CS 229: Assignment 1

Santiago de Buen

October 11, 2020

1 Linear Classifiers (logistic regression and GDA)

(a) **Proof that the Hessian of a logistic regression is positive semidefinitive**

The logistic regression is given by:

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

Taking the gradient:

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

Using chain rule, we know that

$$\frac{\partial log(h_\theta(x^{(i)}))}{\partial\theta} = \frac{\partial log(g(\theta^T x))}{\partial\theta} = \frac{\partial log(g(\theta^T x))}{g(\theta^T x)}\frac{\partial g(\theta^T x)}{\partial\theta^T x}\frac{\partial\theta^T x}{\partial\theta}$$

Substituting derivatives and simplifying,

$$\frac{\partial log(h_\theta(x^{(i)}))}{\partial\theta} = (1-g(\theta^T x))x^{(i)}$$

Analogously, we can use the same process to show that

$$\frac{\partial log(1-h_\theta(x^{(i)}))}{\partial\theta} = -g(\theta^T x)x^{(i)}$$

Substituting both partial derivatives into (1):

$$\nabla_\theta J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\Big(y^{(i)}(1-g(\theta^T x))x^{(i)} - (1-y^{(i)})g(\theta^T x)x^{(i)}\Big)$$

Distributing and simplifying, we obtain a simple expression for the gradient

$$\nabla_\theta J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\Big(g(\theta^T x)) - y^{(i)}\Big)\,x^{(i)} \tag{2}$$

We are now ready to calculate the Hessian

$$\nabla_\theta^2 J(\theta) = \frac{\partial J(\theta)}{\partial \theta^T \partial \theta} = H = \frac{1}{n}\sum_{i=1}^{n} x^{(i)} \frac{\partial g(\theta^T x)}{\partial \theta^T x} \frac{\partial \theta^T x}{\partial \theta}$$

$$H = \frac{1}{n}\sum_{i=1}^{n} x^{(i)} x^{(i)T} (1 - h_\theta(x^{(i)})) h_\theta(x^{(i)})$$

We want to prove that the Hessian is positive semidefinite. That is, for any vector z, it holds true that

$$z^T H z \geq 0$$

Lets pause for a moment and first show that $\sum_i \sum_j z_i x_i x_j z_j = (x^T z)^2 \geq 0$:

$$\sum_i \sum_j z_i x_i z_j x_j = \sum_i z_i x_i \sum_j z_j x_j = (x^T z)(x^T z) = (x^T z)^2 \geq 0$$

We are ready to show that $H \succeq 0$

$$z^T H z = z^T \left(\frac{1}{n}\sum_{i=1}^{n} x^{(i)} x^{(i)T} (1 - h_\theta(x^{(i)})) h_\theta(x^{(i)}) \right) z$$

Expanding the matrices into sum notation

$$z^T H z = \frac{1}{n}\sum_i \sum_j (1 - h_\theta(x^{(i)})) h_\theta(x^{(i)}) z_i x_i x_j z_j$$

$$z^T H z = \frac{1}{n}\sum_i \sum_j (1 - h_\theta(x^{(i)})) h_\theta(x^{(i)}) (x^T z)^2$$

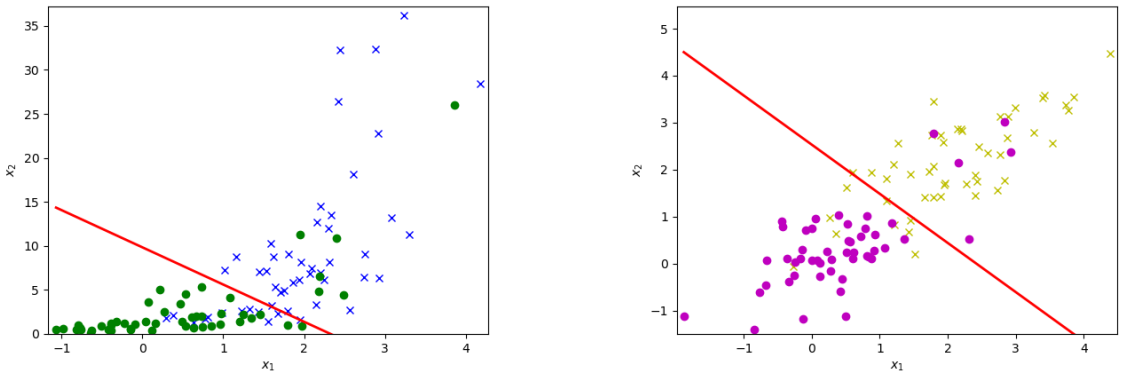
where

$$(1 - h_\theta(x^{(i)})) \in (0, 1)$$
$$(h_\theta(x^{(i)})) \in (0, 1)$$
$$(x^T z)^2 \geq 0$$
$$n > 0$$

Because this is a multiplication of positive terms, it implies that $z^T H z \succeq 0$

(b) **Coding Problem: Logistic Regression**

The coding exercise produces the following two figures:



(a) Dataset 1 (b) Dataset 2

Figure 1: Decision Boundary Produced by Logistic Regression

(c) **GDA joint distribution**

From Bayes rule, we know that

$$P(y=1|x) = \frac{P(x|y=1)P(y=1)}{P(x)}$$

Expanding, we obtain

$$P(y=1|x;...) = \frac{\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)\phi}{\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)\phi + \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right)(1-\phi)}$$

We use the following variable substitutions to simplify algebra

$$z_0 = -\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)$$
$$z_1 = -\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)$$

Simplifying, we obtain

$$P(y=1|x;...) = \frac{exp(z_1)\phi}{exp(z_1)\phi + exp(z_0)(1-\phi)}$$

Rearranging the equation,

$$P(y=1|x;...) = \frac{1}{1+\frac{1-\phi}{\phi}exp(z_0-z_1)}$$

To obtain a similar form to the logistic regression, we can rearrange the denominator

$$P(y=1|x;...) = \frac{1}{1+exp(log(\frac{1-\phi}{\phi}) + z_0 - z_1)} \tag{3}$$

Now, we just need to show that what's inside the exponent can be written as a combination of $\theta^T x$ and θ_0. Lets substitute z_0 and z_1 back in and simplify

$$z_0 - z_1 = -\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0) + \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)$$

$$= -\frac{1}{2}\left[x^T\Sigma^{-1}x - \mu_0^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu_0 + \mu_0^T\Sigma^{-1}\mu_0 - x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu_1 + \mu_1^T\Sigma^{-1}x - \mu_1^T\Sigma^{-1}\mu_1\right]$$

Rearranging and simplifying, we obtain

$$= -\frac{1}{2}\left[2(\mu_1-\mu_0)^T\Sigma^{-1}x + (\mu_0+\mu_1)^T\Sigma^{-1}(\mu_0-\mu_1)\right]$$

Adding back the logarithmic term and distributing the multiplication, we obtain an expression for what's inside the exponent in (3).

$$= exp\left[-(\mu_1-\mu_0)^T\Sigma^{-1}x - \frac{1}{2}(\mu_0+\mu_1)^T\Sigma^{-1}(\mu_0-\mu_1) + log(\frac{1-\phi}{\phi})\right]$$

Where

$$\theta^T = (\mu_1-\mu_0)\Sigma^{-1}$$
$$\theta_0 = \frac{1}{2}(\mu_0+\mu_1)^T\Sigma^{-1}(\mu_0-\mu_1) - log(\frac{1-\phi}{\phi})$$

Therefore, we can rewrite (3) as

$$P(y=1|x;...) = \frac{1}{1+e^{-(\theta^T x+\theta_0)}} \tag{4}$$

(d) **Optimization of Maximum Likelihood Function**

Distributing the logarithm into the multiplication, we obtain

$$\ell(\phi,\mu_0,\mu_1,\Sigma) = \sum_{i=1}^{n}\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) - \frac{d}{2}log(2\pi) - \frac{1}{2}log(|\Sigma|) + ylog(\phi) + (1-y)log(1-\phi)\right) \tag{5}$$

Now we can take the partial derivatives with respect to ϕ, μ_0, μ_1, Σ and set them to zero.

Parameter ϕ:

$$\frac{\partial\ell}{\partial\phi} : \sum_{i=1}^{n}\left(1\{y^i=1\}\frac{1}{\phi} - (1-1\{y^i=1\})\frac{1}{1-\phi}\right) = 0$$

$$\sum_{i=1}^{n}\left(\frac{1\{y^i=1\}(1-\phi)-(1-1\{y^i=1\})\phi}{\phi(1-\phi)}\right)=0$$

Simplifying, and assuming $\phi \neq 0, 1$:

$$\sum_{i=1}^{n}(1\{y^i=1\}-\phi)=0$$

$$\sum_{i=1}^{n}\phi=\sum_{i=1}^{n}1\{y^i=1\}$$

$$n\phi=\sum_{i=1}^{n}1\{y^i=1\}$$

$$\phi=\frac{1}{n}\sum_{i=1}^{n}1\{y^i=1\} \tag{6}$$

Parameter μ:

We know that $\nabla_x x^T Ax = 2Ax$ if A is symmetric. Because we know Σ is symmetric, we can use this property.

Expanding the first term of (5):

$$\sum_{i=1}^{n}\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)=-\frac{1}{2}\sum_{i=1}^{n}(x^T\Sigma^{-1}x-x^T\Sigma^{-1}\mu-\mu^T\Sigma^{-1}x+\mu^T\Sigma^{-1}\mu)$$

Taking the derivative of (5) with respect to μ_0, we obtain:

$$\frac{\partial\ell}{\partial\mu_0}:-\frac{1}{2}\sum_{i=1}^{n}\left(-x^T\Sigma^{-1}\cdot 1\{y^i=0\}-\Sigma^{-1}x\cdot 1\{y^i=0\}+2\Sigma^{-1}\mu_0\cdot 1\{y^i=0\}\right)=0$$

Because Σ is symmetric, $x^T\Sigma^{-1}=\Sigma^{-1}x$

$$\sum_{i=1}^{n}2\Sigma^{-1}\mu_0\cdot 1\{y^i=0\}=\sum_{i=1}^{n}2\Sigma^{-1}x^{(i)}\cdot 1\{y^i=0\}$$

Simplifying, we obtain

$$\sum_{i=1}^{n}\mu_0\cdot 1\{y^i=0\}=\sum_{i=1}^{n}x^{(i)}\cdot 1\{y^i=0\}$$

$$\mu_0=\frac{\sum_{i=1}^{n}1\{y^i=0\}x^{(i)}}{\sum_{i=1}^{n}1\{y^i=0\}} \tag{7}$$

Similarly, μ_1 is an analogous case,where only the indicator function changes

$$\mu_1 = \frac{\sum_{i=1}^{n} 1\{y^i = 1\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^i = 1\}} \tag{8}$$

Parameter Σ:

Keeping only the relevant terms from (5) and substituting $S = \Sigma^{-1}$:

$$\sum_{i=1}^{n} \left(-\frac{1}{2}(x-\mu)^T S(x-\mu) - \frac{1}{2} log(|S^{-1}|) \right)$$

Recall identity $|X^{-1}| = \frac{1}{|x|}$ for invertible x. Using the identity and simplifying:

$$\sum_{i=1}^{n} \left(-\frac{1}{2}(x-\mu)^T S(x-\mu) + \frac{1}{2} log(|S|) \right)$$

Also recall that $\nabla_x |x| = |x|(x^{-1})^T$. Taking the derivative, we obtain

$$\frac{\partial \ell}{\partial \Sigma} : \sum_{i=1}^{n} \left(-\frac{1}{2}(x-\mu)^T (x-\mu) + \frac{1}{2}\frac{1}{|S|}|S|(S^{-1})^T \right) = 0$$

$$\sum_{i=1}^{n} \left((x-\mu)(x-\mu)^T - S^{-1} \right) = 0$$

Substituting back for $S^{-1} = \Sigma$ and distributing the sum:

$$\sum_{i=1}^{n} ((x-\mu)(x-\mu)^T) - \sum_{i=1}^{n} \Sigma = 0$$

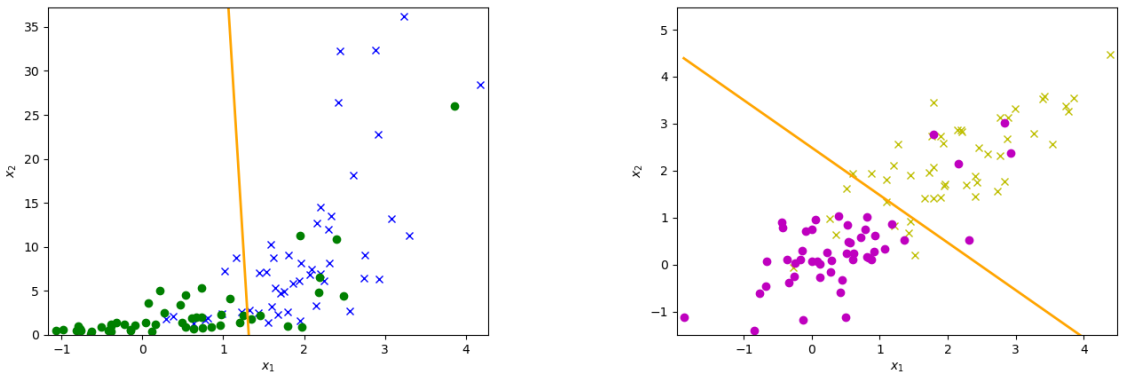
$$n\Sigma = \sum_{i=1}^{n} ((x-\mu)(x-\mu)^T)$$

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} ((x-\mu)(x-\mu)^T) \tag{9}$$

Collectively, (6), (7), (8), (9) are the maximum likelihood estimates of the parameters.

(e) **Coding Problem: Gaussian Discriminant Analysis**

The coding exercise produces the following two figures:

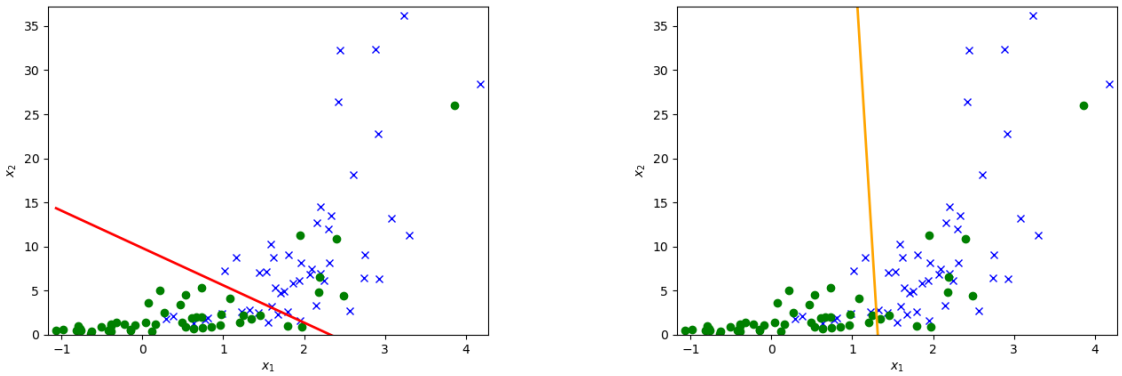


(a) Dataset 1 (b) Dataset 2

Figure 2: Decision Boundary Produced by Gaussian Discriminant Analysis

(f) **Dataset 1: Compare validation set plots obtained in logistic regression and GDA**

We can now compare the graphs produced by logistic regression and GDA directly and note the difference in prediction.



(a) Logistic Regression (b) Gaussian Discriminant Analysis

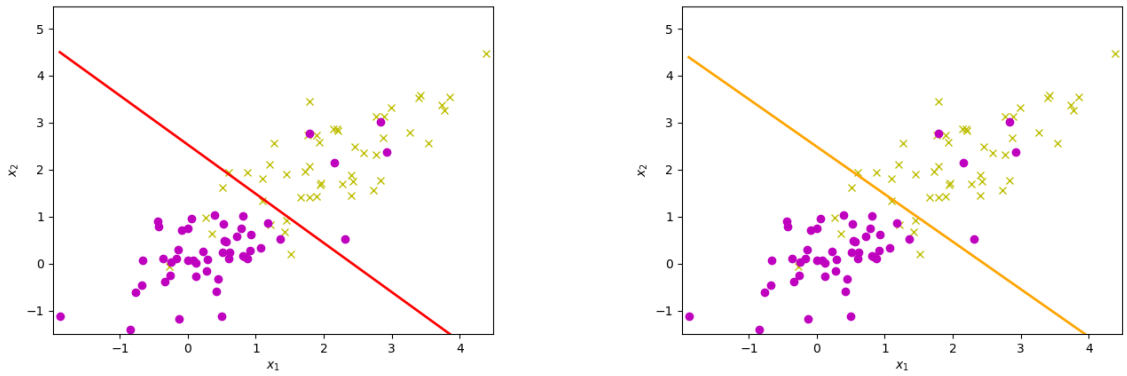
Figure 3: Decision Boundaries for Dataset 1

Because we are plotting two different methods with the same validation dataset, we can notice that the (x_1, x_2) points are the same, but the decision boundary produced by each method is different. Visually, it looks like logistic regression is fitting a decision boundary that is more reasonable than GDA.

(g) **Dataset 2: Compare validation set plots obtained in logistic regression and GDA**

We can now compare the graphs produced by logistic regression and GDA directly and note the difference in prediction.

We know that GDA makes stronger modeling assumptions than logistic regression. In particular, GDA relies on $p(x|y)$ being distributed as multivariate gaussian. We also know that logistic regression is



(a) Logistic Regression

(b) Gaussian Discriminant Analysis

Figure 4: Decision Boundaries for Dataset 2

a more robust method in that it is less sensitive to incorrect modeling assumptions. If the modeling assumptions are correct, we would expect GDA to be marginally better than logistic regression by being asymptotically efficient and requiring less training data to learn adequately.

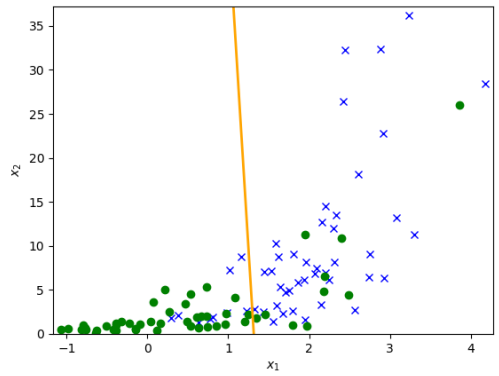
GDA seems to perform worse on Dataset 1; the decision boundaries in Dataset 1 are noticeably different. A possible reason for this behavior is that $p(x|y)$ is not distributed as a multivariate gaussian. For example, if $p(x|y)$ is actually distributed as a Poisson, then logistic regression will still do a good job but GDA will not. In Dataset 1, I would feel more comfortable using Logistic Regression given that its significantly more robust to deviations from modeling assumptions.

(h) **GDA Assumptions**

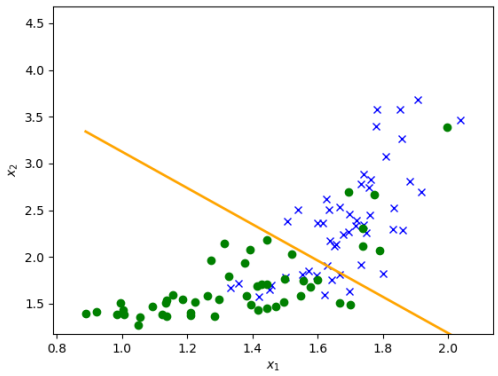
As discussed in (g), it does not look like the GDA modeling assumptions are being satisfied on Dataset 1. It does look like they are satisfied in Dataset 2.

Visually, the plots look like the relationship between x_1 and x_2 in Dataset 1 is exponential. I would try a logarithmic transformation of the $x^{(i)}$'s to neutralize the exponential relationship and have something that resembles a gaussian distribution more closely.

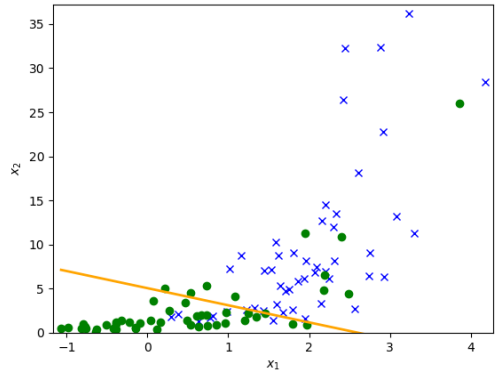
Subplot (a) shows the decision boundary under the original GDA model. Subplot (b) shows the transformed $x^{(i)}$'s, where values have been scaled to avoid negative occurrences in x and a logarithmic function has been applied. We train the model's parameters on this transformed dataset. Subplot (c) shows the decision boundary calculated from the parameters trained on the transformed dataset, and subplot (d) serves as a comparison to the decision boundary produced by the original logistic regression. If we analyze subplots (c) and (d), we can see that the decision boundary produced by GDA with a logarithmic transformation is much closer to the logistic regression's decision boundary. It seems that a logarithmic transformation is a reasonable choice.



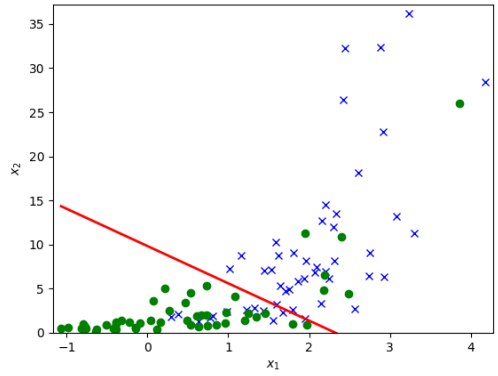
(a) Original GDA Estimation



(b) Logarithmic Transformation Applied



(c) New Decision Boundary w/ Logarithmic Transformation



(d) Original Logistic Regression Estimation

Figure 5: Applying Logarithmic Transformation to Dataset 1

2 Poisson Regression

(a) **Show that Poisson distribution belongs to the Exponential Family**

We can rewrite the Poisson distribution as

$$p(y;\lambda) = \frac{1}{y!} exp(ylog(\lambda) - \lambda)$$

Defining $\eta = log(\lambda)$

$$p(y;\lambda) = \frac{1}{y!} exp(y\eta - e^{\eta})$$

Where

$$b(y) = \frac{1}{y!}$$
$$T(y) = y$$

$$\eta = log(\lambda)$$
$$a(\eta) = e^{\eta}$$

$$p(y;\lambda) = b(y) \cdot exp(\eta T(y) - a(\eta))$$

(b) **Canonical Response Function for this Family**

To obtain the canonical response function, we are interested in calculating the conditional expectation of the distribution. Because we know that a Poisson random variable with parameter λ has mean λ,

$$E[y|x] = \lambda = e^{\eta}$$

If we replace $\eta = \theta^T x$, we obtain

$$E[y|x] = e^{(\theta^T x)}$$

which is the canonical response function.

(c) **Stochastic Gradient Ascent rule for Poisson variable**

The log-likelihood function is given by

$$\ell(y|x;\theta) = log \prod_{i=1}^{n} \left(\frac{1}{y^{(i)}!} exp(y^{(i)} \theta^T x^{(i)} - e^{\theta^T x^{(i)}}) \right)$$

Distributing the logarithm, we obtain

$$\ell(y|x;\theta) = \sum_{i=1}^{n} \left(-log(y^{(i)}!) + y^{(i)} \theta^T x^{(i)} - e^{\theta^T x^{(i)}} \right)$$

Taking the derivative

$$\frac{\partial \ell}{\partial \theta} = \sum_{i=1}^{n} \left(y^{(i)} x^{(i)} - e^{\theta^T x^{(i)}} x^{(i)} \right)$$

The batch gradient ascent rule is given by

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

$$\theta := \theta + \alpha \sum_{i=1}^{n} (y^{(i)} - e^{\theta^T x^{(i)}}) x^{(i)}$$

Because we are interested in Stochastic Gradient Ascent, then we would update the θ vector with each iteration of i:

for $i = 1$ to n,

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

(d) **Coding problem: Website Traffic data**

The following plot shows the relationship between the real and predicted website traffic.

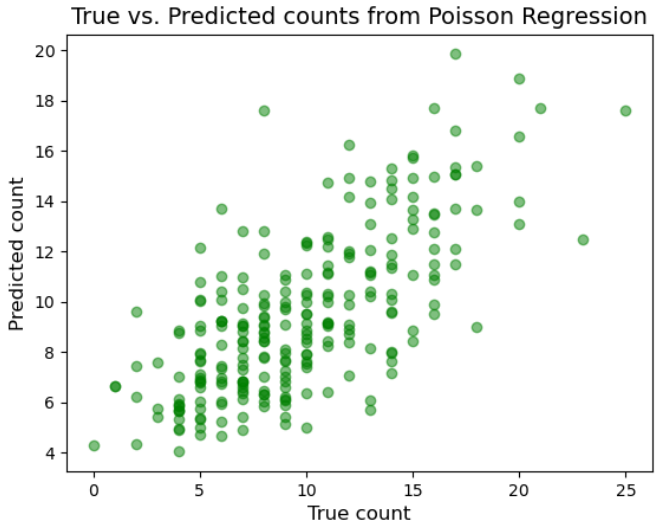


Figure 6: Poisson Regression Results

3 Convexity of Generalized Linear Models

(a) **Mean of an Exponential Family Distribution**

We know that the mean of the distribution is given by

$$\mathbb{E}[Y;\eta] = \int_{-\infty}^{\infty} p(y;\eta) \cdot y \; dy$$

We also know that, by definition, $a(\eta)$ is chosen such that cumulative density function equals 1. Therefore

$$P(Y;\eta) = \int_{-\infty}^{\infty} b(y)exp(\eta T(y) - a(\eta))dy = 1$$

Because we want to take the derivative with respect to $a(\eta)$, lets express the equation in function of $a(\eta)$.

$$\int_{-\infty}^{\infty} b(y)\frac{exp(\eta y)}{exp(a(\eta))}dy = 1$$

After simplifying the algebra, we find

$$a(\eta) = log\left(\int_{-\infty}^{\infty} b(y)exp(\eta y)dy\right)$$

Now we can take the partial derivative of this function with respect to η

$$\frac{\partial a(\eta)}{\partial \eta} = \frac{1}{\int_{-\infty}^{\infty} b(y)exp(\eta y)dy}\int_{-\infty}^{\infty} b(y)exp(\eta y)ydy$$

$$\frac{\partial a(\eta)}{\partial \eta} = \frac{\int_{-\infty}^{\infty} b(y)exp(\eta y)ydy}{exp(a(\eta))}$$

$$\frac{\partial a(\eta)}{\partial \eta} = \int_{-\infty}^{\infty} b(y) exp(\eta T(y) - a(\eta)) y dy \tag{10}$$

$$\frac{\partial a(\eta)}{\partial \eta} = \int_{-\infty}^{\infty} p(y;\eta) y dy$$

$$\frac{\partial a(\eta)}{\partial \eta} = \mathbb{E}[Y;\eta] \tag{11}$$

(b) **Variance of an Exponential Family Distribution**

By definition, the variance of the distribution can be expressed as

$$Var(Y;\eta) = \mathbb{E}[Y^2;\eta] - \mathbb{E}[Y;\eta]^2$$

Taking (10), and calculating the derivative

$$\frac{\partial^2 a(\eta)}{\partial \eta^2} = \int_{-\infty}^{\infty} b(y) y \frac{\partial}{\partial \eta} \left(exp(\eta y - a(\eta))\right) dy$$

$$\frac{\partial^2 a(\eta)}{\partial \eta^2} = \int_{-\infty}^{\infty} b(y) y \cdot exp(\eta y - a(\eta)) \cdot \left(y - \frac{\partial}{\partial \eta} a(\eta)\right) dy$$

Substituting (11) and simplifying

$$\frac{\partial^2 a(\eta)}{\partial \eta^2} = \int_{-\infty}^{\infty} p(y;\eta) y \cdot (y - \mathbb{E}[Y;\eta]) dy$$

$$\frac{\partial^2 a(\eta)}{\partial \eta^2} = \int_{-\infty}^{\infty} p(y;\eta) y^2 dy - \int_{-\infty}^{\infty} p(y;\eta) y \cdot \mathbb{E}[Y;\eta] dy$$

Using the definition of expectation, we obtain

$$\frac{\partial^2 a(\eta)}{\partial \eta^2} = \mathbb{E}[Y^2;\eta] - \mathbb{E}[Y;\eta] \cdot \mathbb{E}[Y;\eta]$$

Therefore, we can prove this is equivalent to the variance

$$\frac{\partial^2 a(\eta)}{\partial \eta^2} = \mathbb{E}[Y^2;\eta] - \mathbb{E}[Y;\eta]^2 = Var(Y;\eta) \tag{12}$$

(c) **Loss Function of an Exponential Family Distribution**

The loss function for the exponential family can be expressed as

$$\ell(\theta) = -log(P(y;\eta(\theta^T x)))$$

$$\ell(\theta) = -log(b(y) exp(\eta y - a(\eta)))$$

$$\ell(\theta) = -log(b(y)) - \theta^T x y + a(\theta^T x)$$

Taking the first derivative

$$\frac{\partial \ell(\theta)}{\partial \theta} = -xy + \frac{\partial a(\theta^T x)}{\partial \theta^T x} \cdot \frac{\partial \theta^T x}{\partial \theta}$$

$$\frac{\partial \ell(\theta)}{\partial \theta} = -xy + \frac{\partial a(\theta^T x)}{\partial \theta^T x} \cdot x$$

Taking the second derivative

$$\frac{\partial \ell(\theta)^2}{\partial^2 \theta} = \frac{\partial}{\partial \theta^T x}\left(\frac{\partial a(\theta^T x)}{\partial \theta^T x}\right) \cdot \frac{\partial \theta^T x}{\partial \theta} \cdot x$$

We can use (12) to simplify

$$\frac{\partial \ell(\theta)^2}{\partial^2 \theta} = H = Var(Y;\eta) \cdot x^2$$

Because we know that the variance of any probability distribution is non-negative, it must be true that

$$H \succeq 0$$

4 Linear Regression: Linear in What?

(a) **Learning degree-3 polynomials of the input**

The objective function $J(\theta)$ is

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(\theta^T \hat{x}^{(i)} - y^{(i)}\right)^2$$

Taking the derivative with respect to θ and substituting in the standard gradient descent rule, we obtain

$$\theta := \theta - \alpha \sum_{i=1}^{n} (\theta^T \hat{x}^{(i)} - y^{(i)})\hat{x}^{(i)}$$

(b) **Coding question: degree-3 polynomial regression**

The degree-3 polynomial regression produces the following plot

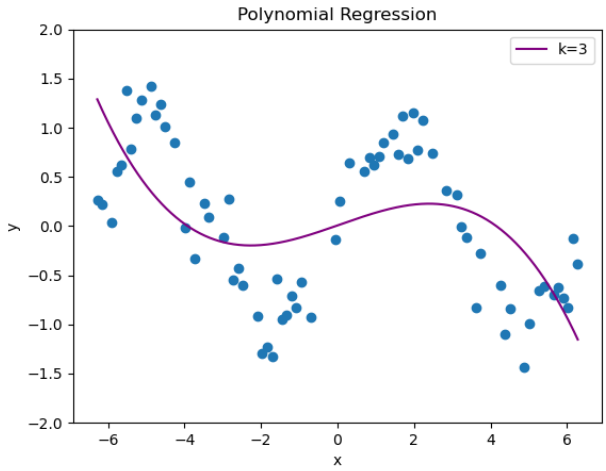


Figure 7: Degree-3 Polynomial Regression

(c) **Coding question: degree-k polynomial regression**

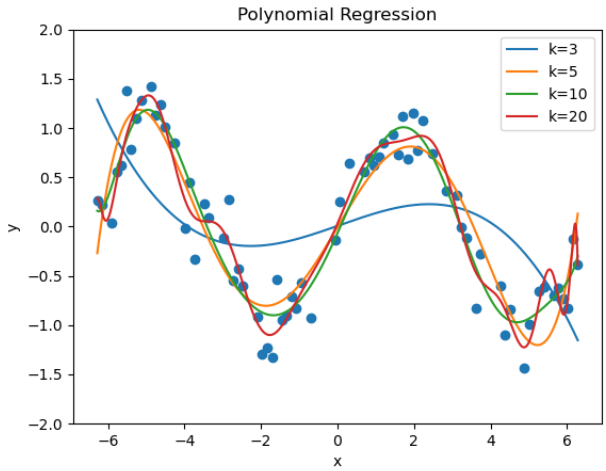


Figure 8: Degree-k Polynomial Regression

The model fits the training data much more closely as we increase k. This makes sense intuitively, as we are calculating more complex functions as k is greater, that can fit training data more closely. However, we run the risk of overfitting to the training data. In particular, we can see that when $k = 20$, the fit jumps around, especially around $x = -3$ and $x = 5$. In this case, $k = 5$ and $k = 10$ seem to be the most reasonable specifications.

(d) **Coding question: other feature maps**

The model including a $sin(x)$ term produces the following fits

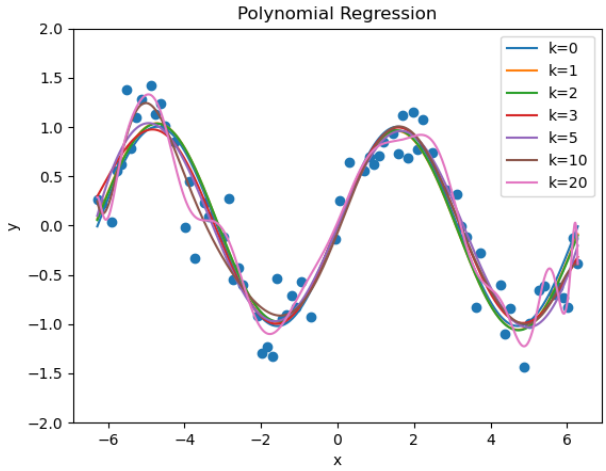


Figure 9: Sine Transformation Degree-k Polynomial Regression

The main difference with the previous model is that, even with a low degree of k, the model produces a good fit. This is a much better feature map, which we knew would be the case given the data generation process $y = sin(x) + \xi$.

(e) **Overfitting with expressive models and small data**

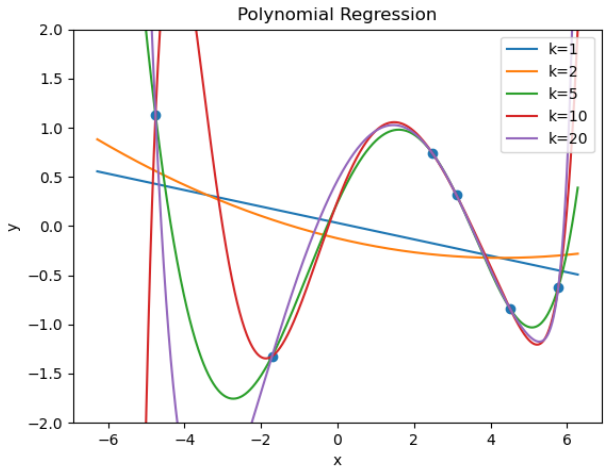


Figure 10: Expressive Models on Small Dataset

As k increases, the model fits the training data perfectly. We have 6 datapoints, and $k+1$ parameters when including the intercept. Therefore, any model with $k \geq 5$ will fit all of the data points perfectly. Clearly, the model is overfitting to the noise produced by the small amount of data that we have, and here one of the lower degree k is likely the better option.

5 Learning Imbalanced Dataset

(a) **Overfitting with expressive models and small data**

(i) A trivial classifier will classify all examples as positive if the majority of the examples are positive, and will classify them all as negative if the majority of examples are negative.

$$\text{Classifier} \begin{cases} \text{All positive if } \rho \geq 0.5 \\ \text{All negative if } \rho < 0.5 \end{cases}$$

Therefore, the classifier will always achieve accuracy of at least

$$A = \frac{max(\#\text{ positive examples}, \#\text{ negative examples})}{\#\text{ examples}}$$

For example, in the spam email example, $\rho = 0.01$. Therefore, we choose to classify all email as not spam. We will predict 99% of the examples correctly.

(ii) To simplify matters, lets rewrite TP + TN + FP + FN as # examples. We know that TP + FN represents the true number of positive examples in the data. Therefore,

$$\rho = \frac{\#\text{ positive examples}}{\#\text{ examples}} = \frac{TP + FN}{TP + TN + FP + FN}$$

Now, we can expand the following expression by substituting the definitions of ρ, A_0 and A_1

$$\rho \cdot A_1 + (1 - \rho)A_0 = \rho \cdot \frac{TP}{TP + FN} + (1 - \rho) \cdot \frac{TN}{TN + FP}$$

$$\rho \cdot A_1 + (1 - \rho)A_0 = \frac{TP + FN}{\#\text{ examples}} \cdot \frac{TP}{TP + FN} + \frac{TN + FP}{\#\text{ examples}} \cdot \frac{TN}{TN + FP}$$

Simplifying, we obtain

$$\rho \cdot A_1 + (1 - \rho)A_0 = \frac{TP}{\#\text{ examples}} + \frac{TN}{\#\text{ examples}} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\rho \cdot A_1 + (1 - \rho)A_0 = A$$

(iii) The accuracy of the majority class will always be 1 and the accuracy of the minority class will always be 0. For example, if $\rho \geq 0.5$, then the trivial classifier will guess all examples as positive, with the following results.

$$TP = \rho$$
$$TN = 0$$
$$FP = 1 - \rho$$
$$FN = 0$$

Calculating accuracy rates,

$$A_1 = \frac{TP}{TP + FN} = \frac{\rho}{\rho + 0} = 100\%$$
$$A_0 = \frac{TN}{TN + FP} = \frac{0}{0 + (1 - \rho)} = 0\%$$

Balanced accuracy is

$$\overline{A} = \frac{1}{2}(A_0 + A_1) = \frac{1}{2} \cdot (1) = 50\%$$

(b) **Coding problem: vanilla logistic regression**

After running the vanilla logistic regression on the imbalanced data set, we find that

$$A = 94.72\%$$
$$A_0 = 98.5\%$$
$$A_1 = 57\%$$
$$\overline{A} = 77.75\%$$

As expected, we can note that the minority class (positive class in our case) has a much lower prediction accuracy. We can also see this visually in the following plot. The red line shows the decision boundary where the model's predicted probability is 0.5.

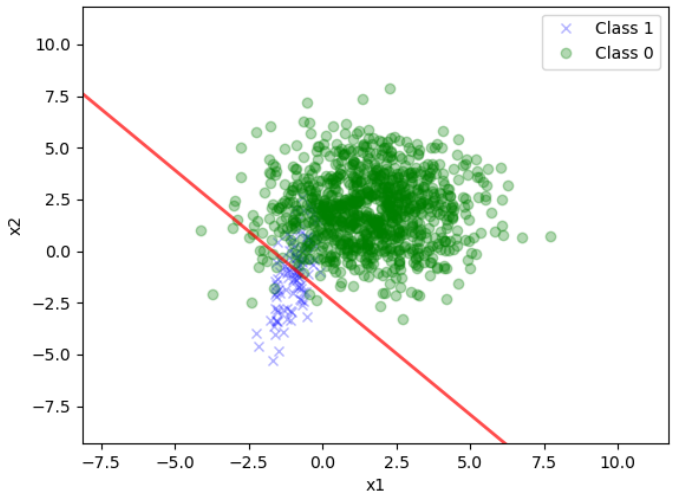


Figure 11: Vanilla Logistic Regression on Imbalanced Data

(c) **Re-sampling/Re-weighting Logistic Regression**

We know that $\mathcal{D}'$ contains each negative example once and each positive example $\frac{\rho}{\kappa}$ times. Then, our new data set contains

$$\mathcal{D}' \text{ contains} \begin{cases} (1-\rho)\cdot n & \text{negative examples} \\ \frac{\rho}{\kappa}\cdot n & \text{positive examples} \end{cases}$$

Substituting $\kappa = \frac{\rho}{1-\rho}$

$$\mathcal{D}' \text{ contains} \begin{cases} (1-\rho)\cdot n & \text{negative examples} \\ (1-\rho)\cdot n & \text{positive examples} \end{cases}$$

It is trivial to see that both classes are balanced in $\mathcal{D}'$ and that there is an equal proportion of positive and negative examples. We can write the accuracy of the model as

$$A(\mathcal{D}') = \left(\frac{1}{2}\right) A_1 + \left(\frac{1}{2}\right) A_0 = \left(\frac{1}{2}\right)(A_0 + A_1) = \overline{A}(\mathcal{D})$$

The empirical loss for logistic regression under the original data is

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

To simplify matters, lets do the following variable change

$$z^{(i)} = y^{(i)} log(h(x^{(i)})) + (1-y^{(i)}) log(1-(h(x^{(i)})))$$

Then, we can express the average empirical loss on dataset $\mathcal{D}$' as

$$J(\theta) = -\frac{1+\kappa}{2n}\sum_{i=1}^{n}\left(w^{(i)} z^{(i)}\right)$$

We can rewrite the sum as the sum of positive and the sum of negative values, substituting the values of $w^{(i)}$. Note the change in range of the sums.

$$J(\theta) = -\frac{1+\kappa}{2n}\left(\sum_{i=1}^{\rho n}\frac{1}{\kappa} z^{(i)} + \sum_{i=\rho n+1}^{n} z^{(i)}\right)$$

Aggregating all terms under a single sum over all examples,

$$J(\theta) = -\frac{1+\kappa}{2n}\sum_{i=1}^{n}\left(\frac{1}{\kappa} z^{(i)}\rho + z^{(i)}(1-\rho)\right)$$

$$J(\theta) = -\frac{1+\kappa}{2n}\left(\frac{\rho}{\kappa} + (1-\rho)\right)\sum_{i=1}^{n} z^{(i)}$$

Substituting $\kappa = \frac{\rho}{1-\rho}$,

$$J(\theta) = -\frac{1}{2n(1-\rho)}\left((1-\rho) + (1-\rho)\right)\sum_{i=1}^{n} z^{(i)}$$

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n} z^{(i)}$$

Substituting back $z^{(i)}$

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

Therefore, we have proven that the average empirical loss for the data set $\mathcal{D}'$ is equal to

$$J(\theta) = -\frac{1+\kappa}{2n}\sum_{i=1}^{n} w^{(i)}\left(y^{(i)}log(h(x^{(i)})) + (1-y^{(i)})log(1-(h(x^{(i)})))\right)$$

(d) **Coding problem: re-weighting minority class**

Implementing the logistic regression on $\mathcal{D}'$, we obtain

$$A = 89.9\%$$
$$A_0 = 89.8\%$$
$$A_1 = 91\%$$
$$\overline{A} = 90.4\%$$

As expected, we can see that the accuracy classifying the majority class fell, but the accuracy classifying the minority class rose significantly. The balanced accuracy is greater in the weighted than in the vanilla logistic regression. We can visualize how the decision boundary has shifted to accommodate the minority class in the following plot.

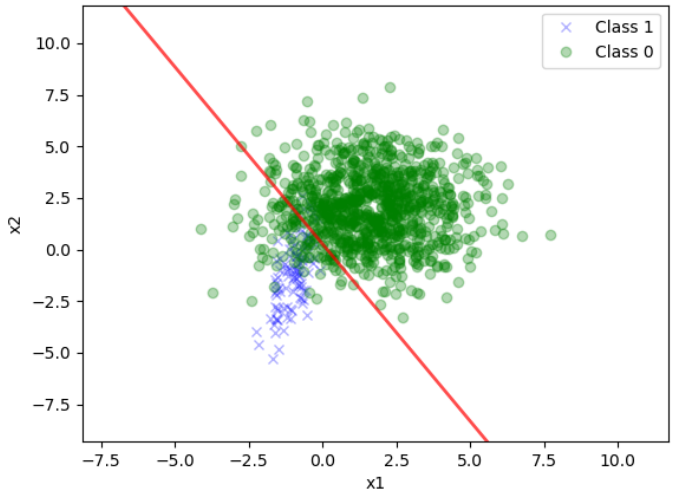


Figure 12: Re-weighted Logistic Regression on Imbalanced Data

6 Convergence and Learning Rate for Gradient Descent

(a) **Quadratic Objective, Scalar Variable**

i. The gradient of the objective function is

$$\nabla J(\theta) = \beta\theta$$

First, lets express $\theta^{[t]}$ in terms of $\theta^{[0]}$:

$$\theta^{[1]} = \theta^{[0]} - \alpha\beta\theta^{[0]}$$
$$\theta^{[2]} = \theta^{[1]} - \alpha\beta\theta^{[1]}$$

Substituting $\theta^{[1]}$

$$\theta^{[2]} = (\theta^{[0]} - \alpha\beta\theta^{[0]}) - \alpha\beta(\theta^{[0]} - \alpha\beta\theta^{[0]})$$
$$\theta^{[2]} = (\theta^{[0]} - \alpha\beta\theta^{[0]})(1 - \alpha\beta)$$
$$\theta^{[2]} = \theta^{[0]}(1 - \alpha\beta)^2$$

Generalizing,

$$\theta^{[t]} = \theta^{[0]}(1 - \alpha\beta)^t$$

Furthermore, we know that the function is convex, and by setting the gradient equal to 0, it is easy to see that the θ that minimizes the function is

$$\theta^* = 0$$

Therefore, we can express the limit as

$$\lim_{t\to\infty} \left|\theta^{[0]}(1 - \alpha\beta)^t - \theta^*\right| = 0$$

We know that

$$\lim_{t\to\infty} \theta^{[0]} \left|(1 - \alpha\beta)^t\right| = 0 \quad \text{if and only if} \quad |(1 - \alpha\beta)| < 1$$

Solving the absolute value,

$$1 > 1 - \alpha\beta > -1$$
$$0 > \alpha\beta > -2$$
$$0 < \alpha\beta < 2$$
$$0 < \alpha < \frac{2}{\beta}$$

Therefore, the range of α is

$$\alpha \in \left(0, \frac{2}{\beta}\right)$$

ii. Given a desired accuracy ϵ, we can substitute for $\theta^{[0]}$ and rewrite the update rule as follows

$$\left|\theta^{[0]}(1-\alpha\beta)^T - \theta^*\right| \leq \epsilon$$

Substituting θ^* and solving for the absolute value,

$$-\epsilon \leq \theta^{[0]}(1-\alpha\beta)^T \leq \epsilon$$

Because we're concerned with the minimum number of iterations only, we can focus on the positive case

$$(1-\alpha\beta)^T \leq \frac{\epsilon}{\theta^{[0]}}$$

For the case in which $(1-\alpha\beta) > 0$:

$$T \cdot log(1-\alpha\beta) \leq log\left(\frac{\epsilon}{\theta^{[0]}}\right)$$

$$T \geq \frac{log\left(\frac{\epsilon}{\theta^{[0]}}\right)}{log(1-\alpha\beta)}$$

The denominator of this equation must be negative in the range of α. The numerator is only positive in the case that $\theta^{[0]} < \epsilon$, which is a trivial case as the condition would be satisfied from the start and $T = 0$. Therefore, we are only interested in the case that $\theta^{[0]} > \epsilon$, which makes the numerator negative. Let us look at the limit of this range to understand how T changes in different values of α.

As $\alpha \to 0$, $log(1-\alpha\beta) \to 0$, which makes $T \to \infty$

As $\alpha \to \frac{1}{\beta}$, $log(1-\alpha\beta) \to -\infty$, which makes $T \to 0$

Therefore, as α grows in the range $\alpha \in \left(0, \frac{1}{\beta}\right)$, T decreases.

For the case in which $(1-\alpha\beta) < 0$:

$$T \cdot log(\alpha\beta - 1) \leq log\left(\frac{\epsilon}{\theta^{[0]}}\right)$$

$$T \geq \frac{log\left(\frac{\epsilon}{\theta^{[0]}}\right)}{log(\alpha\beta-1)}$$

As $\alpha \to \frac{1}{\beta}$, $log(\alpha\beta - 1) \to -\infty$, which makes $T \to 0$

As $\alpha \to \frac{2}{\beta}$, $log(\alpha\beta - 1) \to 0$, which makes $T \to \infty$

In conclusion, T decreases as α approaches $\frac{1}{\beta}$, but increases as α deviates from that value to either direction.

(b) **Quadratic Objective, d-dimensional Variable**

Taking the first derivative of $J(\theta)$:

$$\nabla J(\theta) = \sum_{i=1}^{d} \beta_i \theta_i$$

Now, lets express $\theta^{[t]}$ in terms of $\theta^{[0]}$:

$$\theta^{[1]} = \theta^{[0]} - \alpha \sum_{i=1}^{d} \beta_i \theta_i^{[0]}$$

$$\theta^{[2]} = \theta^{[1]} - \alpha \sum_{i=1}^{d} \beta_i \theta_i^{[1]}$$

Substituting $\theta^{[1]}$

$$\theta^{[2]} = \left(\theta^{[0]} - \alpha \sum_{i=1}^{d} \beta_i \theta_i^{[0]}\right) - \alpha \sum_{i=1}^{d} \beta_i \left(\theta^{[0]} - \alpha \sum_{i=1}^{d} \beta \theta_i^{[0]}\right)$$

$$\theta^{[2]} = \left(\theta^{[0]} - \alpha \sum_{i=1}^{d} \beta_i \theta_i^{[0]}\right) \left(1 - \alpha \sum_{i=1}^{d} \beta_i\right)$$

Generalizing,

$$\theta^{[t]} = \left(\theta^{[0]} - \alpha \sum_{i=1}^{d} \beta_i \theta_i^{[0]}\right) \left(1 - \alpha \sum_{i=1}^{d} \beta_i\right)^{t-1}$$

By setting the derivative of the loss function equal to zero, we know that $\theta_i^* = 0$ for all i. Therefore, we can express the norm as

$$\lim_{t \to \infty} ||\theta^{[t]}||_2 = 0$$

$$\lim_{t \to \infty} \left|\left|\left(\theta^{[0]} - \alpha \sum_{i=1}^{d} \beta_i \theta_i^{[0]}\right) \left(1 - \alpha \sum_{i=1}^{d} \beta_i\right)^{t-1}\right|\right|_2 = 0$$

This will be true if

$$\left|1 - \alpha \sum_{i=1}^{d} \beta_i\right| < 1$$

Solving the equation,

$$1 > 1 - \alpha \sum_{i=1}^{d} \beta_i > -1$$

$$0 > -\alpha \sum_{i=1}^{d} \beta_i > -2$$

$$0 < \alpha \sum_{i=1}^{d} \beta_i < 2$$

$$0 < \alpha < \left(\frac{2}{\sum_{i=1}^{d} \beta_i}\right)$$

When GD converges, $\theta^\dagger = 0$

(c) **Coding question: Quadratic Multivariate Objective**

i. The theoretical derivation from (b) would suggest that $\alpha \in \left(0, \frac{2}{\sum_{i=1}^{d} \beta_i}\right)$.

In this case however, the loss function is not being multiplied by $\frac{1}{2}$, and therefore, $\alpha \in \left(0, \frac{1}{\sum_{i=1}^{d} \beta_i}\right)$ which in this case is $\alpha \in \left(0, \frac{1}{3}\right)$

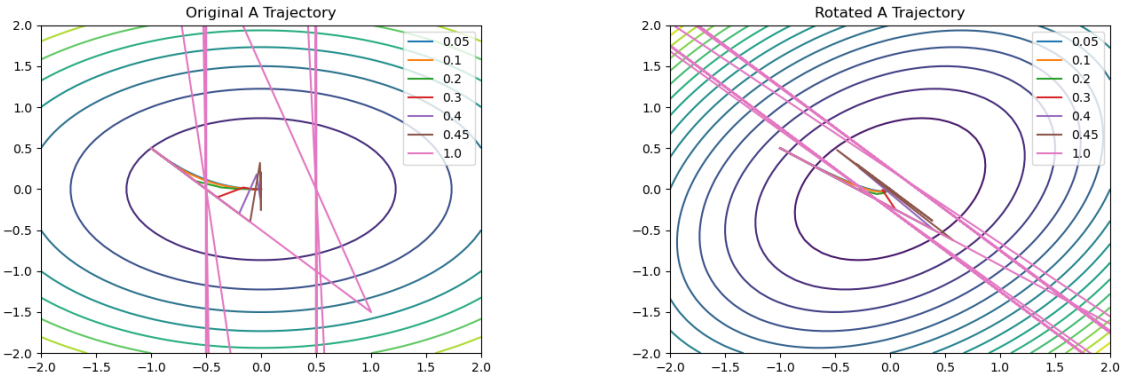
The empirical observations are as follows:

When $\alpha = \frac{1}{3}$, the model converges at $T = 53$
When $\alpha = \frac{1.1}{3}$, the model converges at $T = 75$
When $\alpha = \frac{0.9}{3}$, the model converges at $T = 63$

Therefore, it confirms that $\alpha = \frac{1}{3}$ is the minimum and deviations to either side makes the model converge more slowly. As $T \to \infty$, a learning rate of $\alpha > \frac{1}{3}$ would cause the model to diverge.

When $\alpha \to 0$, for example with $\alpha = 10^{-2}$, the model converges at $T = 2,770$
In a more extreme example, $\alpha = 10^{-5}$ makes the model converges at $T = 2,625,040$
Therefore, the theoretical derivation of the learning rate does indeed match empirical observations.

ii. The following plots show the trajectories of the GD algorithm under the original specification and under a rotated A matrix.



(a) Trajectory under A

(b) Trajectory under rotated A

Figure 13: Gradient Descent Trajectory

It is easy to observe that all specifications converge except when $\alpha = 1$, which is a learning rate that is too high, and jumps over the minimum repeatedly. We can also observe that the shortest path is taken by $\alpha = 0.3$, from the start of the algorithm to reaching the minimum. When comparing the number of iterations that the GD algorithm takes to converge between the original matrix A and rotated A, we obtain the following results.

As we can see, the gradient descent algorithm takes roughly the same amount of iterations under both the rotated and original matrices.

α	$T_{Original}$	$T_{Rotated}$
0.05	539	531
0.10	256	254
0.20	113	113
0.30	63	64
0.40	112	116
0.45	255	260
1.00	NC	NC

Table 1: Convergence speed under different learning rates

In conjunction, the evidence from the plots and table suggest that the gradient descent algorithm is rotation invariant.

(d) **Convergence of Gradient Descent for a General Convex Objective**

Using Taylor's expansion, we know that the following must be true

$$J(\theta^{[t]}) = J(\theta^{[t-1]}) + \nabla J(\theta^{[t-1]})^T(\theta^{[t]} - \theta^{[t-1]}) + \frac{1}{2}(\theta^{[t]} - \theta^{[t-1]})^T \nabla^2_\theta J(\theta^{[t-1]} + c(\theta^{[t]} - \theta^{[t-1]}))(\theta^{[t]} - \theta^{[t-1]})$$

We can perform the following substitutions

$$H = \nabla^2_\theta J(\theta^{[t-1]} + c(\theta^{[t]} - \theta^{[t-1]}))$$
$$z = \theta^{[t]} - \theta^{[t-1]}$$

Using the spectral theorem, we can rewrite H as a product of its orthonormal and diagonal matrices, where the diagonal matrix has the eigenvalues of H in its diagonals.

$$H = U\Lambda U^T$$

We can then diagonalize the quadratic form, and using $\hat{z} = z^T U$, we can write the quadratic form as

$$z^T H z = z^T U \Lambda U^T z = \hat{z}^T \Lambda \hat{z} = \sum_{i=1}^{n} \lambda_i {\hat{z}_i}^2$$

We know that the largest eigenvalue of the Hessian matrix is less than β_{max} for all points θ and therefore,

$$max(\lambda_i) \leq \beta_{max}$$

Therefore, it must be true that

$$\sum_{i=1}^{n} \lambda_i {\hat{z}_i}^2 \leq \sum_{i=1}^{n} \beta_{max} {\hat{z}_i}^2 = \beta_{max} \sum_{i=1}^{n} {\hat{z}_i}^2$$

$$z^T H z \leq \beta_{max} z^T z$$

Substituting this result back in the Taylor Expansion, it now becomes an inequality

$$J(\theta^{[t]}) \leq J(\theta^{[t-1]}) + \nabla J(\theta^{[t-1]})^T(\theta^{[t]} - \theta^{[t-1]}) + \frac{1}{2}\beta_{max}(z^T z)$$

$$J(\theta^{[t]}) \leq J(\theta^{[t-1]}) + \nabla J(\theta^{[t-1]})^T(\theta^{[t]} - \theta^{[t-1]}) + \frac{1}{2}\beta_{max}((\theta^{[t]} - \theta^{[t-1]})^T(\theta^{[t]} - \theta^{[t-1]})) \tag{13}$$

Using the definition of gradient descent

$$\theta^{[t]} = \theta^{[t-1]} - \alpha\nabla J(\theta^{[t-1]}) \tag{14}$$

Substituting (14) in (13):

$$J(\theta^{[t]}) \leq J(\theta^{[t-1]}) + \nabla J(\theta^{[t-1]})^T(\theta^{[t-1]} - \alpha\nabla J(\theta^{[t-1]}) - \theta^{[t-1]}) + \frac{1}{2}\beta_{max}((\theta^{[t-1]} - \alpha\nabla J(\theta^{[t-1]}) - \theta^{[t-1]})^T(\theta^{[t-1]} - \alpha\nabla J(\theta^{[t-1]}) - \theta^{[t-1]})) \tag{15}$$

Grouping terms and simplifying,

$$J(\theta^{[t]}) \leq J(\theta^{[t-1]}) - \alpha\nabla J(\theta^{[t-1]})^T\nabla J(\theta^{[t-1]}) + \frac{1}{2}\beta_{max}\alpha^2 J(\theta^{[t-1]})^T\nabla J(\theta^{[t-1]})$$

$$J(\theta^{[t]}) \leq J(\theta^{[t-1]}) - \alpha\nabla J(\theta^{[t-1]})^T\nabla J(\theta^{[t-1]})\left(1 - \frac{\alpha\beta_{max}}{2}\right) \tag{16}$$

To show convergence, we need to prove that

$$J(\theta^{[t]}) \leq J(\theta^{[t-1]})$$

Let a, b and c be real numbers, where $c > 0$

$$a \leq b - c$$

Because $c > 0$, it must be true that

$$a < b$$

Therefore, to prove convergence we must show that the last term in (16) is strictly positive

$$\alpha\nabla J(\theta^{[t-1]})^T\nabla J(\theta^{[t-1]})\left(1 - \frac{\alpha\beta_{max}}{2}\right) > 0$$

We know $J(\theta^{[t-1]})^T\nabla J(\theta^{[t-1]}) \geq 0$ because it is a quadratic. For the term to be positive, it must hold that

$$\alpha\left(1 - \frac{\alpha\beta_{max}}{2}\right) > 0$$

$$\alpha \cdot (2 - \alpha\beta_{max}) > 0$$

For this to hold true,

$$0 < \alpha < \frac{2}{\beta_{max}}$$

(e) **Learning Rate for Linear Regression**

We can rewrite the objective function as

$$J(\theta) = \frac{1}{2}(X\theta - y)^T(X\theta - y)$$

$$J(\theta) = \frac{1}{2}\left[(X\theta)^T(X\theta) - (X\theta)^T y - y^T X\theta + y^T y\right]$$

$$\nabla J(\theta) = \frac{1}{2}(2X^T X\theta - 2X^T y)$$

$$\nabla J(\theta) = X^T X\theta - X^T y$$

Taking the second derivative,

$$H = \nabla^2 J(\theta) = X^T X$$

Given that $J(\theta)$ is convex and the largest eigenvalue of $X^T X$ is β_{max}, the conditions of (6d) are met. This proves that for an $\alpha \in \left(0, \frac{1}{\beta_{max}}\right)$, it must be true that $J(\theta^{[t]})^T$ converges at $t \to \infty$