COMS 4771 Machine Learning 2023 Spring Homework 1

Danrui Wang - dw3031@columbia.edu

2/14/2023

1 Analyzing Bayes Classifier

(i)(a) A and B are known

Since A, B, and C are exponential random variables with $\lambda = 1$.

$$\begin{aligned} P[Y=1|A,B] &= P[A+B+C<7|A,B] \\ &= P[C<7-A-B] \\ &= \int_0^{7-A-B}(1-e^{-x})dx \\ &= 1-e^{A+B-7} \\ P[Y=0|A,B] &= 1-P[Y=1|A,B] \\ &= e^{A+B-7} \end{aligned}$$

The Bayes optimal classifier would output 1 if

$$\begin{aligned} P[Y=1|A,B] &> P[Y=0|A,B] \\ 1-e^{A+B-7} &> e^{A+B-7} \\ e^{A+B-7} &< \frac{1}{2} \\ A+B-7 &< ln\frac{1}{2} \\ A+B &< 7+ln\frac{1}{2} \end{aligned}$$

and output 0 otherwise, which indicates

$$f(A,B) = \begin{cases} 1 & \text{if } A+B<7+ln\frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

Since A and B are both exponential random variables with both parameter = 1, A+B $\sim$ Erlang(2,1). let x = A+B, we get

$$\begin{gathered} f(x) = xe^{-x} \\ P[Y=1|X=x] = P[Y=1|A+B] = 1-e^{A+B-7} = 1-e^{x-7} \\ P[Y=0|X=x] = P[Y=0|A+B] = e^{A+B-7} = e^{x-7} \end{gathered}$$

$$\begin{aligned} \textbf{Bayes Error} &= E[1[f(x)\neq y]] \\ &= P[f(x)=1, Y=0|X=x] + P[f(x)=0, Y=1|X=x] \\ &= E[1[f(x)=1]] \times P(Y=0|X=x) + E[1[f(x)=0]] \times P(Y=1|X=x) \\ &= \int_0^{7+ln\frac{1}{2}} (xe^{-x}x^{x-7})dx + \int_{7+ln\frac{1}{2}}^{7} (xe^{-x}(1-x^{x-7}))dx \\ &\approx 0.0181357 + 0.0018255 \\ &\approx 0.0199612 \\ &\approx 0.02 \end{aligned}$$

(i)(b)A is known

Since A, B and C are all i.i.d exponential random variable ($\lambda = 1$), if we let B+C = x, x ∼ Erlang(2,1). The CDF of x is

$$F(x) = \int_0^x f(B+C)dx = \int_0^x (te^{-t})dt = 1 - e^{-x} - xe^{-x}$$

If A is known,

$$\begin{aligned} P[Y=1|A] &= P[A+B+C<7|A] \\ &= P[0<B+C<7-A|A] \\ &= F(7-A) - F(0) \\ &= 1 - e^{A-7} - (7-A)e^{A-7} \\ &= 1 + (A-8)e^{A-7} \\ P[Y=0|A] &= 1 - P[Y=1|A] \\ &= (8-A)e^{A-7} \end{aligned}$$

The Bayes Optimal classifier would output 1 if

$$\begin{gathered} P[Y=1|A] > P[Y=0|A] \\ 1 + (A-8)e^{A-7} > (8-A)e^{A-7} \\ A < 5.32165 \\ \approx A < 5.32 \end{gathered}$$

and output 0 otherwise, which indicates

$$f(A) = \begin{cases} 1 & \text{if } A < 5.32 \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} \textbf{Bayes Error} &= E[1[f(A) \neq y]] \\ &= E[1[f(A)=1]] \times P(Y=0|A) + E[1[f(A)=0]] \times P(Y=1|A) \\ &= \int_0^{5.32} e^{-x}(8-A)e^{A-7}dx + \int_{5.32}^{7} e^{-x}(1+(A-8)e^{A-7})dx \\ &\approx 0.026 \end{aligned}$$

(i)(c) None of A,B,C are known

A+B+C ∼ Gamma(3,1)

$$\begin{aligned} P(Y=1) &= P(0 < A+B+C < 7) \\ &= F(7) - F(0) \\ &= 0.97036 \\ P(Y=0) &= 1 - P(Y=1) \\ &= 0.02964 \end{aligned}$$

Since Bayes optimal classifier always outputs 1.

$$\begin{aligned} \textbf{Bayes Error} &= 1 - E[1[f(x)=y]] \\ &= 1 - 0.97036 \\ &= 0.02964 \end{aligned}$$

(ii)

From the examples above, we can see that Bayes error depends on both the distribution of the known and unknown variables. Choosing different distribution on C would definitely change the error rate. if A and B are known, there exists a range that the Bayes classification might make mistakes. If we let C to be uniform, it could make A+B+C>7 very likely to A+B+C<7.

2 Classification with Asymmetric Costs

2.1

When we want to predict whether there will be an earthquake or not, a false negative is more serious than false positive. with classification with asymmetric costs, we can set more cost on false negative (p) and lower on false positive (q).

2.2

3 Finding (local) minima of generic functions

3.1

Since for all a, b $\in \mathbb{R}$, $|f'(a) - f'(b)| \leq L^2|a - b|$, for some L $\geq$ 0 if a = x+h, b = x, for x+h, x$\in \mathbb{R}$, h $\rightarrow$ 0

$$\lim_{h\to 0} |f'(x+h) - f'(x)| \leq L^2|x+h-x|$$
$$\lim_{h\to 0} \frac{|f'(x+h) - f'(x)|}{|h|} \leq L^2$$
$$f''(x) \leq L^2$$

Since x is just a random variable $\in \mathbb{R}$, we can generalize the result as $f''(z) \leq L^2$ for all z. Thus, we've shown that the assumption implies that f has bounded second derivative.

Then we use the Taylor's Remainder Theorem $f(b) = f(a) + f'(a)(b-a) + \frac{1}{2}f''(z)(b-a)^2$ let a = x, b = $\bar{x}$ since $\bar{x} = x - \eta f'(x)$, $f'(x) = \frac{x-\bar{x}}{\eta}$

$$f(\bar{x}) = f(x) + f'(x)(\bar{x} - x) + \frac{1}{2}f''(z)(\bar{x} - x)^2$$
$$f(\bar{x}) = f(x) - \frac{1}{\eta}(\bar{x} - x)^2 + \frac{1}{2}f''(z)(\bar{x} - x)^2$$

Since $f''(z) \leq L^2$,

$$f(\bar{x}) \leq f(x) - \frac{(\bar{x} - x)^2}{\eta} + \frac{(\bar{x} - x)^2}{2}L^2$$
$$f(\bar{x}) \leq f(x) - (\bar{x} - x)^2(\frac{1}{\eta} - \frac{L^2}{2})$$

Since we only need to prove there exists some $\eta > 0$, we can make $\eta \leq \frac{2}{L^2}$, so that $\frac{1}{\eta} - \frac{L^2}{2} \geq 0$ Because $(\bar{x} - x)^2 \geq 0$, we are subtracting a number that ≥ 0 from $f(x)$. This makes

$$f(\bar{x}) \leq f(x) \tag{1}$$

If we go back to Taylor's Remainder Theorem, we can see the only way to let $f(\bar{x}) = f(x)$ is to let $\bar{x} = x$, which means that $f'(x) = 0$ by definition $\bar{x} = x - \eta f'(x)$

Thus, we can say for any $x \in \mathbb{R}$, there exists some $\eta > 0$, such that if $\bar{x} = x - \eta f'(x)$, then $f(\bar{x}) \leq f(x)$, with equality if and only if $f'(x) = 0$.

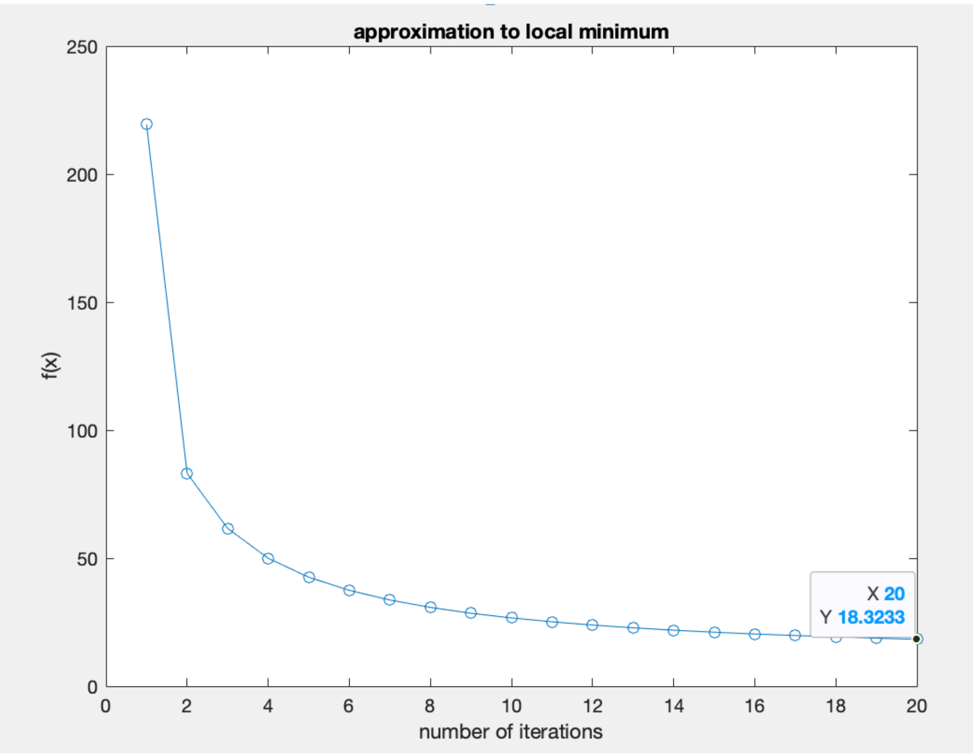
3.2

As proved $f''(z) \leq L^2$ above, $\frac{2}{L^2} \leq \frac{2}{f''(z)}$ Since we've made the assumption about η , $0 < \eta \leq \frac{2}{L^2} \leq \frac{2}{f''(z)}$, I let $\eta = \frac{1}{f''(x)}$ in each iteration. If we start from x_0, the process looks like:

$$\begin{aligned} x_1 &= x_0 - \frac{f'(x_0)}{f''(x_0)} \\ x_2 &= x_1 - \frac{f'(x_1)}{f''(x_1)} \\ x_3 &= x_2 - \frac{f'(x_2)}{f''(x_2)} \\ x_{n-1} &= x_{n-2} - \frac{f'(x_{n-2})}{f''(x_{n-2})} \end{aligned}$$

Ideally iterate throught th process until we find the local minimum at x_n when $x_n = x_{n-1}$

3.3



start from x0=2. After 20 iterations, the minimum is approximated to 18.3233.

3.4

Because as the definition of $\bar{x} = x - \eta f'(x)$, $\bar{x}$ always goes to the direction that has lower $f(\bar{x})$ than $f(x)$ as $f'(x)$ represents whether the function is increasing or decreasing at x. Also, we've proved that $f(\bar{x}) \leq f(x)$, which only equals to each other when the minimum is found ($\bar{x} = x$). This indicates that we can iterate $\eta f'(x)$ step a time to find the minimum. However, we can only say this minimum as the local minima; as the algorithm stops as $\bar{x} = x$, we are not looking for other possible minima that might be lower than our found local minima once we found one. Thus, this technique is only useful for finding local minima.

In order to help find global minima, we can randomly start at multiple points and iterate through the functions, and compare the value of each local minima found. If we start at more separated points, we might find the global minima. Also, we can also set η to different values, and find the one that works the best.

4 Exploring the Limits Current language Models

(i)

$$
\begin{aligned}
P_{Laplace}(w_i|w_{i-2}w_{i-1}) &\underset{approx.}{=} \frac{C(W_{i-2}w_{i-1}w_i)+1}{C(W_{i-2}w_{i-1})+|V|} \\
P(w_{1:n}) = \prod_{i=1}^{n}(P(w_i|w_{i-2}w_{i-1}) &\underset{approx.}{=} \prod_{i=1}^{n}\frac{C(W_{i-2}w_{i-1}w_i)+1}{C(W_{i-2}w_{i-1})+|V|} \\
P(y|w_{1:n}) = \frac{P(w_{1:n}|y)P(y)}{P(w_{1:n})} &\underset{approx.}{=} \frac{P(w_{1:n}|y)P(y)}{\prod_{i=1}^{n}\frac{C(W_{i-2}w_{i-1}w_i)+1}{C(W_{i-2}w_{i-1})+|V|}}
\end{aligned}
$$

(ii)(b)

oov rate for hum bigram = 0.18341131575504704
oov rate for hum trigram = 0.5303352084371321
oov rate for gpt bigram = 0.12031217505834509 oov rate for gpt trigram = 0.3854559003627793

(ii)(c)

The accuracy of bigram is around 0.7067 whereas of trigram is around 0.8456. The OOV rate for bigram is generally lower than trigram. Because bigram is shorter and most of the bigrams are already seen in the training set, this makes the classes looks very similar which makes it hard to classify. Whereas the oov rate for trigram is higher and trigram is longer, these trigrams are more unique which makes it easier to classify which class it belongs to.

(iii)(a)

(iii)(b)

Sentences generated from n-gram models often fail to capture long-range context because these models rely only on the preceding n-1 tokens to predict the next token, without considering the global context of the entire sequence. As a result, n-gram models are limited in their ability to capture long-range dependencies in the text, which can lead to generated text that is less coherent and less grammatically correct than text generated by more sophisticated models that use attention mechanisms.

Attention mechanisms, on the other hand, allow models to attend to all tokens in the input sequence, giving the model a much richer understanding of the context in which each token appears. This allows the model to capture long-range dependencies and semantic relationships between words, resulting in more coherent and natural-sounding generated text.

Also, if we want ngram working like attention transformer, we need to keep track of different ngrams with n in different value. This might need a lot of memory space to keep track of the sentences or even the whole paragraph, which makes it not applicable.