Chris Piech
CS109

PSet #6 Soln
June 5, 2017

Problem Set #6 Solutions

1.

Recall that the density function for the Exponential distribution with given parameter λ is $f(x \mid \lambda) = \lambda e^{-\lambda x}$, where x ≥ 0. Thus, the log-likelihood function to maximize is:

$$LL(\lambda) = \sum_{i=1}^{n} \log(\lambda e^{-\lambda X_i}) = \sum_{i=1}^{n} \left[\log(\lambda) + \log e^{-\lambda X_i} \right] = n\log(\lambda) - \lambda \sum_{i=1}^{n} X_i$$

Taking the derivative of LL(λ) w.r.t. λ, and setting it to 0, yields:

$$\frac{\partial LL(\lambda)}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} X_i = 0$$

Solving for λ gives us: $\frac{n}{\lambda} = \sum_{i=1}^{n} X_i \quad \Rightarrow \quad \hat{\lambda} = \frac{1}{\frac{1}{n}\sum_{i=1}^{n} X_i} = \frac{1}{\overline{X}}$

2.

For the n I.I.D sample pairs (X_1, Y_1), (X_2, Y_2) … (X_n, Y_n), the log-likelihood function for θ = {θ_1, θ_2} under the assumption that $Y = \theta_1 X + \theta_2 + Z$, $Z \sim N(0, \sigma^2)$ is calculated as follows:

$$LL(\theta) = \log \prod_{i=1}^{n} f(X_i, Y_i \mid \theta) = \log \prod_{i=1}^{n} f(Y_i \mid X_i, \theta) * f(X_i)$$

Where

$$f(Y_i \mid X_i, \theta) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(Y_i - \theta_1 X_i - \theta_2)^2}{2\sigma^2}}$$

Combining these two, we get:

$$LL(\theta) = \log \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(Y_i - \theta_1 X_i - \theta_2)^2}{2\sigma^2}} f(X_i) = \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(Y_i - \theta_1 X_i - \theta_2)^2}{2\sigma^2}} + \sum_{i=1}^{n} \log f(X_i)$$

Simplifying this expression gives us:

$$LL(\theta) = n\log(\frac{1}{\sqrt{2\pi\sigma}}) - \frac{1}{2\sigma^2} * \sum_{i=1}^{n} (Y_i - \theta_1 X_i - \theta_2)^2 + \sum_{i=1}^{n} \log f(X_i)$$

3.

If the first *k* input variables X_1, X_2, …, X_k are actually all *identical* copies of each other, this is problematic for Naïve Bayes as the identical variables clearly violate the assumption of conditional independence made by the classifier (unless those inputs are also identical to the class variable Y). More formally, as long as the identical variables are not always the same as Y, they will not satisfy the conditional independence assumption, which states that: $P(X_i, X_j \mid Y) = P(X_i \mid Y)\, P(X_j \mid Y)$. As a result, the more identical copies of a variable there are (i.e., the larger *k* is), the more the output of the classifier will come to be dominated by the value of this "single" input variable, as it is effectively being given *k* times the weight of any other single input feature.

4. Given n training examples (x^1, y^1), (x^2, y^2) … (x^n, y^n), we must calculate the following derivatives:

$$\frac{dLL(\theta)}{d\theta_1}, \frac{dLL(\theta)}{d\theta_2}, \frac{dLL(\theta)}{d\theta_3}, \frac{dLL(\theta)}{d\theta_4}$$

where

$$LL(\theta) = \sum_{i=1}^{n} y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

$$\hat{y}^{(i)} = \sigma(\theta_3 * \mathrm{g} + \theta_4 * \mathrm{h})$$

$$h^{(i)} = \sigma(\theta_2 * x^{(i)})$$

$$g^{(i)} = \sigma(\theta_1 * x^{(i)})$$

Where σ(X) is the sigmoid function.

We note the following (By the chain rule):

$\frac{dLL(\theta)}{d\theta_4} = \frac{dLL(\theta)}{d\hat{y}^{(i)}} * \frac{d\hat{y}^{(i)}}{d\theta_4}$	$\frac{dLL(\theta)}{d\theta_3} = \frac{dLL(\theta)}{d\hat{y}^{(i)}} * \frac{d\hat{y}^{(i)}}{d\theta_3}$
$\frac{dLL(\theta)}{d\theta_2} = \frac{dLL(\theta)}{d\hat{y}^{(i)}} * \frac{d\hat{y}^{(i)}}{dh^{(i)}} * \frac{dh^{(i)}}{d\theta_2}$	$\frac{dLL(\theta)}{d\theta_1} = \frac{dLL(\theta)}{d\hat{y}^{(i)}} * \frac{d\hat{y}^{(i)}}{dg^{(i)}} * \frac{dg^{(i)}}{d\theta_1}$

Therefore, we have to calculate only a few quantities and multiply them together. We also note that to handle the summation, we calculate each derivative with respect to a single piece of data, and sum up all these derivatives at the end.

$$\frac{dLL(\theta)}{d\hat{y}^{(i)}} = \frac{y^{(i)}}{\hat{y}^{(i)}} - \frac{(1 - y^{(i)})}{(1 - \hat{y}^{(i)})}$$

$$\frac{d\hat{y}^{(i)}}{d\theta_4} = \hat{y}^{(i)} * (1\text{-}\hat{y}^{(i)}) * \mathrm{h}^{(i)}$$

$$\frac{d\hat{y}^{(i)}}{d\theta_3} = \hat{y}^{(i)} * (1\text{-}\hat{y}^{(i)})*\mathrm{g}^{(i)}$$

$$\frac{d\hat{y}^{(i)}}{dh^{(i)}} = \hat{y}^{(i)} * (1\text{-}\hat{y}^{(i)})*\theta_4$$

$$\frac{d\hat{y}^{(i)}}{dg^{(i)}} = \hat{y}^{(i)} * (1\text{-}\hat{y}^{(i)})*\theta_3$$

$$\frac{dh^{(i)}}{d\theta_2} = \ \mathrm{h}^{(i)} * (1\text{-}\mathrm{h}^{(i)})* \mathrm{x}^{(i)}$$

$$\frac{dg^{(i)}}{d\theta_1} = \ \mathrm{g}^{(i)} * (1\text{-}\mathrm{g}^{(i)})* \mathrm{x}^{(i)}$$

After combining the terms together, our results are:

$$\frac{dLL(\theta)}{d\theta_4} = \sum_{i=1}^{n}(\frac{y^{(i)}}{\hat{y}^{(i)}} - \frac{(1-y^{(i)})}{(1\text{-}\hat{y}^{(i)})}) * \hat{y}^{(i)}*(1\text{-}\hat{y}^{(i)})*\mathrm{h}^{(i)} = \sum_{i=1}^{n}(y_i - \hat{y}^{(i)})*\mathrm{h}^{(i)}$$

$$\frac{dLL(\theta)}{d\theta_3} = \sum_{i=1}^{n}(\frac{y^{(i)}}{\hat{y}^{(i)}} - \frac{(1-y^{(i)})}{(1-\hat{y}^{(i)})}) * \hat{y}^{(i)}*(1\text{-}\hat{y}^{(i)})*\mathrm{g}^{(i)} = \sum_{i=1}^{n}(y_i - \hat{y}^{(i)})*\mathrm{g}^{(i)}$$

$$\frac{dLL(\theta)}{d\theta_2} = \sum_{i=1}^{n}(\frac{y^{(i)}}{\hat{y}^{(i)}} - \frac{(1-y^{(i)})}{(1-\hat{y}^{(i)})}) * \hat{y}^{(i)}*(1\text{-}\hat{y}^{(i)})*\theta_4 * \mathrm{h}^{(i)}*(1\text{-}\mathrm{h}^{(i)})*\mathrm{x}^{(i)}$$

$$= \sum_{i=1}^{n}(y_i - \hat{y}^{(i)}) * \theta_4 * \mathrm{h}^{(i)} \ * \ (1\text{- }\mathrm{h}^{(i)}) \ * \ \mathrm{x}^{(i)}$$

$$\frac{dLL(\theta)}{d\theta_1} = \sum_{i=1}^{n}(\frac{y^{(i)}}{\hat{y}^{(i)}} - \frac{(1-y^{(i)})}{(1-\hat{y}^{(i)})}) * \hat{y}^{(i)}*(1\text{-}\hat{y}^{(i)})*\theta_3 * g^{(i)}*(1\text{-}\mathrm{g}^{(i)})*\mathrm{x}^{(i)}$$

$$= \sum_{i=1}^{n}(y_i - \hat{y}^{(i)}) * \theta_3 * g^{(i)} \ * \ (1\text{- }\mathrm{g}^{(i)}) \ * \ \mathrm{x}^{(i)}$$

Whew! See why neural nets are complicated?

Note: solutions to programming assignments are not distributed.

Coding 1: Naïve Bayes

a. Using either MLE or Laplace estimators we achieved 100% accuracy on simple-test

b.

Note: In the Netflix dataset, since there were so many training movies, MLE and Laplace had identical parameters up to two decimal places. As a result, all answers are the same for the two classifiers.

i. P(Y = 1) = 0.635

ii. P(Xi = 1|Y = 1), read from left to right:

0.67	0.67	0.59	0.60	0.72
0.61	0.60	0.70	0.86	0.66
0.68	0.66	0.88	0.67	0.51
0.52	0.75	0.71	0.75	

iii. P(Xi = 1|Y = 0), read from left to right:

0.59	0.68	0.68	0.66	0.57
0.60	0.62	0.60	0.86	0.70
0.66	0.64	0.88	0.67	0.57
0.56	0.76	0.57	0.31	

iv. Class 0: tested 265 , correct classified 185
Class 1: tested 235 , correct classified 180
Overall: tested 500 , correct classified 365
Accuracy: 0.730

c.

i. We were looking for you to chose the movies that maximized the ratio: $P(Y = 1| X_i = 1) / P(Y = 1| X_i = 0)$. Use Bayes Theorem to find these probabilities. The five highest movies were:

Index	Movie	Ratio
19	When Harry Met Sally	2.39
18	What Women Want	1.25
1	3 Idiots	1.15
5	How to Lose a Guy in 10 Days	1.27
8	La Vita E Bella	1.16

ii. One of the test examples that was classified incorrectly was (left redacted). The log likelihoods for Y = 0 and Y = 1 were -11.82 and -11.83, respectively. These values are very similar, which indicates that the classifier was close to making the correct decision. Despite the fact that the user liked the the most indicative movie (19) they happen to have not liked the target movie. It is more likely that we don't have enough features to make all predictions correctly.

d.

MLE:
Class 0: tested 109 , correct classified 90
Class 1: tested 75 , correct classified 56
Overall: tested 184 , correct classified 146
Accuracy: 0.79

Laplace:
Class 0: tested 109 , correct classified 89
Class 1: tested 75 , correct classified 55
Overall: tested 184 , correct classified 144
Accuracy: 0.78

e.

MLE:
Class 0: tested 15 , correct classified 10
Class 1: tested 172 , correct classified 135
Overall: tested 187 , correct classified 145
Accuracy: 0.78

Laplace:
Class 0: tested 15 , correct classified 10
Class 1: tested 172 , correct classified 130
Overall: tested 187 , correct classified 140
Accuracy: 0.75

Coding 2: Logistic Regression

a. We were able to achieve 100% accuracy.

b.

i. Parameter weights after training (index from 0 to 14, from left to right):

-1.26	0.15	-0.03	-0.19	-0.11
0.33	0.03	-0.11	0.22	-0.04
-0.08	0.08	0.05	0.02	0.00
-0.11	0.01	-0.02	0.28	1.76

Note that the first parameter corresponds to the offset

ii. Classification accuracy:
Class 0: tested 265 , correct classified 180
Class 1: tested 235 , correct classified 190
Overall: tested 500 , correct classified 370
Accuracy: 0.74

iii. If a user likes movies
#19 (When Harry Met Sally),
#18 (What Women Want),
#1 (3 Idiots),
#5 (How to Lose A Guy in 10 Days), and
#8 (La Vita E Bella),
Then they are more likely to like the target movie (Love Actually). This is similar to what was found for Naive Bayes above and fits with our intuition about genres.

iv. Initial Log Likelihood of training data when parameters are all 0: -3119

v. Log likelihood of training data after training: -2627

c. Results for ancestry-test.txt:
Class 0: tested 109 , correct classified 98
Class 1: tested 75 , correct classified 56
Overall: tested 184 , correct classified 154
Accuracy: 0.84

d. Here are a few examples of learning rates and the respective accuracy:
Learning rate = 1e-4: Accuracy: 0.76
Learning rate = 2e-5: Accuracy: 0.79
Learning rate = 5e-4: Accuracy: 0.73
Learning rate = 4e-6: Accuracy: 0.88
Learning rate = 1e-5: Accuracy: 0.81
Learning rate = 5e-7: Accuracy: 0.94
Learning rate = 1e-7: Accuracy: 0.94
Learning rate = 1e-6: Accuracy: 0.94