}-\mu_{y^{(i)}})(x^{(i)}-\mu_{y^{(i)}})^T
\end{aligned}
$$

The log-likelihood of the data is

$$
\begin{aligned}
\ell(\phi,\mu_0,\mu_1,\Sigma) &= \log\prod_{i=1}^{n} p(x^{(i)},y^{(i)};\phi,\mu_0,\mu_1,\Sigma)\\
&= \log\prod_{i=1}^{n} p(x^{(i)}|y^{(i)};\mu_0,\mu_1,\Sigma)p(y^{(i)};\phi).
\end{aligned}
$$

By maximizing ℓ with respect to the four parameters, prove that the maximum likelihood estimates of ϕ, μ_0, μ_1, and Σ are indeed as given in the formulas above. (You may assume that there is at least one positive and one negative example, so that the denominators in the definitions of μ_0 and μ_1 above are non-zero.)

BEGIN PROOF HERE

First, derive the expression for the log-likelihood of the training data:

$$\ell(\phi, \mu_0, \mu_1, \Sigma)$$

$$= \log \prod_{i=1}^{n} p(x^{(i)}|y^{(i)}; \mu_0, \mu_1, \Sigma) p(y^{(i)}; \phi)$$

$$= \sum_{i=1}^{n} \log p(x^{(i)}|y^{(i)}; \mu_0, \mu_1, \Sigma) + \sum_{i=1}^{n} \log p(y^{(i)}; \phi)$$

$$= \sum_{i=1}^{n} \log \left(\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x^{(i)} - \mu_{y^{(i)}})^T \Sigma^{-1} (x^{(i)} - \mu_{y^{(i)}}) \right) \right) + \sum_{i=1}^{n} \log \left(\phi^{y^{(i)}} (1-\phi)^{(1-y^{(i)})} \right)$$

$$= \sum_{i=1}^{n} -\frac{n}{2} \log(2\pi) - \frac{1}{2}\log(|\Sigma|) - \frac{1}{2}(x^{(i)} - \mu_{y^{(i)}})^T \Sigma^{-1} (x^{(i)} - \mu_{y^{(i)}}) + y^{(i)} \log(\phi) + (1 - y^{(i)}) \log(1-\phi)$$

Now, the likelihood is maximized by setting the derivative (or gradient) with respect to each of the parameters to zero.

For ϕ:

$$\frac{\partial \ell}{\partial \phi}$$

$$= \frac{\partial}{\partial \phi} \sum_{i=1}^{n} y^{(i)} \log(\phi) + (1 - y^{(i)}) \log(1-\phi)$$

$$= \sum_{i=1}^{n} \frac{y^{(i)}}{\phi} - \frac{(1 - y^{(i)})}{(1-\phi)}$$

Setting this equal to zero and solving for ϕ gives the maximum likelihood estimate.

$$\begin{aligned}
0 &= \sum_{i=1}^{n} \frac{1\{y^{(i)}=1\}}{\phi} - \frac{(1-1\{y^{(i)}=1\})}{(1-\phi)} \\
&= \frac{\sum_{i=1}^{n} 1\{y^{(i)}=1\}}{\phi} - \frac{(n-\sum_{i=1}^{n} 1\{y^{(i)}=1\})}{(1-\phi)} \\
&= \sum_{i=1}^{n} 1\{y^{(i)}=1\} - \phi \sum_{i=1}^{n} 1\{y^{(i)}=1\} - n\phi + \phi \sum_{i=1}^{n} 1\{y^{(i)}=1\} \\
&= \sum_{i=1}^{n} 1\{y^{(i)}=1\} - n\phi \\
\phi &= \frac{1}{n} \sum_{i=1}^{n} 1\{y^{(i)}=1\}
\end{aligned}$$

For μ_0:
Hint: Remember that Σ (and thus Σ^{-1}) is symmetric.

$\nabla_{\mu_0}\ell$

$$\begin{aligned}
&= \nabla_{\mu_0} \sum_{i=1}^{n} -\frac{1}{2}(x^{(i)}-\mu_0)^T\Sigma^{-1}(x^{(i)}-\mu_0) \\
&= -\frac{1}{2}\nabla_{\mu_0} \sum_{i=1}^{n} \left(x^{(i)T}\Sigma^{-1}x^{(i)} - x^{(i)T}\Sigma^{-1}\mu_0 - \mu_0^T\Sigma^{-1}x^{(i)} + \mu_0^T\Sigma^{-1}\mu_0 \right) \\
&= -\frac{1}{2} \sum_{i=1}^{n} \nabla_{\mu_0} \left(-2x^{(i)T}\Sigma^{-1}\mu_0 + \Sigma^{-1}\mu_0^2 \right) \\
&= -\frac{1}{2} \sum_{i=1}^{n} \left(-2x^{(i)T}\Sigma^{-1} + 2\Sigma^{-1}\mu_0 \right) \\
&= \sum_{i=1}^{n} \left(\Sigma^{-1}x^{(i)} - \Sigma^{-1}\mu_0 \right)
\end{aligned}$$

Setting this gradient to zero gives the maximum likelihood estimate for μ_0.

$$
\begin{aligned}
0 &= \sum_{i=1}^{n} \left(\Sigma^{-1} x^{(i)} - \Sigma^{-1} \mu_0 \right) \\
&= \sum_{i=1}^{n} 1\{y^{(i)} = 0\} \Sigma^{-1} x^{(i)} - \sum_{i=1}^{n} 1\{y^{(i)} = 0\} \Sigma^{-1} \mu_0 \\
\mu_0 &= \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 0\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = 0\}}
\end{aligned}
$$

For μ_1:
Hint: Remember that Σ (and thus Σ^{-1}) is symmetric.

$$
\begin{aligned}
&\nabla_{\mu_1} \ell \\
&\qquad = \nabla_{\mu_1} \sum_{i=1}^{n} -\frac{1}{2} (x^{(i)} - \mu_1)^T \Sigma^{-1} (x^{(i)} - \mu_1) \\
&\qquad = -\frac{1}{2} \nabla_{\mu_1} \sum_{i=1}^{n} \left(x^{(i)T} \Sigma^{-1} x^{(i)} - x^{(i)T} \Sigma^{-1} \mu_1 - \mu_1^T \Sigma^{-1} x^{(i)} + \mu_1^T \Sigma^{-1} \mu_1 \right) \\
&\qquad = -\frac{1}{2} \sum_{i=1}^{n} \nabla_{\mu_1} \left(-2 x^{(i)T} \Sigma^{-1} \mu_1 + \Sigma^{-1} \mu_1^2 \right) \\
&\qquad = -\frac{1}{2} \sum_{i=1}^{n} \left(-2 x^{(i)T} \Sigma^{-1} + 2 \Sigma^{-1} \mu_1 \right) \\
&\qquad = \sum_{i=1}^{n} \left(\Sigma^{-1} x^{(i)} - \Sigma^{-1} \mu_1 \right)
\end{aligned}
$$

Setting this gradient to zero gives the maximum likelihood estimate for μ_1.

$$\begin{aligned}
0 &= \sum_{i=1}^{n} \left(\Sigma^{-1}x^{(i)} - \Sigma^{-1}\mu_1\right) \\
&= \sum_{i=1}^{n} 1\{y^{(i)} = 1\}\Sigma^{-1}x^{(i)} - \sum_{i=1}^{n} 1\{y^{(i)} = 1\}\Sigma^{-1}\mu_1 \\
\mu_1 &= \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}}
\end{aligned}$$

For Σ, we find the gradient with respect to $S = \Sigma^{-1}$ rather than Σ just to simplify the derivation (note that $|S| = \frac{1}{|\Sigma|}$). You should convince yourself that the maximum likelihood estimate S_n found in this way would correspond to the actual maximum likelihood estimate Σ_n as $S_n^{-1} = \Sigma_n$.
Hint: You may need the following identities:

$$\nabla_S |S| = |S|(S^{-1})^T$$

$$\nabla_S b_i^T S b_i = \nabla_S tr\left(b_i^T S b_i\right) = \nabla_S tr\left(S b_i b_i^T\right) = b_i b_i^T$$

$$\begin{aligned}
&\nabla_S \ell \\
&\quad = \nabla_S \sum_{i=1}^{n} -\frac{1}{2}\log(|\Sigma|) - \frac{1}{2}(x^{(i)} - \mu_{y^{(i)}})^T \Sigma^{-1}(x^{(i)} - \mu_{y^{(i)}}) \\
&\quad = \nabla_S \sum_{i=1}^{n} \frac{1}{2}\log(|S|) - \frac{1}{2}(x^{(i)} - \mu_{y^{(i)}})^T S(x^{(i)} - \mu_{y^{(i)}}) \\
&\quad = \sum_{i=1}^{n} \nabla_S \frac{1}{2}\log(|S|) - \nabla_S \frac{1}{2}(x^{(i)} - \mu_{y^{(i)}})^T S(x^{(i)} - \mu_{y^{(i)}}) \\
&\quad = \sum_{i=1}^{n} \nabla_S \frac{1}{2}\log(|S|) - \nabla_S \frac{1}{2}S(x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T \\
&\quad = \sum_{i=1}^{n} \frac{1}{2|S|} - \frac{1}{2}(x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T
\end{aligned}$$

Next, substitute $\Sigma = S^{-1}$. Setting this gradient to zero gives the required maximum likelihood estimate for Σ.

$$
\begin{aligned}
0 &= \sum_{i=1}^{n} \frac{1}{2}\Sigma - \frac{1}{2}(x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T \\
&= n\Sigma - \sum_{i=1}^{n}(x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T \\
\Sigma &= \frac{1}{n}\sum_{i=1}^{n}(x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T
\end{aligned}
$$

END PROOF

(e) **[2 point(s) Coding]** In `linearclass/src/gda.py`, fill in the code to calculate ϕ, μ_0, μ_1, and Σ, use these parameters to derive θ, and use the resulting GDA model to make predictions on the validation set. Make sure to write your model's predictions on the validation set to the file specified in the code.

To verify a correct implementation, consider creating a plot of the **validation data** with x_1 on the horizontal axis and x_2 on the vertical axis. To visualize the two classes, use a different symbol for examples $x^{(i)}$ with $y^{(i)} = 0$ than for those with $y^{(i)} = 1$. On the same figure, plot the decision boundary found by GDA (i.e, line corresponding to $p(y|x) = 0.5$).

Your plot should look similar to the following:

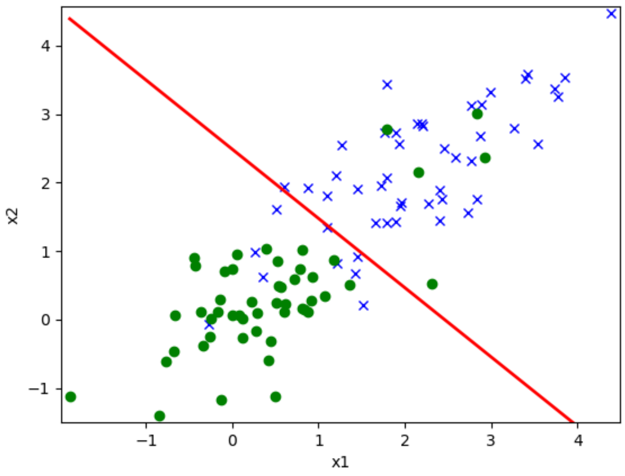
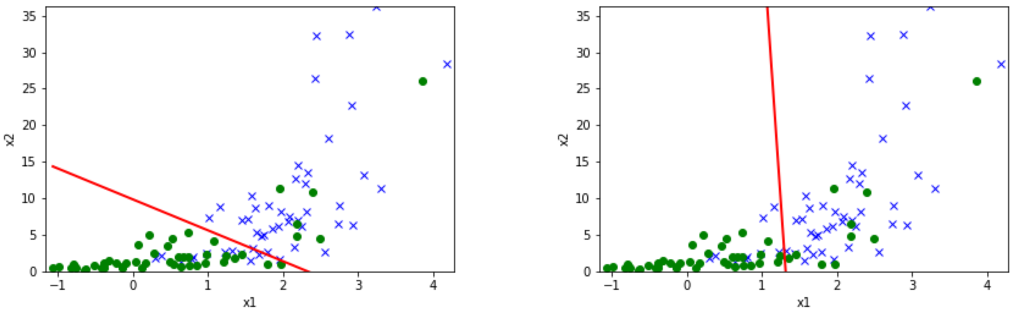


Figure 2: Separating hyperplane for GDA on Dataset 1 (Note: This is for reference only. You are not required to submit a plot.)

(f) **[1 point(s) Written]** For Dataset 1, compare the validation set plots obtained in part (b) and part (e) from logistic regression and GDA respectively, and briefly comment on your observation in a couple of lines.



(a) Logistic Regression

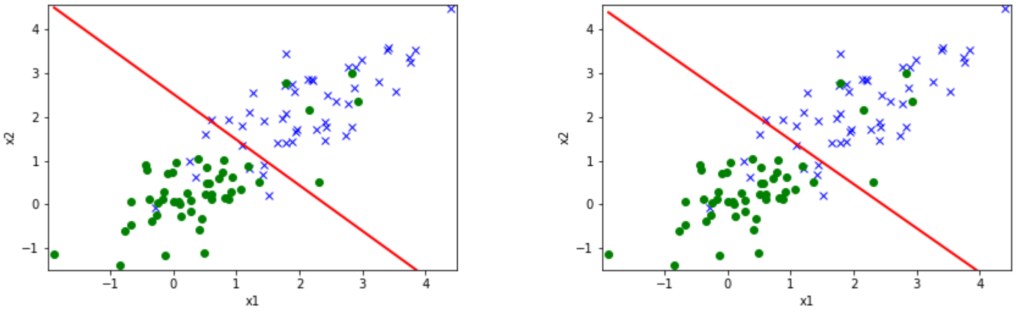
(b) GDA

Figure 3: Dataset 1 Comparison

Logistic regression did a better job at labeling dataset 1. GDA performed less well because we made the assumption that the covariance matrix Σ was identical for both 0 and 1 labels. From the distribution of the data, we can see that it is not Gaussian distribution and 0 and 1 data points do not have identical Σ.

(g) **[4 point(s) Written & Coding]** Repeat the steps in part (b) and part (e) for Dataset 2. Create similar plots on the **validation set** of Dataset 2 and include those plots in your writeup.

On which dataset does GDA seem to perform worse than logistic regression? Why might this be the case?



(a) Logistic Regression

(b) GDA

Figure 4: Dataset 2 Comparison

GDA performed worse than logistic regression for dataset 1. This is the case because in dataset 1, the data is not Gaussian distributed. GDA performed less well because we made the assumption that the covariance matrix Σ was identical for both 0 and 1 labels. From the distribution of the data, we can see that it is not Gaussian distribution and 0 and 1 data points do not have identical Σ.

(h) **[1 point(s) Written]** For the dataset where GDA performed worse in parts (f) and (g), can you find a transformation of the $x^{(i)}$'s such that GDA performs significantly better? What might this transformation be?

In dataset 1, x2 seems to have an exponential characteristic. We can apply a transformation to $x^{(i)}$'s such that GDA performs significantly better. The transformation is to apply log() to x2 in dataset 1.

2. **[15 points] Poisson Regression**

In this question we will construct another kind of a commonly used GLM, which is called Poisson Regression. In a GLM, the choice of the exponential family distribution is based on the kind of problem at hand. If we are solving a classification problem, then we use an exponential family distribution with support over discrete classes (such as Bernoulli, or Categorical). Similarly, if the output is real valued, we can use Gaussian or Laplace (both are in the exponential family). Sometimes the desired output is to predict counts. E.g., predicting the number of emails expected in a day, the number of customers expected to enter a store in the next hour, etc. based on input features (also called covariates). You may recall that a probability distribution with support over integers (i.e. counts) is the Poisson distribution, and it also happens to be in the exponential family.

In the following sub-problems, we will start by showing that the Poisson distribution is in the exponential family, derive the functional form of the hypothesis, derive the update rules for training models, and finally using the provided dataset to train a real model and make predictions on the test set.

(a) **[2 point(s) Written]** Consider the Poisson distribution parameterized by λ:

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

(Here y has positive integer values and $y!$ is the factorial of y.) Show that the Poisson distribution is in the exponential family, and clearly state the values for $b(y)$, η, $T(y)$, and $a(\eta)$.

BEGIN PROOF HERE

$$\begin{aligned}
p(y; \lambda)& \\
&= \frac{e^{-\lambda}\lambda^y}{y!} \\
&= \exp\left(-\lambda + y\log(\lambda) - \log(y!)\right) \\
&= \frac{\exp\left(y\log(\lambda) - \lambda\right)}{\exp\left(\log(y!)\right)} \\
&= \frac{1}{y!}\exp\left(\log(\lambda)y - \lambda\right)
\end{aligned}$$

$$\begin{aligned}
b(y) &= \frac{1}{y!} \\
\eta &= \log(\lambda) \\
T(y) &= y \\
a(\eta) &= \lambda = e^{\eta}
\end{aligned}$$

END PROOF

(b) **[1 point(s) Written]** 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.)

$$\begin{aligned} \mathbb{E}[T(y);\eta] &= \mathbb{E}[y;\eta] \\ \mathbb{E}[y;\eta] &= \lambda \\ &= e^{\eta} \end{aligned}$$

(c) **[7 point(s) Written]** For a training set $\{(x^{(i)}, y^{(i)});\ i = 1, \ldots, n\}$, let the log-likelihood of an example be $\log p(y^{(i)}|x^{(i)};\theta)$. By taking the derivative of the log-likelihood with respect to θ_j, derive the stochastic gradient ascent update rule for learning using a GLM model with Poisson responses y and the canonical response function.

BEGIN PROOF HERE

The log-likelihood of an example $(x^{(i)}, y^{(i)})$ is defined as $\ell(\theta) = \log p(y^{(i)}|x^{(i)};\theta)$. To derive the stochastic gradient ascent rule, use the results in part (a) and the standard GLM assumption that $\eta = \theta^T x$.

$$\begin{aligned} \frac{\partial \ell(\theta)}{\partial \theta_j} &= \frac{\partial \log p(y^{(i)}|x^{(i)};\theta)}{\partial \theta_j} \\ &= \frac{\partial \log \left(\frac{1}{y^{(i)}!} \exp(\eta^T y^{(i)} - e^{\eta}) \right)}{\partial \theta_j} \\ &= \frac{\partial}{\partial \theta_j} \left(\log(\frac{1}{y!}) + \eta^T y^{(i)} - e^{\eta} \right) \\ &= \frac{\partial}{\partial \theta_j} \left(\log(\frac{1}{y!}) + \theta^T x^{(i)} y^{(i)} - e^{\theta^T x^{(i)}} \right) \\ &= y^{(i)} x^{(i)} - e^{\theta^T x^{(i)}} x^{(i)} \\ &= \left(y^{(i)} - e^{\theta^T x^{(i)}} \right) x^{(i)} \end{aligned}$$

The stochastic gradient ascent update rule is

$$\theta_j := \theta_j + \alpha \frac{\partial \ell(\theta)}{\partial \theta_j},$$

which reduces here to:

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - e^{\theta^T x^{(i)}} \right) x^{(i)}$$

END PROOF

(d) **[5 point(s) Coding]**
Consider a website that wants to predict its daily traffic. The website owners have collected a dataset of past traffic to their website, along with some features which they think are useful in predicting the number of visitors per day. The dataset is split into train/valid sets and the starter code is provided in the following files:

- `poisson/src/train,valid.csv`
- `poisson/src/poisson.py`

We will apply Poisson regression to model the number of visitors per day. Note that applying Poisson regression in particular assumes that the data follows a Poisson distribution whose natural parameter is a linear combination of the input features (*i.e.,* $\eta = \theta^T x$). In `poisson/src/poisson.py`, implement Poisson regression for this dataset and use *full batch gradient ascent* to maximize the log-likelihood of θ. For the stopping criterion, check if the change in parameters has a norm smaller than a small value such as 10^{-5}.

Using the trained model, predict the expected counts for the **validation set**. To verify a correct implementation, consider creating a scatter plot between the true counts vs predicted counts (on the validation set). In the scatter plot, let x-axis be the true count and y-axis be the corresponding predicted expected count. Note that the true counts are integers while the expected counts are generally real values.

Your plot should look similar to the following:

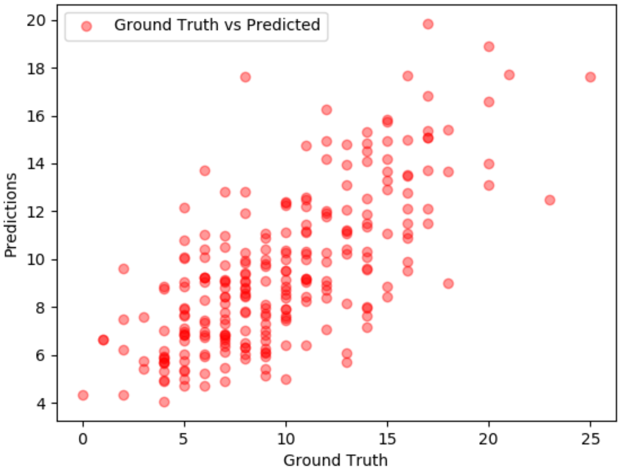


Figure 5: Ground Truth vs Prediction plot on the validation set (Note: This is for reference only. You are not required to submit a plot.)