CS229 Fall 2017, Problem Set #1: Supervised Learning

Armand Sumo – armandsumo@gmail.com

June 14, 2021

Collaborators:

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

(The code used to generate the graphs can be found in the file *assignment-1.py*.)

1. Logistic regression

Average empirical loss for logistic regression:

$$J(\theta) = -\tfrac{1}{m}\sum_{i=1}^{m} log(h_\theta(y^i x^i))$$

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

(a)

$$\begin{aligned}
\nabla_\theta J(\theta) &= -\frac{1}{m}\sum_{i=1}^{m}\frac{1}{g(\theta^T y^{(i)} x^{(i)})}\nabla_\theta g(\theta^T y^{(i)} x^{(i)}) \\
&= -\frac{1}{m}\sum_{i=1}^{m}\frac{1}{g(\theta^T y^{(i)} x^{(i)})} y^{(i)} x^{(i)} g(\theta^T y^{(i)} x^{(i)})(1 - g(\theta^T y^{(i)} x^{(i)})) \\
&= -\frac{1}{m}\sum_{i=1}^{m}\frac{1}{y}^{(i)} x^{(i)}(1 - g(\theta^T y^{(i)} x^{(i)}))
\end{aligned}$$

$$\begin{aligned}
H_{i,j} = \frac{\partial}{\partial\theta_j}\left[\nabla_\theta J(\theta)\right]_i &= \frac{1}{m}\sum_{i=1}^{m}(y^{(i)})^2 x_j^{(i)} x_i^{(i)} g(\theta^T y^{(i)} x^{(i)})(1 - g(\theta^T y^{(i)} x^{(i)})) \\
&= \frac{\partial}{\partial\theta_j}\left[\nabla_\theta J(\theta)\right]_i && \text{H is symmetric}
\end{aligned}$$

Let's show that for any vector z,
$z^T H z \geq 0$

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

$$\begin{aligned}
z^T H z &= \sum_i z_i^T (Hz)_i = \sum_i \sum_j z_i (H_{i,j}) z_j \\
&= \sum_i \sum_j z_i (\frac{1}{m} \sum_{k=1}^{m} (y^{(k)})^2 x_j^{(k)} x_i^{(k)} g(\theta^T y^{(k)} x^{(k)})(1 - g(\theta^T y^{(k)} x^{(k)}))) z_j \\
&= \frac{1}{m} \sum_{k=1}^{m} \sum_i \sum_j (y^{(k)})^2 z_j x_j^{(k)} z_i x_i^{(k)} g(\theta^T y^{(k)} x^{(k)})(1 - g(\theta^T y^{(k)} x^{(k)})) \\
&= \frac{1}{m} \sum_{k=1}^{m} (y^{(k)})^2 g(\theta^T y^{(k)} x^{(k)})(1 - g(\theta^T y^{(k)} x^{(k)}))((x^{(k)})^T z)^2
\end{aligned}$$

For any vector z, $g(z) \in [0, 1]$, hence $z^T H z \geq 0$.
This implies that H is positive semi-definite, therefore J is convex and has no local minima other than the global one.

(b) After implementing Newton's method for optimizing $J(\theta)$ and applying it to fit a logistic regression model to the data, I obtained a parameter vector:
$\theta = [-2.61847133, 0.75979248, 1.1707512]^T$.

(c)

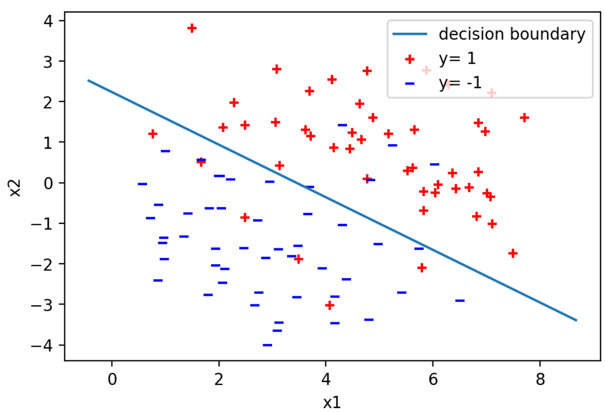


Figure 1: Training data and decision boundary fit by logistic regression

2. Poisson regression and the exponential family

(a) We consider the Poisson distribution parametrized by λ:

$$p(y;\lambda) = \frac{e^{-y}\lambda^y}{y!} = \frac{\exp(y\log(\lambda) - \lambda)}{y!} = b(y)(\exp(\eta^T T(y) - a(\eta))$$

The Poisson distribution is in the exponential family, with:

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

(b) We want to perform regression using a GLM model with a Poisson response variable. To construct the GLM model, we make the following assumptions: - $y|x;\theta \sim \text{ExponentialFamily}(\eta)$
- our goal is to predict the expected value of $T(y)$ given x. Because T(y)=y, this means we would like the hypothesis $h_\theta(x)$ to satisfy: $h_\theta(x) = \mathbb{E}[y|x]$
- The natural parameter η and the inputs x are related linearly $y = \theta^T x$ It follows that our hypothesis will output:

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

Therefore, the canonical response of this family is $g(z) = h(\theta^T z) = e^z$.

(c) Our model assumes that the conditional probability of y given x is:
$p(y^{(i)}|x^{(i)};\theta) = \dfrac{\exp(y^{(i)}\theta^T x^{(i)} - e^{\theta^T x^{(i)}})}{y^{(i)}!}$
We now maximize the likelihood $L(\theta)$ of our parameter θ using gradient ascent.

$$\begin{aligned} \ell(\theta) &= \log(L(\theta)) = \log(p(y^{(i)}|x^{(i)};\theta)) = y^{(i)}\theta^T x^{(i)} - e^{\theta^T x^{(i)}} - \log(y^{(i)}!) \\ \frac{\partial \ell(\theta)}{\partial \theta_j} &= y^{(i)}x_j^{(i)} e^{\theta^T x^{(i)}} = x_j^{(i)}(y^{(i)} - e^{\theta^T x^{(i)}}) \end{aligned}$$

We obtain the following stochastic gradient ascent update rule:

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

with $h_\theta(x) = e^{\theta^T x}$

(d) We now use GLM for any member of the exponential family for which $T(y) = y$, and the canonical response $h(x)$ for the family. From our model's assumptions,

$$p(y|X;\theta) = b(y)(\exp(\eta^T T(y) - a(\eta)) = b(y)(\exp(\eta^T y - a(\eta))$$
$$\ell(\theta) = \log p(y|X;\theta) = \eta^T y - a(\eta) + log(b(y))$$

For a single parameter θ_i,

$$\frac{\partial \ell(\theta)}{\partial \theta_i} = \frac{\partial}{\partial \theta_i}(\theta^T x)^T y - \frac{\partial}{\partial \theta_i} a(\theta^T x)$$

To determine $a(\eta)$, we use the fact that for $p(y|X;\theta)$ to be a pdf, it must integrate to 1.

$$\int_y p(y|X;\theta)\, dy = 1$$
$$\int_y b(y)(\exp(\eta^T T(y) - a(\eta))\, dy = 1$$
$$e^{a(\eta)} = \int_y b(y) \exp(\eta^T y))\, dy$$
$$a(\eta) = log \int_y b(y) \exp(\eta^T y))\, dy$$

Let f be a differentiable function such that $a(\eta) = \log f(\eta)$. Using the chain rule, $\frac{\partial a(\eta)}{\partial \eta} = \frac{\partial \log f(\eta)}{\partial \eta} = \frac{\partial f(\eta)}{\partial \eta}\frac{1}{f(\eta)}$. Hence,

$$\begin{aligned}
\frac{\partial a(\eta)}{\partial \eta} &= \frac{1}{\int_y b(y) \exp(\eta^T y))\, dy} \int_y b(y) \frac{\partial \exp(\eta^T y)}{\partial \eta}\, dy \\
&= \frac{1}{\int_y b(y) \exp(\eta^T y)\, dy} \int_y b(y) \exp(\eta^T y) \frac{\partial \eta^T y}{\partial \eta}\, dy \\
\frac{\partial a(\theta^T x)}{\partial \theta_i} &= \frac{1}{\int_y b(y) \exp(\eta^T y))\, dy} \int_y b(y) \exp(\eta^T y) \frac{\partial x^T \theta y}{\partial \theta_i}\, dy \\
&= \frac{1}{\int_y b(y) \exp(\eta^T y))\, dy} \int_y b(y) \exp(\eta^T y) x_i y\, dy \\
&= \int_y \frac{b(y) \exp(\eta^T y)\, dy}{\int_y b(y) \exp(\eta^T y)\, dy} x_i y\, dy \\
&= \int_y b(y) \frac{\exp(\eta^T y)}{\exp(a(\eta))} x_i y\, dy \\
&= \int_y p(y|X;\theta) x_i\, dy \\
&= xi \int_y p(y|X;\theta)\, dy = x_i \mathbb{E}\left[y|x;\theta\right] = x_i h_\theta(x)
\end{aligned}$$

It follows that:

$$\begin{aligned}\frac{\partial \ell(\theta)}{\partial \theta_i} &= \frac{\partial}{\partial \theta_i}(\theta^T x)^T y - \frac{\partial}{\partial \theta_i} a(\theta^T x) \\ &= x_i y - x_i h_\theta(x) = x_i(y - h_\theta(x))\end{aligned}$$

Therefore, the stochastic gradient ascent on the log likelihood of $p(y|X;\theta)$ results in the update rule:

$$\boxed{\theta_i := \theta_i - \alpha(h_\theta(x) - y)x_i}$$

3. Gaussian discriminant analysis

(a) Suppose we are given a dataset $\{(x^{(i)}, y^{(i)}); i = 1, \ldots, m\}$, consisting of m independent examples where $x^{(i)} \in \mathbb{R}^n$ and $y^{(i)} \in \{-1, 1\}$. We model the joint distribution of (x, y) according to:

$$\begin{aligned} p(y) &= \begin{cases} \phi & y = 1 \\ 1 - \phi & y = -1 \end{cases} = \phi^{1\{y=1\}}(1-\phi)^{1\{y=-1\}} \\ p(x|y=-1) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_{-1})^T \Sigma^{-1}(x-\mu_{-1})\right) \\ p(x|y=1) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_{1})^T \Sigma^{-1}(x-\mu_{1})\right) \end{aligned}$$

(There are two mean vectors μ_1, μ_{-1} but only one covariance matrix Σ.)
Suppose we already fit ϕ, Σ, μ_1 and μ_{-1} and want to make a prediction at some new query point x. The posterior distribution of the label x takes the form:

$$\begin{aligned} p(y=1|x;\phi,\Sigma,\mu_1,\mu_{-1}) &= \frac{p(x|y=1;\phi,\Sigma,\mu_1,\mu_{-1})p(y=1)}{p(x,\phi,\Sigma,\mu_1,\mu_{-1})} \\ &= \frac{p(x|y=1;\phi,\Sigma,\mu_1,\mu_{-1})p(y=1)}{p(x|y=1)p(y=1)+p(x|y=-1)p(y=-1)} \\ &= \frac{1}{1+\dfrac{p(x|y=-1)p(y=-1)}{p(x|y=1)p(y=1)}} \\ &= \frac{1}{1+\dfrac{\exp\left(-\frac{1}{2}(x-\mu_{-1})^T\Sigma^{-1}(x-\mu_{-1})\right)(1-\phi)}{\exp\left(-\frac{1}{2}(x-\mu_{1})^T\Sigma^{-1}(x-\mu_{1})\right)\phi}} \end{aligned}$$

Note that because $x|y = 1$ and $x|y = -1$ share the same covariance matrix Σ, the terms in $\dfrac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}$ cancel one another.

$$p(y=1|x;\phi,\Sigma,\mu_1,\mu_{-1}) =$$
$$\frac{1}{1+\exp\left(\log(\frac{\phi}{1-\phi}) - \frac{1}{2}(x-\mu_{-1})^T\Sigma^{-1}(x-\mu_{-1}) + \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)}$$
$$p(y=-1|x;\phi,\Sigma,\mu_1,\mu_{-1}) =$$
$$\frac{1}{1+\exp\left(\log(\frac{1-\phi}{\phi}) - \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) + \frac{1}{2}(x-\mu_{-1})^T\Sigma^{-1}(x-\mu_{-1})\right)}$$

More generally,

$$p(y|x;\phi,\Sigma,\mu_1,\mu_{-1}) =$$
$$\frac{1}{1+\exp\left[y\left(\log(\frac{1-\phi}{\phi})\ \underbrace{-\frac{1}{2}(x-\mu_{-1})^T\Sigma^{-1}(x-\mu_{-1}) + \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)}_{(1)}\right]}$$

Let $j = 1$ or -1.

$$(x-\mu_j)^T\Sigma^j(x-\mu_1) = (x^T\Sigma^{-1}x - 2x^T\Sigma^{-1}u_j + u_j^T\Sigma^{-1}u_j)$$

(Σ^{-1} is symmetric therefore $x^T\Sigma^{-1}u_j = u_j^T\Sigma^{-1}x$)

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

Hence,

$$p(y=1|x;\phi,\Sigma,\mu_1,\mu_{-1}) =$$
$$\frac{1}{1+\exp\left(-y(\log(\frac{\phi}{1-\phi}) + \frac{1}{2}(-\Sigma^{-1}\mu_{-1}^T\mu_{-1} + \Sigma^{-1}\mu_1^T\mu_1) + (\Sigma^{-1}\mu_1 - \Sigma^{-1}\mu_{-1})^Tx\right)}$$

$$\boxed{p(y=1|x;\phi,\Sigma,\mu_1,\mu_{-1}) = \frac{1}{1+\exp(-y(\theta_0+\theta^Tx))}}$$

with

$\theta_0 = \log(\dfrac{\phi}{1-\phi}) + \dfrac{1}{2}(-\Sigma^{-1}\mu_{-1}^T\mu_{-1} + \Sigma^{-1}\mu_1^T\mu_1)$ and $\theta = \Sigma^{-1}\mu_1 - \Sigma^{-1}\mu_{-1}$

(b) (proved in (c))

(c) The log likelihood of the data is:

$$\ell(\phi,\Sigma,\mu_1,\mu_{-1}) = \log\prod_{i=1}^{m} p(x^{(i)},y^{(i)},\phi,\Sigma,\mu_1,\mu_{-1})$$
$$= \log\prod_{i=1}^{m} p(x^{(i)}|y^{(i)},\Sigma,\mu_1,\mu_{-1})p(yi,\phi)$$
$$= \sum_{i=1}^{m}\log(p(yi,\phi)) + \sum_{i=1}^{m}\log(p(x^{(i)}|y^{(i)},\Sigma,\mu_1,\mu_{-1}))$$

$\ell(\phi,\Sigma,\mu_1,\mu_{-1}) = \sum_{i=1}^{m}\log(\phi^{1\{y^{(i)}=1\}}) + \log((1-\phi)^{1\{y^{(i)}=-1\}})$
$+\sum_{i=1}^{m}\log\left(\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\right)+\left(-\frac{1}{2}(x^{(i)}-\mu_{y^{(i)}})^T\Sigma^{-1}(x^{(i)}-\mu_{y^{(i)}})\right)$
$=\sum_{i=1}^{m} 1\{y^{(i)}=1\}\log(\phi)+1\{y^{(i)}=-1\}log(1-\phi)$
$-m\log((2\pi)^{n/2}|\Sigma|^{1/2})-\sum_{i=1}^{m}\frac{1}{2}(x^{(i)}-\mu_{y^{(i)}})^T\Sigma^{-1}(x^{(i)}-\mu_{y^{(i)}})$
In order to find the estimator of each of the parameters Σ,μ_1, $\mu{-1}$ and ϕ, we compute the gradient of the log likelihood with respect to each parameter:

$$\nabla_\Sigma\ell(\phi,\Sigma,\mu_1,\mu_{-1}) = -\nabla_\Sigma m\log((2\pi)^{n/2}|\Sigma|^{1/2}) - \nabla_\Sigma\frac{1}{2}(x^{(i)}-\mu_{y^{(i)}})^T\Sigma^{-1}(x^{(i)}-\mu_{y^{(i)}})$$
$$\nabla_\Sigma m\log((2\pi)^{n/2}|\Sigma|^{1/2}) = -\frac{m}{2}\nabla_\Sigma(|\Sigma|)) = \frac{\partial\log(|\Sigma|)}{\partial|\Sigma|}\nabla_\Sigma|\Sigma| = \frac{1}{|\Sigma|}(\Sigma^{-T}|\Sigma|) = \Sigma^{-T}$$

Since

$$\frac{\partial}{\partial\Sigma_{k,l}}|\Sigma| = \frac{\partial}{\partial\Sigma_{k,l}}\sum_{i=1} n(-1)^{i+j}\Sigma_{i,j}|\Sigma_{\backslash i\backslash j}| = (-1)^{k+l}|\Sigma_{k\backslash k,\backslash l}| = (adj(\Sigma))_{l,k}$$

and

$$\nabla_\Sigma|\Sigma| = adj(\Sigma)^T = (|\Sigma|\Sigma^{-1})^T = \Sigma^{-T}|\Sigma|)$$

For a non-singular matrix X, $\dfrac{\partial a^T X^{-1} b}{\partial X} = -X^{-T}ab^TX^{-T}$
In our case,

$$\nabla_\Sigma\sum_{i=1}^{m}\frac{1}{2}(x^{(i)}-\mu_{y^{(i)}})^T\Sigma^{-1}(x^{(i)}-\mu_{y^{(i)}}) = -\frac{1}{2}\sum_{i=1}^{m}\Sigma^{-1}(x^{(i)}-\mu_{y^{(i)}})(x^{(i)}-\mu_{y^{(i)}})^T\Sigma^{-1}$$

and

$$\nabla_\Sigma \ell(\phi, \Sigma, \mu_1, \mu_{-1}) = -\frac{m}{2}\Sigma^{-1} + \frac{1}{2}\sum_{i=1}^{m} \Sigma^{-1}(x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T \Sigma^{-1}$$

At an extremum, the gradient is equal to zero,

$$0 = -m\Sigma^{-1} + \sum_{i=1}^{m} \Sigma^{-1}(x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T \Sigma^{-1}$$

We obtain an estimator of the parameter Σ:

$$\boxed{\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T}$$

$$\begin{aligned}
\frac{\partial}{\partial \phi}\ell(\phi, \Sigma, \mu_1, \mu_{-1}) &= \frac{1}{\phi}\sum_{i=1}^{m} 1\{y^{(i)} = 1\} - \frac{1}{1-\phi}\sum_{i=1}^{m} 1\{y^{(i)} = -1\} \\
&= \sum_{i=1}^{m} \frac{1\{y^{(i)} = 1\}}{\phi} - \frac{1\{y^{(i)} = -1\}}{\phi}
\end{aligned}$$

by setting it to the 0 vector,

$$\begin{aligned}
0 &= \sum_{i=1}^{m} \frac{(1-\phi)1\{y^{(i)} = 1\} - \phi 1\{y^{(i)} = -1\}}{\phi(1-\phi)} \\
&= \sum_{i=1}^{m} 1\{y^{(i)} = 1\} - \phi 1\{y^{(i)} = 1\} - \phi 1\{y^{(i)} = -1\} \\
&= \sum_{i=1}^{m} 1\{y^{(i)} = 1\} - \phi \underbrace{\left(1\{y^{(i)} = 1\} + 1\{y^{(i)} = -1\}\right)}_{=1} = \sum_{i=1}^{m} 1\{y^{(i)} = 1\} - m\phi
\end{aligned}$$

We obtain the estimator of the parameter ϕ

$$\boxed{\phi = \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}$$

$$\nabla_{\mu_1}\ell(\phi, \Sigma, \mu_1, \mu_{-1}) = -\frac{1}{2}\sum_{i=1}^{m} \nabla_{\mu_1} 1\{y^{(i)} = 1\}(x^{(i)} - \mu_1)^T \Sigma^{-1}(x^{(i)} - \mu_1)$$

$$= -\frac{1}{2}\sum_{i=1}^{m} 1\{y^{(i)} = 1\}\nabla_{(x^{(i)} - \mu_1)}(x^{(i)} - \mu_1)^T \Sigma^{-1}(x^{(i)} - \mu_1) \cdot \nabla_{\mu_1}(x^{(i)} - \mu_1)$$

Σ^{-1} is symmetric therefore $\nabla_{(x^{(i)}-\mu_1)}(x^{(i)}-\mu_1)^T\Sigma^{-1}(x^{(i)}-\mu_1) = 2\,\Sigma^{-1}(x^{(i)}-\mu_1)$

$$\nabla_{\mu_1}\ell(\phi,\Sigma,\mu_1,\mu_{-1}) = \sum_{i=1}^{m} 1\{y^{(i)}=1\}\Sigma^{-1}(x^{(i)}-\mu_1)$$

at an extremum the gradient is equal to the zero vector,

$$0 = \sum_{i=1}^{m} 1\{y^{(i)}=1\}\Sigma^{-1}(x^{(i)}-\mu_1)$$

by pre-multiplying both sides by Σ

$$0 = \sum_{i=1}^{m} 1\{y^{(i)}=1\}(x^{(i)}-\mu_1)$$

We obtain the estimator of the parameter μ_1:

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

conversely an estimator of the parameter μ_{-1},

$$\boxed{\mu_{-1} = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=-1\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)}=-1\}}}$$

4. Linear invariance of optimization algorithms

We consider using some iterative optimization algorithm (such as Newton's method, or gradient descent) to minimize some continuously differentiable function $f(x)$ that can be defined as

$$\begin{aligned} f : \mathbb{R}^n &\mapsto \mathbb{R}^m \\ x = (x_1,\ldots,x_n)^T &\rightarrow (f_1(x_1,\ldots,x_n)^T, f_2(x_1,\ldots,x_n)^T,\ldots,f_m(x_1,\ldots,x_n)^T)^T \end{aligned}$$

where the f_i-s are continuously differentiable real-valued functions. Let $A \in \mathbb{R}^{n\times n}$ be some non-singular matrix and let's define a function g, by $g(z) = f(Az)$. Consider we use the same iterative optimization algorithm to optimize g, (with initialization $z^{(0)} = \vec{0}$). The optimization algorithm is said to be invariable to linear reparameterizations if the values $z^{(1)}, z^{(2)}, \ldots$ satisfy $z^{(i)} = A^{-1}x^{(i)}$ for all i.

(a) We'll show by induction that this is true for the Newton optimization algorithm. In order to avoid tensor notation, we will restrict ourselves to a real valued (multivariable) function f, which is equivalent to studying the optimization algorithm for each

component f_i of $f(x)$.
The second order approximation of f near $x^{(i)}$ is the quadratic function of $x^{(i)}$ defined by

$$f(x) = f(x^{(i)}) + \nabla f(x^{(i)})^T(x - x^{(i)}) + \frac{1}{2!}(x - x^{(i)})^T Hf(x^{(i)})(x - x^{(i)})$$

Where $\nabla f(x^{(i)})$ and $Hf(x^{(i)})$ denote respectively the Gradient and Hessian f with respect to x ,evaluated at a point $x^{(i)}$. We now take the gradient of both sides with respect to x:
$f(x^{(i)})$ is a constant so its gradient is $\vec{0}$

$$\nabla_x(\nabla f(x^{(i)})^T(x - x^{(i)})) = \nabla_x f(x^{(i)})$$

because f is continuously differentiable, its Hessian matrix is symmetric.Then,

$$\nabla_x\left(\frac{1}{2}(x - x^{(i)})^T Hf(x^{(i)})(x - x^{(i)})\right) = Hf(x^{(i)})$$

At an extremum, $\nabla_x(f(x)) = 0$, the update rule follows:

$$x^{(i+1)} = x^{(i)} - (Hf(x^{(i)}))^{-1}\nabla_x f(x^{(i)})$$

because g is also continously differentiable, we get the update rule:

$$z^{(i+1)} = z^{(i)} - (Hg(z^{(i)}))^{-1}\nabla_x g(z^{(i)})$$

<u>Base case:</u> $z^{(0)} = \vec{0} = A^{-1}x^{(0)}$
<u>Induction step:</u> we suppose that for a certain non-zero integer i, the following is true:

$$(H_i) : z^{(i)} = A^{-1}x^{(i)}$$

Before going any further we must first prove the following equalities:
$\nabla g(z) = A^T\nabla f(Az)$ and $Hg(z) = A^T Hf(Az)A$

$$[\nabla g(z)]_i = \frac{\partial g(z)}{\partial z_i} = \frac{\partial f(Az)}{\partial z_i} = \nabla f(Az) \cdot \frac{\partial f(Az)}{\partial z_i} \qquad = \nabla f(Az)A_{\cdot,i}$$

By convention, the gradient is a column vector so:

$$[\nabla g(z)] = A^T\nabla f(Az)$$

Let $h(z) = \nabla g(z) = A^T\nabla f(Az)$ The Hessian of g at z is

$$h'(z) = A^T\nabla^2 f(Az)A$$

Where ' denotes the derivative operator(transpose of the gradient).
We can now begin the induction step:

$$\begin{aligned}
z^{(i+1)} &= z^{(i)} - Hg(z^{(i)})^{-1}\nabla_x g(z^{(i)}) \\
Az^{(i+1)} &= Az^{(i)} - A(Hg(z^{(i)}))^{-1}\nabla_x g(z^{(i)}) \\
&= x^{(i)} - A(A^T\nabla^2 f(Az)A)^{-1}A^T\nabla f(x^{(i)}) \qquad (H_i) \\
&= x^{(i)} - A(A^{-1}Hf^{-1}(x^{(i)})A^{-T})A^T\nabla f(x^{(i)}) \\
&= x^{(i)} - Hf^{-1}(x^{(i)})\nabla f(x^{(i)}) = x^{(i+1)}
\end{aligned}$$

Hence,

$$\boxed{z^{(i+1)} = A^{-1}x^{(i+1)}}$$

Because it is true for an arbitrary non-zero integer i, we can conclude that the Newton update is invariant to linear transformation.

(b) Following the same reasoning as in (a),
the gradient update of x can be expressed as, with $\alpha \in \mathbb{R}$:

$$x^{(i+1)} = x^{(i)} - \alpha\nabla_f(x^{(i)})$$

On z,

$$\begin{aligned}
z^{(i+1)} &= z^{(i)} - \alpha\nabla_g(z^{(i)}) = z^{(i)} - \alpha\nabla_f(Az^{(i)}) = z^{(i)} - \alpha A^T\nabla_f(x^{(i)}) \\
Az^{(i+1)} &= x^{(i)} - \alpha AA^T\nabla_f(x^{(i)}) \neq x^{(i)} - \alpha\nabla_f(x^{(i)}) = x^{(i+1)}
\end{aligned}$$

(Assuming A is not the identity matrix.)
This shows that the gradient descent optimization algorithm is not invariant to linear transformation.

5. Regression for denoising quasar spectra

(a) Locally weighted linear regression
We want to minimize

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

where $w^{(i)}$ is the weight for a training example (i).
Let X be the m-by-$d+1$ design matrix that contains the training examples' input values in its rows and y be an m-dimensional vector containing all the target values from the training set: $X = \begin{bmatrix} — & (x^{(1)})^T & — \\ — & (x^{(2)})^T & — \\ & \vdots & \\ — & (x^{(m)})^T & — \end{bmatrix}$; $y = \begin{bmatrix} — & y^{(1)} & — \\ — & y^{(2)} & — \\ & \vdots & \\ — & y^{(m)} & — \end{bmatrix}$

(i)

$$
\begin{aligned}
(X\theta - y)_j &= (x^{(j)})^T\theta - y^{(j)} \\
[W(X\theta - y)]_i &= W_i(X\theta - y) = \sum_{i=1}^{m} W_{i,j}(x^{(j)})^T\theta - y^{(j)} \\
(X\theta - y)_i^T &= (x^{(i)})^T\theta - y^{(i)}
\end{aligned}
$$

$$
\begin{aligned}
(X\theta - y)^T W (X\theta - y) &= \sum_{i=1}^{m} (X\theta - y)_i^T \left[W(X\theta - y)\right]_i \\
&= \sum_{i=1}^{m} \big((x^{(i)})^T\theta - y^{(i)}\big)\left(\sum_{i=1}^{m} W_{i,j}(x^{(j)})^T\theta - y^{(j)}\right)
\end{aligned}
$$

Let's define the matrix $W \in \mathbb{R}^{m\times m}$ such that:

$$
W_{i,j} = \begin{cases} \frac{w^{(i)}}{2} & i = j \\ 0 & i \neq j \end{cases}
$$

Then,

$$
\begin{aligned}
(X\theta - y)^T W (X\theta - y) &= \sum_{i=1}^{m} \big((x^{(i)})^T\theta - y^{(i)}\big)\big(\frac{w^{(i)}}{2}((x^{(i)})^T\theta - y^{(i)})\big) \\
&= \frac{1}{2}\sum_{i=1}^{m} w^{(i)}\big((x^{(i)})^T\theta - y^{(i)}\big)^2 \\
&= J(\theta)
\end{aligned}
$$

(ii)

$$
\begin{aligned}
J(\theta) &= \theta^T \underbrace{(X^T W X)}_{\text{symmetric}} \theta - \theta^T X^T W y - y^T W X \theta + y^T \theta y \\
\nabla_\theta J(\theta) &= 2X^T W X\theta - 2X^T W y
\end{aligned}
$$

At an extremum, $\nabla_\theta J(\theta) = 0$;

$$
0 = 2X^T W X\theta - 2X^T W y
$$

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

(iii) Suppose we have a training set $\{(x^{(i)}, y^{(i)});\ i = 1, \cdots, m\}$of m independent examples in which $y^{(i)}$ are observed in different variances. Specifically, suppose that:

$$
p(y^{(i)}|x^{(i)}, \theta) = \frac{1}{\sqrt{2\pi}\sigma^{(i)}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2(\sigma^{(i)})^2}\right)
$$

The log likelihood of the parameter θ is:

$$\ell(\theta) = \log \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}, \theta) = \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma^{(i)}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2(\sigma^{(i)})^2}\right)$$

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

Therefore, maximizing $\ell(\theta)$ gives the same result as minimizing

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

Which reduces to solving a weighted linear regression problem with weights:

$$w^{(i)} = \frac{1}{(\sigma^{(i)})^2}$$

(b) Visualizing the data

(i) I used the normal equations to implement unweighted linear regression ($y = \theta^T x$) on the first training example. I obtained the optimal parameter vector:

$$\theta = [2.51339906e + 00, -9.81122145e - 04]^T$$

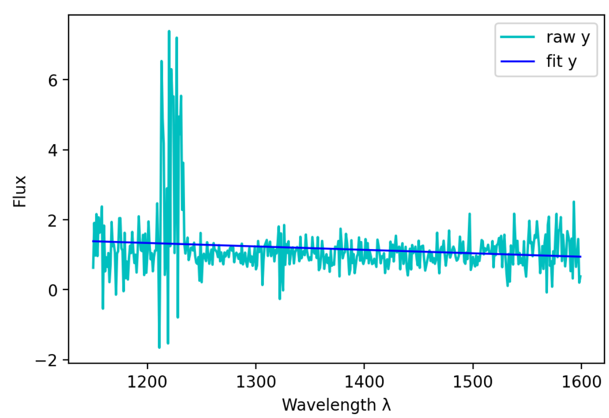
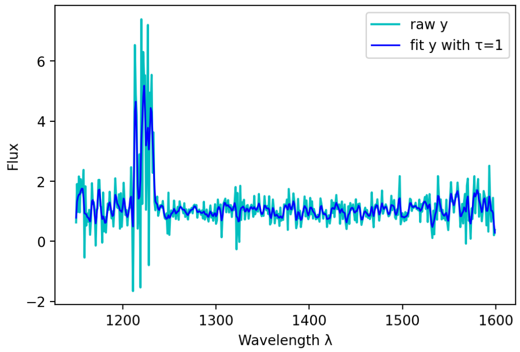


Figure 2: Raw data and straight line resulting from fit

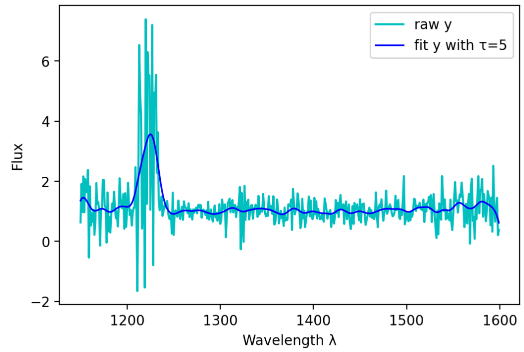
(ii) included in (iii)

(iii) I implemented weighted linear regression on the first training example. When evaluating $h(\cdot)$ at a query point x, I used the weights:

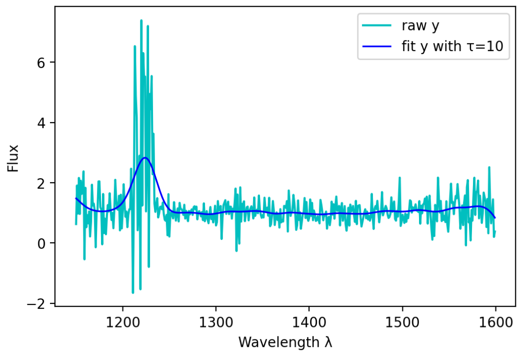
$$w^{(i)} = \exp\left(-\frac{(x - x^{(i)})^2}{2\tau^2}\right) \qquad \text{where } \tau \text{ is the bandwith parameter}$$



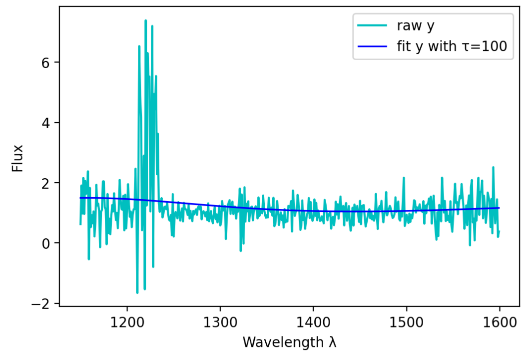
(a) $\tau = 1$



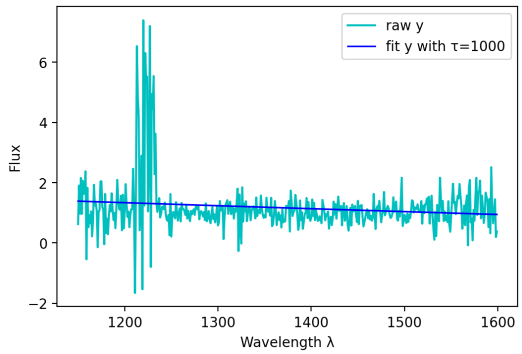
(b) $\tau = 5$



(c) $\tau = 10$



(d) $\tau = 100$



(e) $\tau = 1000$

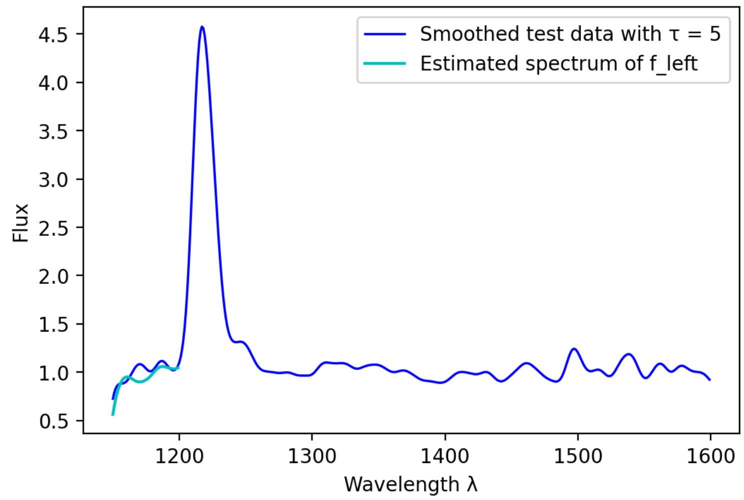
Figure 3: Raw data and curve fit by weighted linear regression for different values of τ. The smaller the bandwidth parameter, the tighter the fit of the curve on the raw data.

(c) Predicting quasar spectra through functional regression

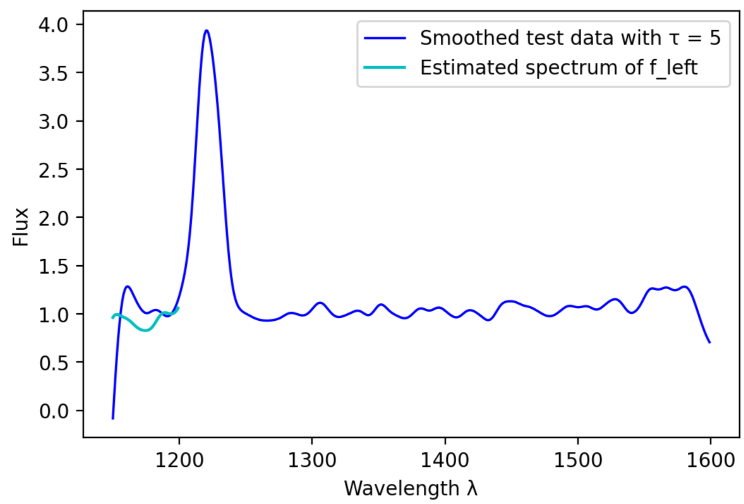
(i) (see *assignment-1.py*)

(ii) I performed weighted regression on the *locally weighted regressions* to construct estimators of the left spectra for all training examples.
The average error over the training data is: 0.28.

(iii) I performed the same operations as in (ii) over the test examples.
The average error over the test data is: 0.10.



(a) Test Example 1



(b) Test Example 6

Figure 4: Smoothed curve obtained through locally weighted regression and estimated curve of f_{left} through functional regression.