LinearRegression

Sunday, August 9, 2020 6:01 PM

Data :

	Lot.Area	Bedroom.AbvGr	SalePrice
0	31770	3	215.0
1	11622	2	105.0
2	14267	3	172.0
3	11160	3	244.0
4	13830	3	189.9
...	...	...	...
96	3182	2	151.0
97	2544	2	149.5
98	2544	2	152.0
99	4403	2	222.0
100	2117	3	177.5

101 rows × 3 columns

Notation : Lot Area = x_1, Bedroom.AbvGr = x_2

Sale price = y

Training set : $(x^{(i)}, y^{(i)})$ where $i = 0 : 100$

i.e (X, Y)

hypothesis : $h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$

$= \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2$

$= \sum_{i=0}^{n} \theta_i x_i = \theta^T x$

n = no of training examples

① Iterative : $\theta = \theta + \alpha \sum_{i=1}^{n} (y^{(i)} - h_\theta(x^{(i)})) \, x^{(i)}$

Since, $\frac{\partial}{\partial \theta} J(\theta) = \left(h_\theta(x^{(i)}) - y^{(i)}\right) x^{(i)}$

where, $J(\theta)$ is mean squared error
((ost function / loss)

(2) Implementaion of Normal Equation (Closed form)

$$X = \begin{matrix} x_0 & x_1 & x_2 \end{matrix}$$
$$\begin{bmatrix} 1 & * & * \\ \vdots & * & * \\ 1 & \vdots & \vdots \end{bmatrix} \Big\} \rightsquigarrow n \text{ examples}$$
$$\rightsquigarrow \begin{bmatrix} -(x^{(1)})^T- \\ -(x^{(2)})^T- \\ \vdots \\ -(x^{(n)})^T- \end{bmatrix}$$

$$Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

$$h_\theta(x^{(i)}) = X\theta - Y$$

$$J(\theta) = \frac{1}{2}(X\theta - Y)(X\theta - Y)$$

$$\nabla_\theta J(\theta) = \nabla_\theta \frac{1}{2}(X\theta - Y)(X\theta - Y)$$

$$= \frac{1}{2}\nabla_\theta \left((X\theta)^T X\theta - (X\theta)^T Y - Y(X\theta) + Y^T Y\right)$$

$$= \frac{1}{2} \nabla_\theta \left(\theta^T (X^T X) \theta - 2 (X^T Y)^T \theta \right)$$

$$= \frac{1}{2} \left(2 X^T X \theta - 2 X^T Y \right)$$

$$= X^T X \theta - X^T Y$$

To find $\min_\theta \theta$,

$$\nabla_\theta J(\theta) = 0 \Rightarrow X^T X \theta - X^T Y = 0$$

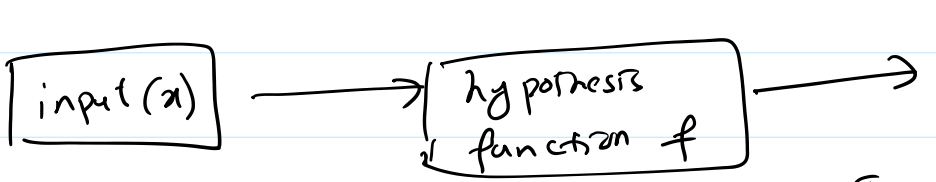
$$\theta = (X^T X)^{-1} X^T Y$$

[Using, $a^T b = b^T a$, $\nabla_a b^T a = b$ $\nabla_a x^T A x = 2Ax$ for a symmetric matrix 'A']

Code Implementation Note

Define X and Y, then use normal equation to find 'θ' parameter directly.

(A) Derivation of logistic Regression



"learn f"

True label → y

Div(ŷ, y) is loss.

$y|x;\theta \sim$ Bernoulli(ϕ)

or generally ~ Exponential family (η)

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

$h(x) = E[y|x]$

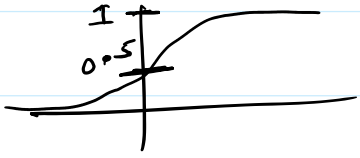
or $h_\theta(x) = E[y|x;\theta] = \phi = \frac{1}{1+e^{-\eta}}$

$= \frac{1}{1+e^{-\theta^T x}}$

$g(\eta) = E[T(y);\eta]$ is canonical response function

In our case Sigmoid.

$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$ when g is sigmoid



PMF

$P(y|x;\theta) = \phi^y (1-\phi)^{1-y}$

$$= (h_\theta(x))^y \left(1 - h_\theta(x)\right)^{1-y}$$

Likelihood (function of θ)

$$L(\theta) = P(y \mid X; \theta)$$

$$= \prod_{i=1}^{n} P(y^{(i)} \mid x^{(i)}; \theta)$$

$$= \prod_{i=1}^{n} \left(h_\theta(x^{(i)})\right)^{y^{(i)}} \left(1 - h_\theta(x^{(i)})\right)^{(1-y^{(i)})}$$

$\begin{cases} y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)} \\ \text{assumption,} \\ \epsilon^{(i)} \text{ are IID} \\ \text{distributed} \end{cases}$

Since log is strictly increasing function, we could find θ that maximize $\log(L(\theta))$

So,

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{n} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))$$

$$\frac{\partial}{\partial \theta_j} \ell(\theta) = (y - h_\theta(x)) x_j$$

Note $[\, g'(x) = g(x)(1 - g(x))$

So, $\theta := \theta + \alpha \left(y^{(i)} - h_\theta(x^{(i)})\right) x$

(B) Problem set 1(A) and 1(B)

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

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

where $y^{(i)} \in \{0, 1\}$, $h_\theta(x) = g(\theta^T x)$ and $g(z) = 1/(1 + e^{-z})$.
Find the Hessian H of this function, and show that for any vector z, it holds true that

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

Hint: You may want to start by showing that $\sum_i \sum_j z_i x_i x_j z_j = (x^T z)^2 \geq 0$. Recall also that $g'(z) = g(z)(1 - g(z))$.
Remark: This is one of the standard ways of showing that the matrix H is positive semi-definite, written "$H \succeq 0$." This implies that J is convex, and has no local minima other than the global one. If you have some other way of showing $H \succeq 0$, you're also welcome to use your method instead of the one above.
Answer:

$\nabla_\theta J(\theta) = \begin{bmatrix} \frac{\partial}{\partial \theta_1} J(\theta) \\ \vdots \\ \frac{\partial}{\partial \theta_i} J(\theta) \end{bmatrix}$ For simplicity assume we have only one training example on the set. So we have

$$\frac{\partial}{\partial \theta_i} J(\theta) = \frac{\partial}{\partial \theta_i}\left(y \log(h_\theta(x)) + (1-y)\log(1-h_\theta(x))\right)$$

$$= \frac{y}{h_\theta(x)} \frac{\partial}{\partial \theta_i} h_\theta(x) + \frac{(1-y)}{1-h_\theta(x)} \frac{\partial}{\partial \theta_i}(1 - h_\theta(x))$$

$$= g(\theta^T x)(1 - g(\theta^T x)) x_i \left(\frac{y}{h_\theta(x)} + \frac{y-1}{1-h_\theta(x)}\right)$$

$$= g'(\theta^T x) x_i \left(\frac{y - h_\theta(x)}{h_\theta(x)(1-h_\theta(x))}\right)$$

$$= g(\theta^T x)(1 - g(\theta^T x)) x_i \frac{(y - h_\theta(x))}{g(\theta^T x)}$$

$$= (y - h_\theta(x)) x_i$$

$\left(\because h_\theta(x) = g(\theta^T x), \; g'(\theta^T x) = g(\theta^T x)(1 - g(\theta^T x))\right)$

$i \leftrightarrow j$ s.t. $\frac{\partial}{\partial \theta_j} J(\theta) = (y - h_\theta(x)) x_j$

Now

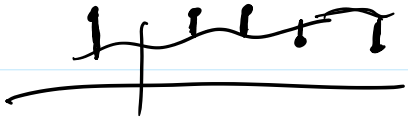
$H_{ij} = x_j \, h_\theta(x)(h_\theta(x) - 1) x_i$

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

$$= \log(n)(\log n + 1) \sum_i \sum_j z_{ni} z_{nj}$$
$$= \log(n)(\log n + 1)(nz)^2$$
$$\geq 0$$

So $\Sigma Hz \geq 0$ therefore $H \geq 0$ i.e. H is PSD so $S(\theta)$ is convex

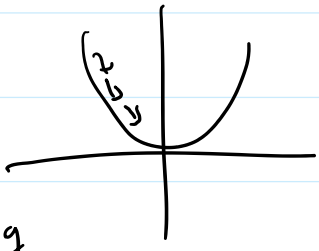
Use input sample to estimate error.



$$E[div(f(x;\theta), g(x))] \approx \frac{1}{n} \sum_{i=1}^{n} div(f(x_i;\theta), \hat{y}_i)$$

Implementation Notes (C)

Instead of iteratively going down hill by 'α' learning rate (Gradient descent), we used quadratic approximation using Newton's method. For vector input,



$$\theta := \theta - H^{-1} \nabla_\theta \ell(\theta)$$ (Newton-Raphson method "numerical analysis")

H is Hessian, $H_{ij} = \frac{\partial^2}{\partial \theta_i \theta_j} \ell(\theta)$

Algorithm

(Training)

① Initialize the parameters θ

(2) Define $h_\theta(x) = g(\theta^T x)$ where g is Sigmoid

$y \in \{0, 1\}$ (in PS1)

(2) Compute $\nabla_\theta J(\theta)$

empirical loss, $\frac{1}{n} (y - h_\theta(x)) x$

(3) Compute Hessian H,

average, $\frac{1}{n} h_\theta(x)(1 - h_\theta x)\, x^T x$

(4) Use Newton-Raphson, Repeat until Convergence

Testing

(5) To predict, feed the input x to hypothesis, $h_\theta(x)$ to produce prediction.

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

Answer:

GDA

Wednesday, August 12, 2020 9:36 AM

(A) Derivation of GDA model

Comparison with Discriminative algorithm:

In Disc we have to classify $y=0$ or $y=1$ (binary example) given feature: x. Logictic Reg tried to find a straight line (Decision boundary) that separates the two level. While, GDA looks at how $y=1$ and $y=0$ are maple up of and based the prediction of a example by comparing if it fits $y=1$ or $y=0$ properties.

Disc Algo: Find $p(y|x)$ directly

GDA: Find $p(x|y)$ and $p(y)$ then use from to to find $p(y|x)$

↳ makes strong assumption

eg, $p(x|y=0)$, $p(x|y=1)$

$p(y)$ → class prior

Use bayes rule, $p(y|x) = \frac{p(x|y)\,p(y)}{p(x)}$

total prob

$$\arg\max_y p(y|x) = \arg\max_y p(x|y)\,p(y)$$

Model:

$y \sim \text{Bimonial}(\phi)$

$x|y=0 \sim \mathcal{N}(\mu_0, \Sigma)$

$x|y=1 \sim \mathcal{N}(\mu_1, \Sigma)$

$p(y)$

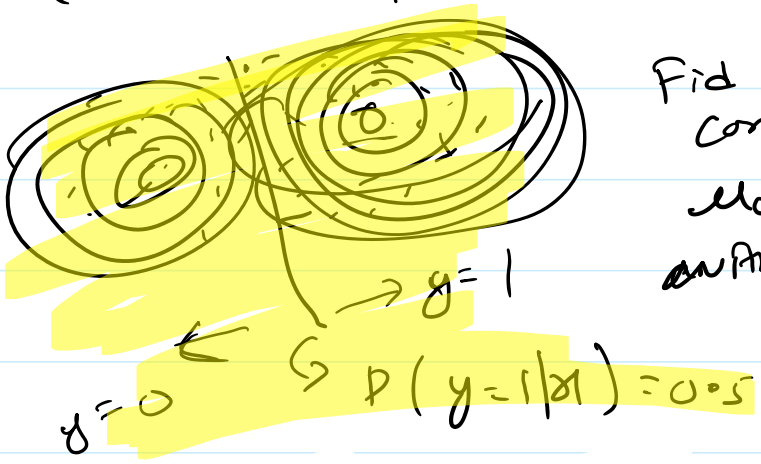
$p(x|y=0)$

$p(x|y=1)$ } → find prob mass/ density funct

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

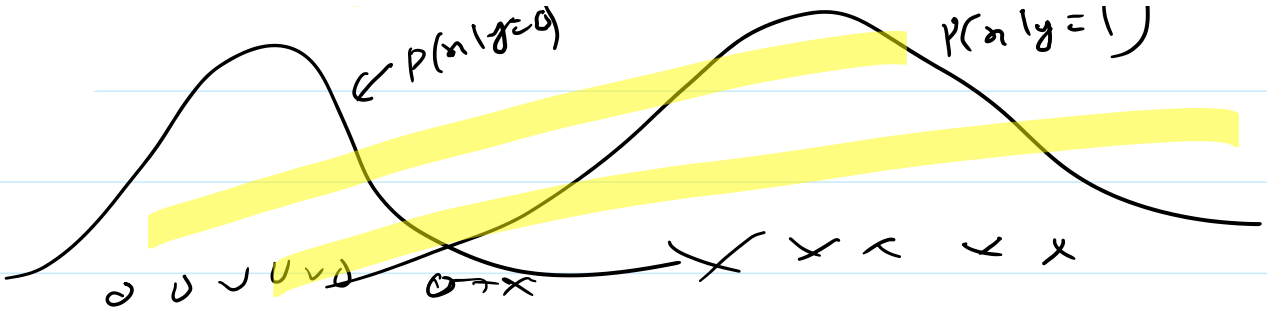
$$= \log \prod_{i=1}^{n} p(x^{(i)}|y^{(i)}; \mu_0, \mu_1, \Sigma) \; p(y^{(i)}; \phi)$$

Take derivative of ℓ w.r.t parameter $(\phi, \mu_0, \mu_1, \Sigma)$ to find the parameter that maximize the likelihood. (Done on Probb Set below)

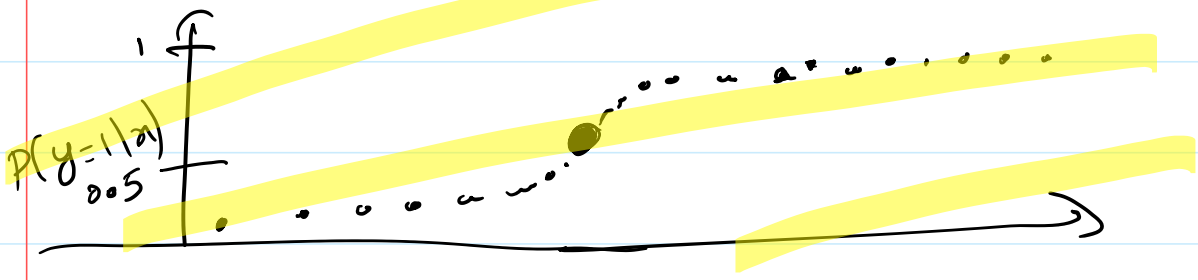


Fid gaussian controlled by μ_0, μ_1 to shift and with shape shape Σ.

GDA and logistic



$P(y=1) = 0.5$ Since data is half and half



Generative

$y \sim \text{Binomial}(\phi)$

$x|y=0 \sim \mathcal{N}(\mu_0, \Sigma)$

$x|y=1 \sim \mathcal{N}(\mu_1, \Sigma)$

$\Rightarrow$ $\not\Leftarrow$

discriminative (logistic)

$$p(y=1|x) = \frac{1}{1+e^{-\theta^T x}}$$

(B) Problem Set 1 (C) (d) (e)

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

$$
\begin{aligned}
p(y) &= \begin{cases} \phi & \text{if } y = 1 \\ 1 - \phi & \text{if } y = 0 \end{cases} \\
p(x|y=0) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right) \\
p(x|y=1) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right),
\end{aligned}
$$

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

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

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

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

Answer:

$$
P(y=1|x) = \frac{P(x|y=1)\,P(y=1)}{P(x|y=1)\,P(y=1) + P(x|y=0)\,P(y=0)}
$$

$$
= \frac{1}{1 + \frac{P(x|y=0)\,P(y=0)}{P(x|y=1)\,P(y=1)}}
$$

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

$$
z = e^{\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0) - \left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)\right)} e^{\log\left(\frac{1-\phi}{\phi}\right)}
$$

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

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

So we can take

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

$\theta^T x + \theta_0$ → $e^{\cdot}$

for coding part

Using $(A+B)C = (AC)+(BC)$ Also $A^T+B^T = (A+B)^T$

(d) [7 points] 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 $|\Sigma| = \sigma^2$. Given the dataset, we claim that the maximum likelihood estimates of the parameters are given by

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

The log-likelihood of the data is

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

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

Answer:

$$\ell = \log\prod_{i=1}^{m} 1\{y^{(i)}=0\}\, p(x^{(i)}|y^{(i)}=0)p(y^{(i)}=0) + 1\{y^{(i)}=1\}\, p(x^{(i)}|y^{(i)}=1)\,p(y^{(i)}=1)$$

$$= \log\prod_{i=1}^{m}\left[1\{y^{(i)}=0\}\frac{1}{\sqrt{2\pi}\sigma^2}\exp\left(\frac{-(x-\mu_0)^2}{2\sigma^2}\right)(1-\phi) + \cdots\right]$$

$$= \sum_{i=1}^{m}\log\left[1\{y^{(i)}=0\}\frac{1}{\sqrt{2\pi}\sigma^2}\exp\left(-\frac{(x-\mu_0)^2}{2\sigma^2}\right)(1-\phi) + \cdots\right]$$

$$= \sum_{i=1}^{m}\log\left(1\{y^{(i)}=0\}\frac{1}{\sqrt{2\pi}\sigma^2}\exp\left(-\frac{(x-\mu_0)^2}{2\sigma^2} + 1\{y^{(i)}=1\}\cdots\right)\right) + \log\left(\phi^{y^{(i)}}(1-\phi)^{1-y^{(i)}}\right)$$

$$\frac{\partial\ell}{\partial\phi} = \sum_{i=1}^{m}\frac{\partial}{\partial\phi}\left[y^{(i)}\log(\phi) + (1-y^{(i)})\log(1-\phi)\right] = 0$$

$$\sum_{i=1}^{m}\frac{y^{(i)}}{\phi} + \frac{1-y^{(i)}}{1-\phi} = 0$$

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

$y^{(i)} \in \{0,1\}$

$$\sum_{i=1}^{m} y^{(i)} + \phi\sum_{i=1}^{m}(1-2y^{(i)}) = 0 \Rightarrow \phi = \frac{\sum_{i=1}^{m} y^{(i)}}{\sum_{i=1}^{m}(1-y^{(i)})}$$

$$\sum_{i=1}^{m} y^{(i)} + \phi \sum_{i=1}^{m} (1 - 2y^{(i)}) = 0 \Rightarrow \phi = \frac{\sum_{i=1}^{m} y^{(i)}}{\sum_{i=1}^{m} (2y^{(i)} - 1)}$$

$$\phi = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{m} \Rightarrow \sum_{i=1}^{m} 1 \left[\text{for any } y^{(i)}\right]$$

$$\frac{d}{dx} \log(x-2)^2 = \frac{1}{(x-2)^2} e^{(x-2)^2} \frac{d}{dx}(x-2)^2$$

$$\ell = \frac{\partial}{\partial \mu_0} \sum_{i=1}^{m} \log\left(1\{y^{(i)}=0\} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x^{(i)}-\mu_0)^2}{2\sigma^2}\right)\right)$$

$$= \sum_{i=1}^{m} 1\{y^{(i)}=0\} \frac{\partial}{\partial \mu_0} \log\left(e^{-\frac{(x^{(i)}-\mu_0)^2}{2\sigma^2}}\right) \quad \text{or } \log(e^x) = x \left[\log \text{ and } e \text{ are inverse functions}\right]$$

$$= \sum_{i=1}^{m} 1\{y^{(i)}=0\} \frac{\partial}{\partial \mu_0}\left(-\frac{(x^{(i)}-\mu_0)^2}{2\sigma^2}\right) = -1\{y^{(i)}=0\} \sum_{i=1}^{m} \frac{x^{(i)} - \mu_0}{\sigma^2} = 0$$

$$\text{Sol} \quad -\sum_{i=1}^{m} 1\{y^{(i)}=0\} x^{(i)} + \sum_{i=1}^{m} 1\{y^{(i)}=0\} \mu_0 = 0$$

$$\therefore \mu_0 = \sum_{i=1}^{m} 1\{y^{(i)}=0\} x^{(i)} / \sum_{i=1}^{m} 1\{y^{(i)}=0\}$$

Similarly for $\Sigma = \sigma^2$, Note $y = \sigma^2$

$$\sqrt{y} = \sigma$$

Bad approximation of GDA on data set 1 $\Rightarrow$ Since the dataset is not Gaussian
Fix (take $\log(x_i^{(i)}$'s))

⑥ Implementation Notes

Algorithm

① Compute $\phi, \mu_0, \mu_1, \Sigma$ using above formula

② compute θ using (*)

(3) Find $P(y=1|x)$; if $P(y=1|x) \geq 0.5$ predict that label is '1' else predict that label is 0.

IncompleteLabels

Wednesday, August 12, 2020 1:46 PM

2. **[30 points] Incomplete, Positive-Only Labels**

In this problem we will consider training binary classifiers in situations where we do not have full access to the labels. In particular, we consider a scenario, which is not too infrequent in real life, where we have labels only for a subset of the positive examples. All the negative examples and the rest of the positive examples are unlabelled.

That is, we assume a dataset $\{(x^{(i)}, t^{(i)}, y^{(i)})\}_{i=1}^{m}$, where $t^{(i)} \in \{0, 1\}$ is the "true" label, and where

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

All labeled examples are positive, which is to say $p(t^{(i)} = 1 \mid y^{(i)} = 1) = 1$, but unlabeled examples may be positive or negative. Our goal in the problem is to construct a binary classifier h of the true label t, with only access to the partial labels y. In other words, we want to construct h such that $h(x^{(i)}) \approx p(t^{(i)} = 1 \mid x^{(i)})$ as closely as possible, using only x and y.

Real world example: Suppose we maintain a database of proteins which are involved in transmitting signals across membranes. Every example added to the database is involved in a signaling process, but there are many proteins involved in cross-membrane signaling which are missing from the database. It would be useful to train a classifier to identify proteins that should be added to the database. In our notation, each example $x^{(i)}$ corresponds to a protein, $y^{(i)} = 1$ if the protein is in the database and 0 otherwise, and $t^{(i)} = 1$ if the protein is involved in a cross-membrane signaling process and thus should be added to the database, and 0 otherwise.

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

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

Note this is equivalent to saying that labeled examples were selected uniformly at random from the set of positive examples. Prove that the probability of an example being labeled differs by a constant factor from the probability of an example being positive. That is, show that $p(t^{(i)} = 1 \mid x^{(i)}) = p(y^{(i)} = 1 \mid x^{(i)})/\alpha$ for some $\alpha \in \mathbb{R}$.

Answer:

Qno.2

a $P(y=1|t=1,x)\,P(t=1|x)\,P(x) = P(y=1,t=1,x) = P(t=1|y=1,x)\,P(y=1|x)\,P(x)$

So, $P(t=1|x) = P(y=1|x)\,\dfrac{P(t=1|y=1,x)}{P(y=1|t=1,x)}$

$P(t=1|y=1,x)=1$, $P(y=1|t=1,x) = P(y=1|t=1)$

$P(t=1|x) = \dfrac{P(y=1|x)}{P(y=1|t=1)}$ $\qquad P(y=1|t=1)=\alpha$

(b) $h(x) \approx P(y=1|x) = P(t=1|x)/\alpha \approx \alpha$ for all $x \in V_t$

(b) [5 points] Suppose we want to estimate α using a trained classifier h and a held-out validation set V. Let V_+ be the set of labeled (and hence positive) examples in V, given by $V_+ = \{x^{(i)} \in V \mid y^{(i)} = 1\}$. Assuming that $h(x^{(i)}) \approx p(y^{(i)} = 1 \mid x^{(i)})$ for all examples $x^{(i)}$, show that

$$h(x^{(i)}) \approx \alpha \quad \text{for all } x^{(i)} \in V_+.$$

You may assume that $p(t^{(i)} = 1 \mid x^{(i)}) \approx 1$ when $x^{(i)} \in V_+$.

Answer:

(c) [5 points] **Coding problem.** The following three problems will deal with a dataset which we have provided in the following files:

`data/ds3_{train,valid,test}.csv`

Each file contains the following columns: x_1, x_2, y, and t. As in Problem 1, there is one example per row.

First we will consider the ideal case, where we have access to the true t-labels for training. In `src/p02cde_posonly`, write a logistic regression classifier that uses x_1 and x_2 as input features, and train it using the t-labels (you can ignore the y-labels for this part). Output the trained model's predictions on the test set to the file specified in the code.

Answer:

(d) [5 points] **Coding problem.** We now consider the case where the t-labels are unavailable, so you only have access to the y-labels at training time. Add to your code in `p02cde_posonly.py` to re-train the classifier (still using x_1 and x_2 as input features), but using the y-labels only.

Answer:

(e) [10 points] **Coding problem.** Using the validation set, estimate the constant α by averaging your classifier's predictions over all labeled examples in the validation set:

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

Add code in `src/p02cde_posonly.py` to rescale your classifier's predictions from part (d) using the estimated value for α.

Finally, using a threshold of $p(t^{(i)} = 1 \mid x^{(i)}) = 0.5$, make three separate plots with the decision boundaries from parts (c) - (e) plotted on top of the test set. Plot x_1 on the horizontal axis and x_2 on the vertical axis, and use two different symbols for the positive ($t^{(i)} = 1$) and negative ($t^{(i)} = 0$) examples. In each plot, indicate the separating hyperplane with a red line.

Answer:

Remark: We saw that the true probability $p(t \mid x)$ was only a constant factor away from $p(y \mid x)$. This means, if our task is to only rank examples (*i.e.* sort them) in a particular order (e.g, sort the proteins in order of being most likely to be involved in transmitting signals across membranes), then in fact we do not even need to estimate α. The rank based on $p(y \mid x)$ will agree with the rank based on $p(t \mid x)$.

Possion_GLM

Wednesday, August 12, 2020 2:10 PM

3. **[25 points] Poisson Regression**

(a) [5 points] 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 the values for $b(y)$, η, $T(y)$, and $a(\eta)$.

Answer:

Ans. (a)

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

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

So, $b(y) = 1/y!$ $\quad T(y) = y$

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

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

Answer:

(c) [7 points] For a training set $\{(x^{(i)}, y^{(i)}); i = 1, \ldots, m\}$, 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 update rule for learning using a GLM model with Poisson responses y and the canonical response function.

Answer:

(b)

$h_\theta(x) = E(T(y)|x;\theta) = \lambda = e^{\eta} = e^{\theta^T x}$ So, $h_\theta(x) = e^{\theta^T x}$ [$g(\eta) = e^{\eta}$ ← canonical response function]

By assumption ③ GLM

$\ell(\theta) = \sum_{i=1}^{m} b(y)\, e^{\eta^T T(y) - a(\eta)} = \sum_{i=1}^{m} \log\left(\frac{1}{y!} e^{\log\lambda\, y - e^{\eta}}\right) = \sum_{i=1}^{m} \log\left(\frac{1}{y!}\right) + \log e^{y\log y}$

$= \sum_{i=1}^{m} \log(1/y!) + y\log\lambda - e^{\theta^T x}$ (for $\eta = \theta^T x$)

$\frac{\partial}{\partial\theta_j}\ell(\theta) = \sum_{i=1}^{m} \frac{\partial}{\partial\theta_j}\left(\log(1/y!) + y^{(i)}\log\lambda - e^{\theta^T x^{(i)}}\right) = \sum_{i=1}^{m} y^{(i)} x_j^{(i)} - e^{\theta^T x^{(i)}} x_j^{(i)}$

So, Stochastic Gradient ascent rule is; $\boxed{\theta := \theta + \alpha\left(y^{(i)} - e^{\theta^T x^{(i)}}\right) x^{(i)}}$

(d) [7 points] **Coding problem.** Consider a website that wants to predict its daily traffic. The website owners have collected a dataset of past traffic to their website, along with some features which they think are useful in predicting the number of visitors per day. The dataset is split into train/valid/test sets and follows the same format as Datasets 1-3:

`data/ds4_{train,valid}.csv`

We will apply Poisson regression to model the number of visitors per day. Note that applying Poisson regression in particular assumes that the data follows a Poisson distribution whose natural parameter is a linear combination of the input features (*i.e.*, $\eta = \theta^T x$). In `src/p03d_poisson.py`, implement Poisson regression for this dataset and use gradient ascent to maximize the log-likelihood of θ.

Answer:

Convexity of Generalized Linear Models

Wednesday, August 12, 2020 2:21 PM

(a) [5 points] Derive an expression for the mean of the distribution. Show that $\mathbb{E}[Y \mid X; \theta]$ can be represented as the gradient of the log-partition function a with respect to the natural parameter η.

Hint: Start with observing that $\frac{\partial}{\partial \eta} \int p(y; \eta) dy = \int \frac{\partial}{\partial \eta} p(y; \eta) dy$.

Answer:

$$\int p(y;\eta)\, dy = 1$$

$$\frac{\partial}{\partial \eta} \int p(y;\eta)\, dy = 0$$

$$\int \frac{\partial}{\partial \eta} p(y;\eta)\, dy = 0$$

$$\int \frac{\partial}{\partial \eta} b(y) \exp(\eta y - a(\eta))\, dy = 0$$

$$\int b(y) \exp(\eta y - a(\eta))(y - a'(\eta))\, dy = 0$$

$$\int \{ p(y;\eta) - p(y;\eta) a'(\eta) \}\, dy = 0$$

$$\int y\, p(y;\eta)\, dy = \int p(y;\eta)\, a'(\eta)\, dy$$

$$E[y \mid x; \eta] = a'(\eta) \int p(y;\eta)\, dy$$

So, $E[Y \mid x; \eta] = a'(\eta)$

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

Answer:

$$\frac{\partial}{\partial \eta^2} \int p(y;\eta)\, dy = \frac{\partial}{\partial \eta} \int (y - a'(\eta))\, b(y) \exp(\eta y - a(\eta))\, dy$$

$$= \int \frac{\partial}{\partial \eta} (\quad)\, dy$$

$$= \int (y - a'(\eta))^2 b(y) \exp(\eta y - a\eta) - b(y) \exp(\eta y - a\eta)\, a'\eta$$

$$= \int b(y) \exp(\eta y - a(\eta)) \left[(y - a'(\eta))^2 - a''(\eta) \right] dy$$

$$= \int p(y;\eta) \left[(y - a'(\eta))^2 - a''(\eta) \right] dy$$

$$= \int p(y;\eta)(y - a'(\eta))^2 dy - \int p(y;\eta)\, a''(\eta)\, dy = 0$$

or/ $$\int p(y;\eta)\left[y^2 - 2y\, \underbrace{a'(\eta)}_{E[Y|x]} + (a'\eta)^2 \right] dy = a''(\eta)$$

or/ $$\int p(y;\eta) y^2 dy - 2\, E[Y|x]^2 + E[Y|x]^2$$

$$E[Y^2|x;\eta] - E[Y|x]^2 = a''(\eta)$$

$$\therefore \quad Var(Y|x;\theta) = a''(\eta)$$

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

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

Answer:

Remark: The main takeaways from this problem are:

- Any GLM model is convex in its model parameters.
- The exponential family of probability distributions are mathematically nice. Whereas calculating mean and variance of distributions in general involves integrals (hard), surprisingly we can calculate them using derivatives (easy) for exponential family.

$$\ell(\theta) = \sum_{i=1}^{m} \log\left(p(y^{(i)}|x^{(i)};\theta) \right) = \sum_{i=1}^{m} \log p(y^{(i)};\eta)$$

$$\ell(\theta) = -\sum_{i=1}^{m} \log b(y^{(i)}) + y^{(i)} \theta^T x^{(i)} - a(\theta^T x^{(i)})$$

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

$$= \sum_{i=1}^{m} \left[a''(\theta^T x^{(i)}) \, x_j^{(i)} x_k^{(i)} \right]$$

$\forall z,$

$$z^T H z = \sum_i \sum_j z_i H_{ij} z_j = \sum_i \sum_j z_i \, a''(\theta^T x) \, x_j x_k z_j$$

$$= a''(\theta^T x)(x^T z)^2$$

Since, $a''(\theta^T x)$ is $Var(y|x;\theta) \geq 0$, $a''(\theta^T x)$ is non-negative

Hence $z^T H z \geq 0$ So, ℓ is convex

Locally weighted linear regression

Wednesday, August 12, 2020 2:29 PM

5. **[25 points] Locally weighted linear regression**

(a) [10 points] Consider a linear regression problem in which we want to "weight" different training examples differently. Specifically, suppose we want to minimize

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

In class, we worked out what happens for the case where all the weights (the $w^{(i)}$'s) are the same. In this problem, we will generalize some of those ideas to the weighted setting.

i. [2 points] Show that $J(\theta)$ can also be written

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

for an appropriate matrix W, and where X and y are as defined in class. Clearly specify the value of each element of the matrix W.

ii. [4 points] If all the $w^{(i)}$'s equal 1, then we saw in class that the normal equation is

$$X^T X\theta = X^T y,$$

and that the value of θ that minimizes $J(\theta)$ is given by $(X^T X)^{-1} X^T y$. By finding the derivative $\nabla_\theta J(\theta)$ and setting that to zero, generalize the normal equation to this weighted setting, and give the new value of θ that minimizes $J(\theta)$ in closed form as a function of X, W and y.

iii. [4 points] Suppose we have a dataset $\{(x^{(i)}, y^{(i)}); i = 1\ldots, m\}$ of m independent examples, but we model the $y^{(i)}$'s as drawn from conditional distributions with different levels of variance $(\sigma^{(i)})^2$. Specifically, assume the model

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

That is, each $y^{(i)}$ is drawn from a Gaussian distribution with 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.

Answer:

(i) $X = \begin{bmatrix} 1 & \\ \vdots & \\ m & \end{bmatrix}$ $\begin{matrix} x_1 & \cdots & x_n \end{matrix}$

$W = \begin{bmatrix} \frac{w^{(1)}}{2} & 0 & \\ & \frac{w^{(2)}}{2} & \\ 0 & & \frac{w^{(m)}}{2} \end{bmatrix}$

$s.t \quad W(X\theta - y) = \begin{bmatrix} \frac{w^{(1)}}{2}(\theta^T x^{(1)} - y^{(1)}) \\ \vdots \\ \frac{w^{(m)}}{2}(\theta^T x^{(m)} - y^{(m)}) \end{bmatrix}$

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

$= \nabla_\theta (\theta^T X^T - y^T)(WX\theta - Wy)$

$= \nabla_\theta (\theta^T X^T W X\theta - \theta^T X^T Wy - y^T WX\theta + y^T Wy)$

$= \nabla_\theta (\theta^T X^T W X\theta - 2y^T WX\theta + y^T Wy)$

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

$$X^T W X \theta = X^T W^T y$$

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

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

Maximizing $\ell(\theta)$ is equivalent to minimizing $J\theta = \sum_{i=1}^{m} \frac{1}{2(\sigma^{(i)})^2}(y^{(i)} - \theta^T x^{(i)})^2$

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

(b) [10 points] **Coding problem.** We will now consider the following dataset (the formatting matches that of Datasets 1-4, except $x^{(i)}$ is 1-dimensional):

`data/ds5_{train,valid,test}.csv`

In `src/p05b_lwr.py`, implement locally weighted linear regression using the normal equations you derived in Part (a) and using

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

Train your model on the `train` split using $\tau = 0.5$, then run your model on the `valid` split and report the mean squared error (MSE). Finally plot your model's predictions on the validation set (plot the training set with blue 'x' markers and the validation set with a red 'o' markers). Does the model seem to be under- or overfitting?

Answer:

(c) [5 points] **Coding problem.** We will now tune the hyperparameter τ. In `src/p05c_tau.py`, find the MSE value of your model on the validation set for each of the values of τ specified in the code. For each τ, plot your model's predictions on the validation set in the format described in part (b). Report the value of τ which achieves the lowest MSE on the `valid` split, and finally report the MSE on the `test` split using this τ-value.

Answer: