CS 229, Fall 2017 Problem Set #1: Supervised Learning

Rocío Ventura Abreu

19-10-2017

Disclaimer

These problem sets have been developed by the CS229 team at Stanford University. If you are following this course or a related one, please consider whether checking my solutions might constitute a violation of the honour code.

1 Logistic regression

a. Consider the average empirical loss (the risk) for logistic regression:

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

where $y^{(i)} \in \{-1, 1\}$, $h_\theta = g(\theta^T x)$ and $g(z) = 1/(1+e^{-x})$. Find the Hessian H of this function, and show that for any vector z_1 holds true that

$$z_1^T H z_1 \geq 0$$

<u>Solution</u>

Hessian:

$$\frac{\delta J}{\delta \theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left[\left(\frac{1}{1+e^{-y^{(i)}\theta^T x^{(i)}}}\right)\left(e^{-y^{(i)}\theta^T x^{(i)}}\right)\left(-y^{(i)}x_j^{(i)}\right)\right]$$

$$
\begin{aligned}
H_{jk} &= \frac{\delta J}{\delta \theta_j \theta_k} \\
&= \frac{1}{m}\sum_{i=1}^{m}\left[\left(\frac{-1}{\left(1+e^{-y^{(i)}\theta^T x^{(i)}}\right)^2}\right)\left(e^{-y^{(i)}\theta^T x^{(i)}}\right)\left(-y^{(i)}x_k^{(i)}\right)\left(e^{-y^{(i)}\theta^T x^{(i)}}\right)\left(-y^{(i)}x_j^{(i)}\right)\right. \\
&\left. + \left(\frac{1}{1+e^{-y^{(i)}\theta^T x^{(i)}}}\right)\left(e^{-y^{(i)}\theta^T x^{(i)}}\right)\left(-y^{(i)}x_k^{(i)}\right)\left(e^{-y^{(i)}\theta^T x^{(i)}}\right)\left(-y^{(i)}x_j^{(i)}\right)\right] \\
&= \frac{1}{m}\sum_{i=1}^{m}\left[\underbrace{\frac{e^{-y^{(i)}\theta^T x^{(i)}}}{1+e^{-y^{(i)}\theta^T x^{(i)}}}}_{\in(0,1)}\left(y^{(i)}\right)^2 x_j^{(i)}x_k^{(i)}\left(1-\underbrace{\frac{e^{-y^{(i)}\theta^T x^{(i)}}}{1+e^{-y^{(i)}\theta^T x^{(i)}}}}_{\in(0,1)}\right)\right] \\
&= \frac{1}{m}\sum_{i=1}^{m}\left[C_i x_j^{(i)}x_k^{(i)}\right] \text{ where } C_i \geq 0
\end{aligned}
$$

$$
\begin{aligned}
z_{1j}^T H_{jk} z_{1k} &= \sum_{j=1}^{n+1} z_j \left(\sum_{k=1}^{n+1}\left(\sum_{i=1}^{m}\left[\frac{C_i}{m}x_j^{(i)}x_k^{(i)}\right] z_k\right)\right) \\
&= \sum_{i=1}^{m}\frac{C_i}{m}\left(\sum_{j=1}^{n+1}\sum_{k=1}^{n+1} z_j x_j^{(i)}x_k^{(i)} z_k\right) = \sum_{i=1}^{m}\frac{C_i}{m}\left({x^{(i)}}^T z\right)^2 \geq 0
\end{aligned}
$$

b. We have provided two data files:

- `http://cs229.stanford.edu/ps/ps1/logistic_x.txt`
- `http://cs229.stanford.edu/ps/ps1/logistic_y.txt`

These files contain the inputs ($x^{(i)} \in \mathbb{R}^2$) and outputs ($y^{(i)} \in \{-1, 1\}$), respectively for a binary classification problem, with one training example per row. Implement Newton's method for optimizing $J(\theta)$, and apply it to fit a logistic regression model to the data. Initialize Newton's method with $\theta = \vec{0}$ (the vector of all zeros). What are the coefficients θ resulting from your fit?

Solution

$\theta = [-2.62, \quad 0.76, \quad 1.17]$

See script `PSET1_1b.py` *in the scripts folder for the full code.*

c. Plot the training data (your axes should be x_1 and x_2, corresponding to the two coordinates of the inputs, and you should use a different symbol for each point plotted to indicate whether that example had label 1 or -1). Also plot on the same figure the decision boundary fit by logistic regression. (This should be a straight line showing the boundary separating the region where $h_\theta(x) > 0.5$ from where $h_\theta(x) \leq 0.5$.)

Solution

The boundary line is defined by $h_\theta = 0.5$. Since h_θ is the logistic regression, that is equivalent to enforce $\theta^T x = 0$.

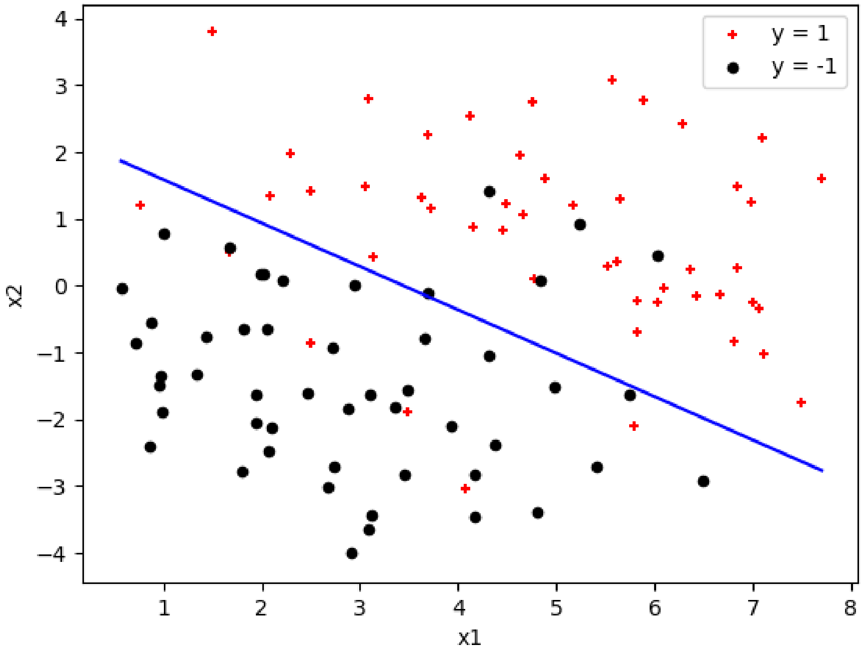


Figure 1: Training set and hypothesis $h_\theta(x) = 0.5$

2 Poisson regression and the exponential family

a. Consider the Poisson distribution parameterized by λ:

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

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

Solution

For the generic exponential family,

$$p(y; \eta) = b(y) \exp(\eta^T T(y) - a(\eta)) \tag{1}$$

$$\log p(y; \eta) = \log(b(y)) + \eta^T T(y) - a(\eta) \tag{2}$$

For the Poisson distribution,

$$p(y; \lambda) = \frac{e^{-\lambda}\lambda^y}{y!} \tag{3}$$

$$\log p(y; \lambda) = \log \frac{1}{y!} + y \log \lambda - \lambda \tag{4}$$

And so from (2) and (4), the Poisson distribution is part of the exponential family, where:

$$\eta = \log \lambda \qquad T(y) = y \qquad a(\eta) = \lambda \qquad b(y) = \frac{1}{y!}$$

b. 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.)

Solution

Canonical response function $= E[T(y); \eta] = E[y; \eta(\lambda)] \overset{y \sim Poisson}{=} \lambda = \exp(\eta)$.

c. For a training set $\{(x^{(i)}, y^{(i)}); i = 1, ..., m\}$, 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 rule for learning using a GLM model with Poisson responses y and the canonical response function.

Solution

As a precondition for applying GLM, $\eta^{(i)} = \theta^T x^{(i)}$.

$$\begin{aligned} l^{(i)}(\theta) = \log p^i &\overset{(4)}{=} \log \frac{1}{y^{(i)}!} + y^{(i)}\eta^i - \exp(\eta^i) \\ &= \log \frac{1}{y^{(i)}!} + y^{(i)}\theta^T x^{(i)} - \exp(\theta^T x^{(i)}) \\ l(\theta) &= \sum_{i=1}^{m} l^{(i)}(\theta) \end{aligned}$$

Since we are asked for stochastic gradient descent, we will only consider the gradient $\frac{\delta l^{(i)}(\theta)}{\delta \theta_j}$.

$$\frac{\delta l^{(i)}(\theta)}{\delta \theta_j} = y^{(i)}x_j^{(i)} - y^{(i)}x_j^{(i)} \exp(y^{(i)}\theta^T x^{(i)})$$

Stochastic gradient ascent rule:

Loop until convergence {
    for $i = 1, ..., m$ {
        $\theta_j = \theta_j + \alpha \frac{\delta l^{(i)}(\theta)}{\delta \theta_j} \quad \forall j$
    }
}

d. Consider using GLM with a response variable from any member of the exponential family in which $T(y) = y$, and the canonical response function $h(x)$ for the family. Show that stochastic gradient ascent on the log-likelihood log $p(y|X; \theta)$ results in the update rule $\theta_i := \theta_i - \alpha(h(x) - y)x_i$.

Solution

What needs to be proved is that $\frac{\delta \log p^i}{\delta \theta_j} = (y^{(i)} - h(x^{(i)}))x_j^{(i)} \quad \forall i$. From the definition of a GLM (see (2)),

$$\begin{aligned}
\log p^i &= \log(b(y^{(i)}) + {\eta^i}^T T(y^{(i)}) - a(\eta^i) \\
&= \log(b(y^{(i)})) + (\theta^T x^{(i)})^T y^{(i)} - a(\theta^T x^{(i)}) \\
\frac{\delta \log p^i}{\delta \theta_j} &= y^{(i)} x_j^{(i)} - \frac{\delta a(\eta^i)}{\delta(\eta^i)} x_j^{(i)}
\end{aligned}$$

And so what we finally should prove is that $h(\theta^T x^{(i)}) = h(\eta^i) = \frac{\delta a(\eta^i)}{\delta(\eta^i)}$.

By the definition of a probability distribution,

$$\begin{aligned}
1 &= \int_{-\infty}^{+\infty} b(y^{(i)}) \exp({\eta^i}^T y^{(i)}) \exp(-a(\eta^i)) \mathrm{d}y \quad \forall i \\
\exp(-a(\eta^i)) &= \frac{1}{\int_{-\infty}^{+\infty} b(y^{(i)}) \exp({\eta^i}^T y^{(i)}) \mathrm{d}y} \\
\frac{\delta \exp(-a(\eta^i))}{\delta \eta^i} &= \frac{-\int_{-\infty}^{+\infty} b(y^{(i)}) y^{(i)} \exp\left({\eta^i}^T \left(y^{(i)} - 1\right)\right) \exp\left(\eta^i\right) \mathrm{d}y}{\left(\int_{-\infty}^{+\infty} b(y^{(i)}) \exp({\eta^i}^T y^{(i)}) \mathrm{d}y\right)^2} \\
-\frac{\delta a}{\delta \eta^i} \exp(-a(\eta^i)) &= -\exp(a(\eta^i))^2 \int_{-\infty}^{+\infty} y \cdot p(y|x) \mathrm{d}y \\
&= -\exp(a(\eta^i)) E[y^{(i)}; \eta^i] = -\exp(a(\eta^i)) h(\eta^i)
\end{aligned}$$

And by simplifying on both sides of the equality,

$$\frac{\delta a(\eta^i)}{\delta(\eta^i)} = h(\eta^i)$$

3 Gaussian discriminant analysis

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

$$
\begin{aligned}
p(y) &= \begin{cases} \phi & y = 1 \\ \phi & y = -1 \end{cases} \\
p(x|y=1) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right) \\
p(x|y=-1) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_{-1})^T\Sigma^{-1}(x-\mu_{-1})\right)
\end{aligned}
$$

Here, the parameters of our model are ϕ, Σ, μ_1 and μ_{-1}.

a. Suppose we have already fit ϕ, Σ, μ_1 and μ_{-1}, and now want to make a prediction at some new query point x. Show that the posterior distribution of the label at x takes the form of a logistic function, and can be written

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

where $\theta \in \mathbb{R}^n$ and the bias term $\theta_0 \in (R)$ are some appropriate functions of ϕ, Σ, μ_1 and μ_{-1}.

<u>Solution</u>

For easiness in the development, let A and E_i be

$$
\begin{aligned}
A &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \\
E_i &= \exp\left(-\frac{1}{2}(x-\mu_i)^T\Sigma^{-1}(x-\mu_i)\right) \quad \text{with} \quad i \in \{-1,1\}
\end{aligned}
$$

$$
\begin{aligned}
p(y|x;params) &= \frac{p(x|y)p(y)}{p(x)} \\
p(x) = p(x|y &= 1)p(y=1) + p(x|y=-1)p(y=-1)
\end{aligned}
$$

Therefore,

$$
p(y|x;params) = \begin{cases} \dfrac{E_1\phi}{E_1\phi + E_{-1}(1-\phi)} & y = 1 \\ \dfrac{E_{-1}(1-\phi)}{E_1\phi + E_{-1}(1-\phi)} & y = -1 \end{cases}
$$

Expressing $p(y|x;params)$ in a compact way,

$$
\begin{aligned}
p(y|x) &= \frac{1}{1+B(x)^{-y}} \quad \text{where} \\
B(x) &= \frac{E_1\phi}{E_{-1}(1-\phi)}
\end{aligned}
$$

We can now elaborate on $B(x)$:

$$B(x) = \exp\left(\underbrace{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) + \frac{1}{2}(x-\mu_{-1})^T\Sigma^{-1}(x-\mu_{-1})}_{C(x)}\right)$$
$$\cdot \exp\left(\underbrace{\log\left(\frac{\phi}{1-\phi}\right)}_{D}\right)$$

$C(x)$ seems like a quadratic function of x, but the second order terms cancel out, and what is left is an affine function of x of the type $C(x) = C_1 x + X_0$.

Renaming, $\exp(C_1 x + C_0 + D) = \exp(\theta_1 x + \theta_0)$, and hence:

$$p(y|x) = \frac{1}{1 + \exp(\theta_1 x + \theta_0)^{-y}}$$

b. For this part of the problem only, you may assume n (the dimension of x) is 1, so that $\Sigma = [\sigma^2]$ is just a real number, and likewise the determinant of Σ is given by $|\Sigma| = \sigma^2$. The log-likelihood of the data is

$$\begin{aligned} l(\phi, \Sigma, \mu_1, \mu_{-1}) &= \log \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \phi, \Sigma, \mu_1, \mu_{-1}) \\ &= \sum_{i=1}^{m} \log p(x^{(i)}|y^{(i)}; \Sigma, \mu_1, \mu_{-1}) + \sum_{i=1}^{m} \log p(y^{(i)}; \phi) \end{aligned}$$

By maximising l with respect to the parameters, prove that the maximum likelihood estimates of ϕ, Σ, μ_1 and μ_{-1} are indeed as given in the formulas.

Conditional Gaussian distribution: $p(x|y=k) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu_k)^2}{2\sigma^2}\right)$ with $k = \{-1, 1\}$

Prior Bernoulli distribution: $p(y) = \begin{cases} \phi & y = 1 \\ 1-\phi & y = -1 \end{cases}$

Derivatives:

(a) Derivative w.r.t. ϕ:

$$
\begin{aligned}
\frac{\delta l}{\delta \phi} &= \sum_{y^{(i)}=1} \frac{1}{\phi} \cdot 1 + \sum_{y^{(i)}=-1} \frac{1}{(1-\phi)} \cdot (-1) \\
&= \frac{1}{\phi} \sum_{i=1}^{m} \mathbb{1}\left\{y^{(i)} = 1\right\} - \frac{1}{1-\phi} \sum_{i=1}^{m} \mathbb{1}\left\{y^{(i)} = -1\right\} = 0 \\
\implies & (1-\phi) \sum_{i=1}^{m} \mathbb{1}\left\{y^{(i)} = -1\right\} - \phi \sum_{i=1}^{m} \mathbb{1}\left\{y^{(i)} = 1\right\} = 0 \\
\implies & - m\phi + \sum_{i=1}^{m} \mathbb{1}\left\{y^{(i)} = 1\right\} = 0 \\
\implies & \phi = \frac{\sum_{i=1}^{m} \mathbb{1}\left\{y^{(i)} = 1\right\}}{m}
\end{aligned}
$$

(b) Derivative w.r.t. μ_{-1}:

$$
\begin{aligned}
\frac{\delta l}{\delta \mu_{-1}} &= \sum_{y^{(i)}=-1} \frac{1}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x^{(i)} - \mu_{-1})^2\right)} \cdot \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x^{(i)} - \mu_{-1})^2}{2\sigma^2}\right) \frac{(x^{(i)} - \mu_{-1})}{\sigma^2} = 0 \\
\implies & \frac{1}{\sigma^2} \left(\sum_{i=1}^{m} \mathbb{1}\left\{y^{(i)} = -1\right\} x^{(i)} - \mu_{-1} \sum_{i=1}^{m} \mathbb{1}\left\{y^{(i)} = -1\right\} \right) = 0 \\
\implies & \mu_{-1} = \frac{\sum_{i=1}^{m} \mathbb{1}\left\{y^{(i)} = -1\right\} x^{(i)}}{\sum_{i=1}^{m} \mathbb{1}\left\{y^{(i)} = -1\right\}}
\end{aligned}
$$

(c) Derivative w.r.t. μ_1:
Proceeding in the same way as for μ_{-1},

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

(d) Derivative w.r.t. σ^2:

$$\frac{\delta l}{\delta \sigma^2} = \sum_{y^{(i)}=-1} \left[\frac{1}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x^{(i)}-\mu_{-1})^2\right)} \cdot \left(\frac{\pi}{\left(\sqrt{2\pi\sigma^2}\right)^3} \cdot \exp\left(-\frac{(x^{(i)}-\mu_{-1})^2}{2\sigma^2}\right) + \right.\right.$$

$$\left.\left. + \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \frac{+(x^{(i)}-\mu_{-1})^2}{2(\sigma^2)^2} \exp\left(-\frac{(x^{(i)}-\mu_{-1})^2}{2\sigma^2}\right)\right)\right] +$$

$$\sum_{y^{(i)}=1} \left[\frac{1}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x^{(i)}-\mu_{1})^2\right)} \cdot \left(\frac{\pi}{\left(\sqrt{2\pi\sigma^2}\right)^3} \cdot \exp\left(-\frac{(x^{(i)}-\mu_{1})^2}{2\sigma^2}\right) + \right.\right.$$

$$\left.\left. + \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \frac{+(x^{(i)}-\mu_{1})^2}{2(\sigma^2)^2} \exp\left(-\frac{(x^{(i)}-\mu_{1})^2}{2\sigma^2}\right)\right)\right]$$

$$= \sum_{y^{(i)}=-1} \left[\frac{\pi}{2\pi\sigma^2} + \frac{(x^{(i)}-\mu_{-1})^2}{2(\sigma^2)^2}\right] + \sum_{y^{(i)}=1} \left[\frac{\pi}{2\pi\sigma^2} + \frac{(x^{(i)}-\mu_{1})^2}{2(\sigma^2)^2}\right]$$

$$= \frac{m}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} \sum_{i=1}^{m} (x^{(i)} - \mu_{y^{(i)}})^2 = 0$$

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

4 Linear invariance of optimisation algorithms

Consider using an iterative optimization algorithm (such as Newton's method, or gradient descent) to minimize some continuously differentiable function $f(x)$. Suppose we initialize the algorithm at $x^{(0)} = 0$. When the algorithm is run, it will produce a value of $x \in \mathbb{R}^n$ for each iteration: $x^{(1)}, x^{(2)}, \ldots$.

Now, let some non-singular square matrix $A \in \mathbb{R}^{nxn}$ be given, and define a new function $g(z) = f(A \cdot z)$. Consider using the same iterative optimization algorithm to optimize g (with initialization $z^{(0)} = 0$). If the values $z^{(1)}, z^{(2)}, \ldots$ produced by this method necessarily satisfy $z^{(i)} = A^{-1}x^{(i)} \quad \forall i$, we say this optimization algorithm is invariant to linear reparametrizations.

a. Show that Newton's method (applied to find the minimum of a function) is invariant to linear reparameterizations.

Solution

Newton's method: $z^{(i+1)} = z^{(i)} - \left(\nabla_z g(z^{(i)})\right)^{-1} \cdot g(z^{(i)})$. Multiplying both sides by A,

$$A \cdot z^{(i+1)} = \underbrace{A \cdot z^{(i)}}_{\mathrm{I}} - \underbrace{A \cdot \left(\nabla_z g(z^{(i)})\right)^{-1}}_{\mathrm{III}} \cdot \underbrace{g(z^{(i)})}_{\mathrm{II}}$$

$$\implies x^{(i+1)} = x^{(i)} - \left(\nabla_x f(x^{(i)})\right)^{-1} \cdot f(x^{(i)})$$

where

I. $A \cdot z^{(i)} = x^{(i)}$

II. $g(z^{(i)}) = f(A \cdot z^{(i)}) = f(x^{(i)})$

III. $A \cdot \left(\nabla_z g(z^{(i)})\right)^{-1} = \left(\nabla_z g(z^{(i)}) \cdot A^{-1}\right)^{-1} = \left(\nabla_z f(A \cdot z^{(i)}) \cdot A^{-1}\right)^{-1} = \left(\nabla_{A \cdot z} f(A \cdot z^{(i)})\right)^{-1} = \left(\nabla_x f(x^{(i)})\right)^{-1}$

b. Is gradient descent invariant to linear reparameterizations? Justify your answer.

Solution

Gradient descent: $\theta^{(i+1)} = \theta^{(i)} - \alpha \cdot f(\theta^{(i)})$, where $f(\theta)$ is the function we are trying to minimise. Since α is not a linear function of θ, we cannot establish the equality used in aIII in the previous subsection, and so it is not linearly invariant.

5 Regression for denoising quasar spectra [1]

For the full introduction see the original problem set description located at `http://cs229.stanford.edu/ps/ps1/ps1.pdf`.

Getting the data. We will be using data generated from the Hubble Space Telescope Faint Object Spectrograph (HST-FOS), Spectra of Active Galactic Nuclei and Quasars[2]. We have provided two comma-separated data files located at:

- Training set: `http://cs229.stanford.edu/ps/ps1/quasar_train.csv`
- Test set: `http://cs229.stanford.edu/ps/ps1/quasar_test.csv`

Each file contains a single header row containing 450 numbers corresponding integral wavelengths in the interval [1150, 1600] Å. The remaining lines contain relative flux measurements for each wavelength. Specifically, `quasar_train.csv` contains 200 examples and `quasar_test.csv` contains 50 examples.

a. Locally weighted linear regression

Consider a linear regression problem in which we want to "weight" different training examples differently. Specifically, suppose we want to minimize

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

i Show that $J(\theta)$ can also be written as

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

for an appropriate diagonal matrix W. State clearly what W is.

Solution

[1] Ciollaro, Mattia, et al. "Functional regression for quasar spectra." arXiv:1404.3168 (2014)

[2] `https://hea-www.harvard.edu/FOSAGN/`

$$
\begin{aligned}
J(\theta) &= (X\theta - y)^T W (X\theta - y) \\
&= \begin{bmatrix} (X\theta-y)_1 & (X\theta-y)_2 & \cdots & (X\theta-y)_m \end{bmatrix} \begin{bmatrix} W_1 & 0 & \cdots & 0 \\ 0 & W_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W_m \end{bmatrix} \begin{bmatrix} (X\theta-y)_1 \\ (X\theta-y)_2 \\ \vdots \\ (X\theta-y)_m \end{bmatrix} \\
&= \begin{bmatrix} (X\theta-y)_1 & (X\theta-y)_2 & \cdots & (X\theta-y)_m \end{bmatrix} \begin{bmatrix} W_1(X\theta-y)_1 \\ W_2(X\theta-y)_2 \\ \vdots \\ W_m(X\theta-y)_m \end{bmatrix} \\
&= \sum_{i=1}^{m} W_i (X\theta - y)_i^2
\end{aligned}
$$

where $(X\theta - y)_i = \theta_1 x_1^{(i)} + \ldots + \theta_n x_n^{(i)} - y^{(i)} = \theta^T x^{(i)} - y^{(i)}$ and $W_i = \frac{1}{2} w^{(i)}$.

ii By finding the derivative $\nabla_\theta J(\theta)$ and setting that to zero, generalize the normal equation to this weighted setting, and give the new value of θ that minimizes $J(\theta)$ in closed form as a function for X, W and y.

<u>Solution</u>

$$
\begin{aligned}
\nabla_\theta J(\theta) =& \nabla_\theta \left((X\theta - y)^T W (X\theta - y) \right) = \nabla_\theta \left((\theta^T X^T - y^T)(WX\theta - Wy) \right) \\
=& \nabla_\theta \left(\theta^T XWX\theta - \theta^T X^T Wy - y^T WX\theta + y^T Wy \right) \\
=& \underbrace{\nabla_\theta tr\left(\theta^T XWX\theta\right)}_{I} - \underbrace{\nabla_\theta tr\left(\theta^T X^T Wy\right)}_{II} - \underbrace{\nabla_\theta tr\left(y^T WX\theta\right)}_{III} + \underbrace{\nabla_\theta tr\left(y^T Wy\right)}_{=0 \text{ (constant w.r.t. } \theta)} \\
=& \underbrace{2X^T WX\theta}_{I} - \underbrace{X^T Wy}_{II} - \underbrace{X^T Wy}_{III} \\
=& 2 \cdot \left(X^T WX\theta - X^T Wy \right)
\end{aligned}
$$

Setting the gradient to zero,

$$
\nabla_\theta J(\theta) = 0 \Leftrightarrow X^T WX\theta = X^T Wy \Leftrightarrow \theta = \left(X^T WX \right)^{-1} X^T Wy
$$

iii Suppose we have a training set $\{(x^{(i)}, y^{(i)}); i = 1, \ldots, m\}$ of m independent examples, but in which the $y^{(i)}$'s were observed with differing variances. Specifically, suppose that

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

i.e. $y^{(i)}$ has mean $\theta^T x^{(i)}$ and variance $(\sigma^{(i)})^2$ where the $\sigma^{(i)}$'s are fixed, known constants. Show that finding the maximum likelihood estimate of θ reduces to solving a weighted linear regression problem. State clearly what the $w^{(i)}$'s are in terms of the $\sigma^{(i)}$'s.

<u>Solution</u>

Maximising the likelihood is equivalent to maximising the log-likelihood $l(\theta)$.

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

Since the first term is known and constant, we only need to maximise

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

which is just $J(\theta)$ when $w^{(i)} = \frac{-1}{2\left(\sigma^{(i)}\right)^2}$.

b. Visualising the data

i Use the normal equations to implement (unweighted) linear regression $y = \theta^T X$) on the *first* training example. On one figure, plot both the raw data and the straight line resulting from your fit. State the optimal θ resulting from the linear regression.

Solution

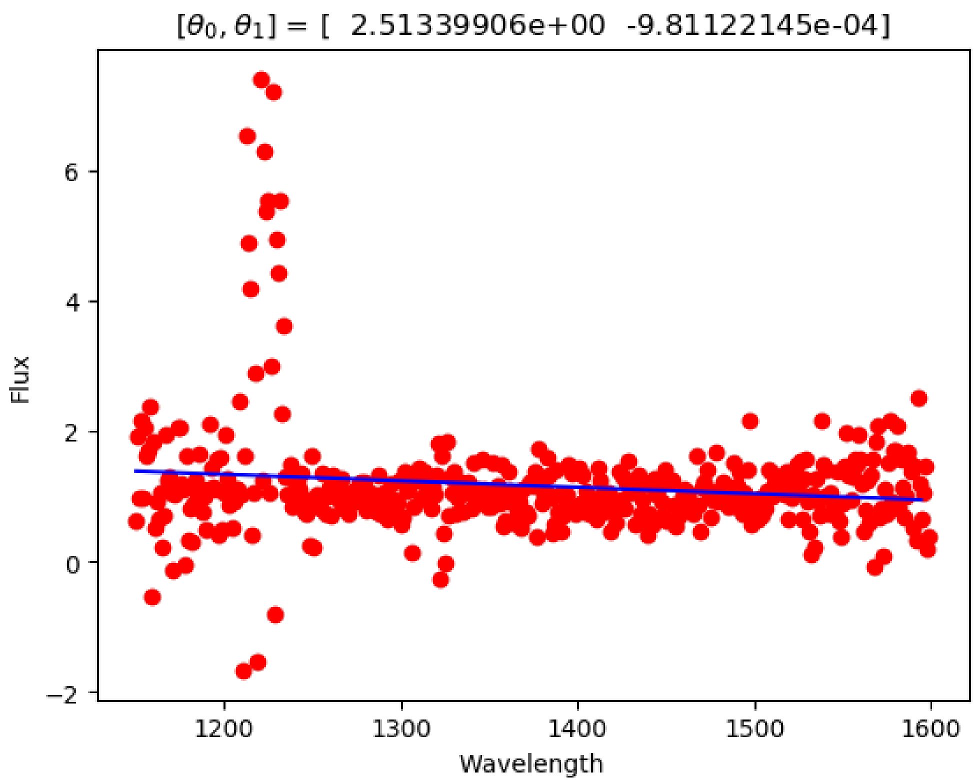


Figure 2: First training example and linear fit

ii Implement locally weighted linear regression on the *first* training example. Use the normal equations derived in part aii. On a different figure, plot both the raw data and the smooth curve resulting from your fit. When evaluating $h(\cdot)$ at a query point x, use weights

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

with bandwidth parameter $\tau = 5$.

Solution

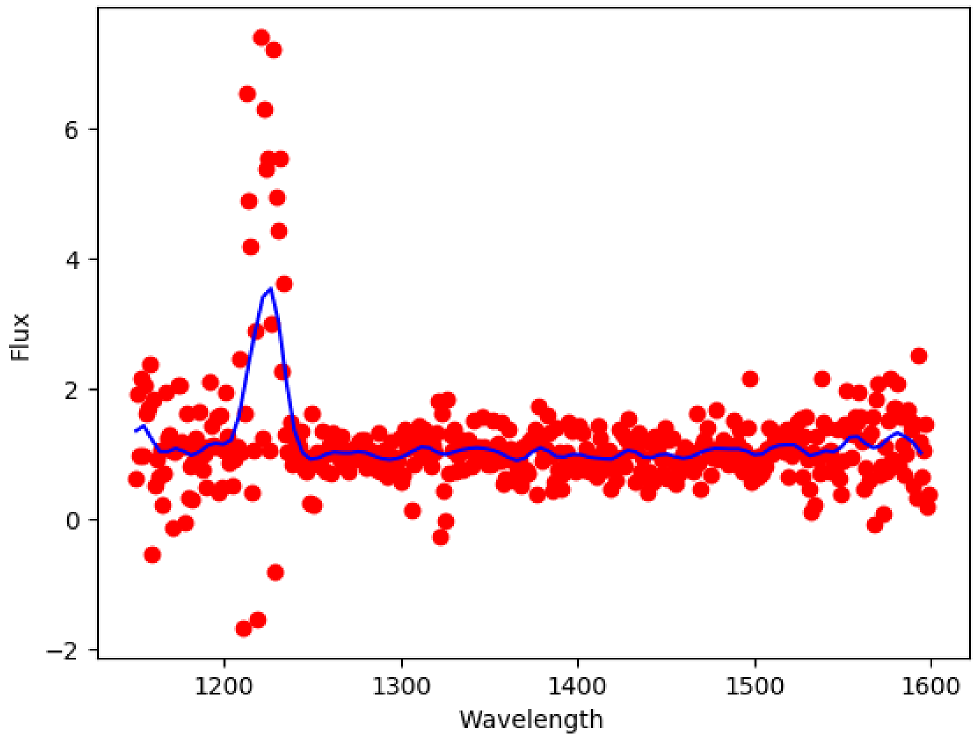
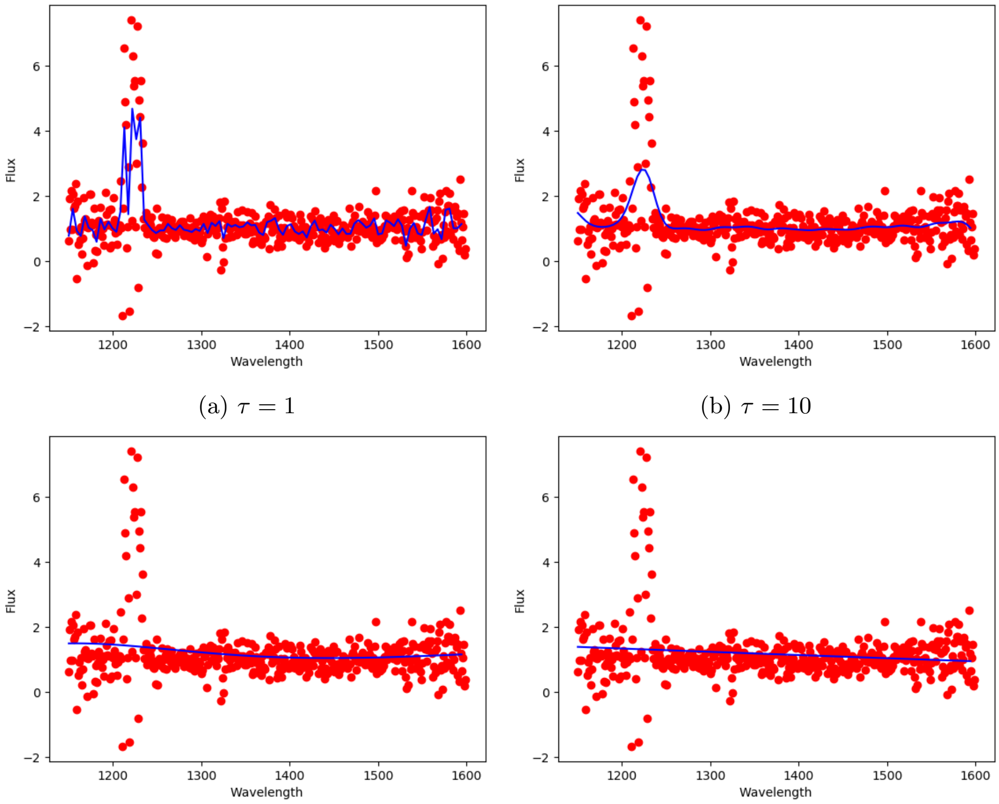


Figure 3: First training example and weighted fit

iii Repeat the previous step four more times with $\tau = 1, 10, 100$ and 1000. Plot the resulting curves. Comment on what happens to the locally weighted linear regression line as τ varies.

Solution



(a) $\tau = 1$

(b) $\tau = 10$

(c) $\tau = 100$

(d) $\tau = 1000$

The larger the bandwidth parameter, the further the examples with a significant weight for a given x. In the limit, when $\tau \to \infty$ the result is a flat distribution of weights, and we retrieve the linear regression model, very underfitted. On the other hand, when $\tau \to 0$ the weight distribution will be very localised, leading to an overfitted model very sensitive to any individual example.

c. Predicting quasar spectra with functional regression

We now wish to predict an entire part of a spectrum - a curve - from noisy observed data. We begin by supposing that we observe a random sample of m absorption-free spectra, which is possible for quasars very close (in a sense relative to the size of the universe!) to Earth. For a given spectrum f, we define f_{right} to be the spectrum to the right of the Lyman-α line. Let f_{left} be the spectrum within the Lyman-α forest region, that is, for lower wavelengths. To make the results cleaner, we define:

$$f(\lambda) = \begin{cases} f_{left}(\lambda) & \text{if } \lambda < 1200 \\ f_{right}(\lambda) & \text{if } \lambda \geq 1300 \end{cases}$$

We will learn a function r (for regression) that maps an observed f_{right} to an unobserved target f_{left} (note that f_{left} and f_{right} don't cover the entire spectrum). This is useful in practice because we observe f_{right} with *only* random noise: there is no systematic absorption, which we cannot observe directly. By predicting f_{left} from a noisy version of f_{right}, we can estimate the unobservable spectrum of a quasar.

We formulate the functional regression task as the goal of learning the function r mapping f_{right} to f_{left}:

$$r\left(f_{right}(\lambda)\right) = \mathbb{E}\left(f_{left}|f_{right}\right)(\lambda)$$

for λ in the Lyman-α forest.

i First, we must smooth the data in the training dataset to make it more useful for prediction. For each $i = 1, \ldots, m$, define $f^{(i)}(\lambda)$ to be the weighted linear regression estimate the i$^{\text{th}}$ spectrum. Use your code from part bii above to smooth all spectra in the training set using $\tau = 5$. Do the same for the test set. Apply smoothing to the entire spectra (including both f_{left} and f_{right}) for both train and test. We will now operate on these smoothed spectra.

Solution

See script `PSET1_5.py` *in the scripts folder for the full code.*

ii Using your estimated regression functions $f^{(i)}$ for $i = 1, \ldots, m$, we now wish to estimate the unobserved spectrum f_{left} of a quasar from its (noisy) observed spectrum f_{right}. To do so, we perform a weighted regression of the *locally weighted regressions.* In particular, given the noisy spectrum observation:

$$f_{obs}(\lambda) = f(\lambda) + noise(\lambda) \text{ for } \lambda \in \{1300, \ldots, 1599\}$$

We define a metric d which takes as input, two spectra f_1 and f_2, and outputs a scalar:

$$d(f_1, f_2) = \sum_i \left(f_1(\lambda_i) - f_2(\lambda_i)\right)^2$$

The metric d computes squared distance between the new datapoint and previous datapoints. If f_1 and f_2 are right spectra, then we take the preceding sum only over $\lambda \in \{1300, \ldots 1599\}$, rather than the entire spectrum.

Based on this distance function, we may definte the nonparametric *functional* regression estimator, which is a locally weighted sum of *functions* f_{left} from the training data (this is like locally weighted linear regression, except that instead of predicting $y \in \mathbb{R}$ we predict a function f_{left}). Specifically, let f_{right} denote the right side of a spectrum, which we have smoothed using locally weighted linear regression. We wish to estimate the associated *left* spectrum f_{left}. Define the function $\ker(t) = \max\{1 - t, 0\}$ and let $neighb_k(f_{right})$ denote the k indices $i \in \{1, \ldots, m\}$ of the training set that are closest to f_{right}, that is

$$d(f^{(i)}_{right}, f_{right}) < (f^{(j)}_{right}, f_{right}) \text{ for all } i \in neighb_k(f_{right}),\ j \notin neighb_k(f_{right})$$

and $neighb_k(f_{right})$ contains exactly k indices. In addition, let

$$h = \max_{i \in \{1,\ldots,m\}} d(f^{(i)}_{right}, f_{right})$$

Then define the estimated function $\widehat{f_{left}} : \mathbb{R} \to \mathbb{R}$ by

$$\widehat{f_{left}}(\lambda) = \frac{\sum_{i \in neighb_k(f_{right})} \ker(d(f_{right}^{(i)}, f_{right})/h) f_{left}^{(i)}(\lambda)}{\sum_{i \in neighb_k(f_{right})} \ker(d(f_{right}^{(i)}, f_{right})/h)}$$

Include f_{right} from training in its own neighbourhood. Recall that $f_{right}^{(i)}$ is the smoothed estimate of the i$^{\text{th}}$ training spectrum.

Construct the functional regression estimate $\widehat{f_{left}}(\lambda)$ for each spectrum in the entire training set using $k = 3$ nearest neighbours: for each $j = 1, \ldots, m$, construct the estimator using $f_{right} = f_{right}^{(j)}$. Then compute the error $d(f_{left}^{(i)}, f_{left})$ between the true spectrum $f_{left}^{(j)}$ and your estimated spectrum $\widehat{f_{left}}$ for each j, and return the average over the training data. What is your average training error?

Solution

Average training error: 2.62

See script `PSET1_5ciii.py` *in the scripts folder for the full code.*

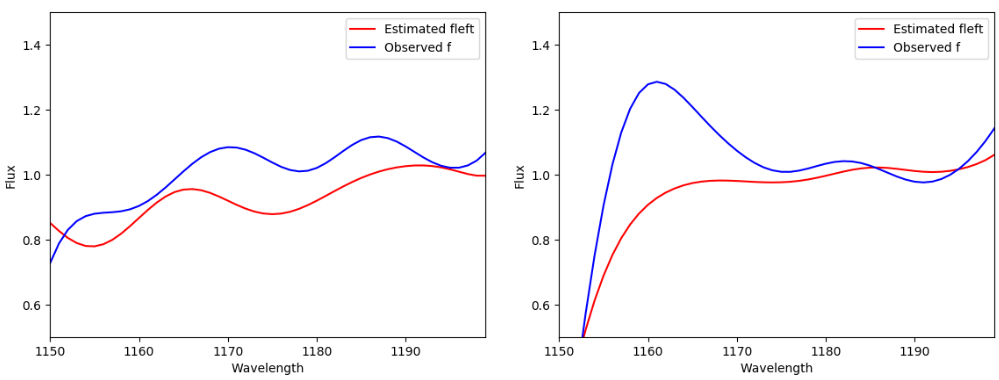
iii Perform function regression on the test set using the same procedure as in the previous subquestion. What is your average test error?

For test examples 1 and 6, inlcude a plot with both the entire smooth spectrum and the fitted curve $\widehat{f_{left}}$ curve on the same graph. You should submit two plots, one for test example 1 and one for test example 6.

Solution

Average test error: 2.66

See script `PSET1_5ciii.py` *in the scripts folder for the full code.*



(a) Example $m = 1$ (b) Example $m = 6$