

CS769 Advanced Natural Language Processing
Homework 2
Reference Answer

February 18, 2010

Answer 1.1

There are 70 distinct ASCII printable characters in the file alice.txt.

Note: if you think whitespace to be an ASCII printable character, then you will have 71 distinct ASCII printable characters. The following answers for Problem 1 are based on 70 distinct ASCII printable characters, but either 70 or 71 is considered to be correct during grading.

Answer 1.2

The characters with the corresponding MLE θ are:

e	0.115379	.	0.008429	”	0.000974
t	0.088049	v	0.006929	L	0.000845
a	0.070269	I	0.006318	B	0.000784
o	0.068692	-	0.005766	Q	0.000724
h	0.061107	A	0.005499	G	0.000715
n	0.059435	T	0.004068	K	0.000707
i	0.058444	!	0.003878	z	0.000664
s	0.054143	H	0.002448	F	0.000638
r	0.045662	W	0.002043	U	0.000569
d	0.040844	:	0.002008	P	0.000560
l	0.039801	S	0.001879	*	0.000517
u	0.029321	?	0.001741	(0.000483
g	0.021099	M	0.001724)	0.000483
w	0.021013	;	0.001672	V	0.000362
,	0.020840	D	0.001655	J	0.000069
c	0.019427	E	0.001620	X	0.000034
y	0.018513	O	0.001517	-	0.000034
f	0.016608	C	0.001250	[0.000017
m	0.016436	x	0.001241]	0.000017
,	0.015186	R	0.001207	0	0.000009
p	0.012575	j	0.001189	3	0.000009
b	0.011928	q	0.001077	Z	0.000009
‘	0.009558	N	0.001034		
k	0.009274	Y	0.000983		

Answer 1.3

The likelihood $P(D|\theta) = 1.2507 \times 10^{-111}$.

Since $P(D|\theta) = \binom{N}{n_1 \dots n_K} \prod_{k=1}^K \theta_k^{n_k}$, one issue is overflow when computing the combinatorial number $\binom{N}{n_1 \dots n_K} = \frac{N!}{n_1! \dots n_K!}$. There are 116,026 printable characters in that file, i.e. $N = 116,026$, $N!$ is definitely overflow the maximum number in Matlab. Actually, the *factorial* function provided by Matlab can only be accurate for $N \leq 21$. The solution is to use log likelihood. Another issue is underflow when compute $\prod_{k=1}^K \theta_k^{n_k}$, the solution is also sum over $n_k \log(\theta_k)$ instead of compute the production.

Answer 1.4

$$\begin{aligned} \|\theta_{10} - \theta_{100}\| &= 0.2724 \\ \|\theta_{100} - \theta_{1000}\| &= 0.2272 \\ \|\theta_{1000} - \theta_{10000}\| &= 0.0393 \\ \|\theta_{10000} - \theta_{100000}\| &= 0.0165 \\ \|\theta_{100000} - \theta_{144943}\| &= 0.0024 \end{aligned}$$

The Euclidean distance between adjacent θ pairs is decreasing in the number of printable characters counted. This means that when we get a corpus with large enough size, the estimation of θ will converge to some constant value.

Answer 2.1

The probability that a monkey types a word with length i is

$$P(i) = p^i(1 - 2p)$$

The longer word has the lower probability and is expected appear fewer in the corpus. So if we rank all words by their probabilities in decreasing order, the longer word is expected to be after the shorter word. The number of words with length i is 2^i . So the rank r_i of a word with length i satisfies

$$\sum_{j=1}^{i-1} 2^j < r_i \leq \sum_{j=1}^i 2^j$$

For those ranks $r = \sum_{j=1}^i 2^j = 2(2^i - 1)$, its length $i = \log_2(\frac{r}{2} + 1)$.

For those ranks $\sum_{j=1}^{i-1} 2^j < r < \sum_{j=1}^i 2^j$, $i \approx \log_2(\frac{r}{2} + 1)$.

The frequency of this word in a corpus with N words is

$$\begin{aligned} p(i) &= p^{\log_2(\frac{r}{2}+1)}(1 - 2p) \\ &= \left(\frac{r}{2} + 1\right)^{\log_2 p}(1 - 2p) \\ &= (p^{-1} - 2)(r + 2)^{\log_2 p} \end{aligned}$$

Answer 2.2

The likelihood function is

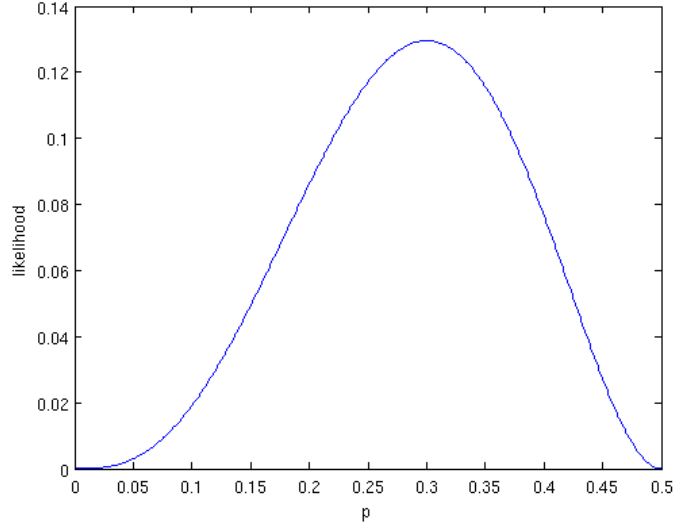
$$L(p) = \binom{5}{2, 1, 2} \cdot p^2 \cdot p^1 \cdot (1 - 2p)^2 = 30p^3(1 - 2p)^2$$

The function curve is on the next page.

Answer 2.3

Since Dirichlet prior is the conjugate prior of multinomial distribution, the posterior distribution is also a Dirichlet distribution $p \text{Dir}(2.2, 1.3, 2.5)$. So the mean of posterior distribution is $\frac{1}{6}(2.2, 1.3, 2.5) = (0.3667, 0.2167, 0.4167)$ and mode is $(0.4, 0.1, 0.5)$.

Answer 3.1



Denote the K possible outcomes for x is $\{x_1, x_2, \dots, x_K\}$. The maximum element of θ is θ_{max} . Let the guess is y , so the probability that the guess is correct is $p(x = y)$.

Strategy 1 $p_1(x = y) = p(x = x^*) = \theta_{max}$.

Strategy 2

$$p_2(x = y) = \sum_{i=1}^K p(x = x_i | y = x_i) p(y = x_i) = \sum_{i=1}^K p(x = x_i) p(y = x_i) = \sum_{i=1}^K \theta_i^2$$

Since

$$\sum_{i=1}^K \theta_i^2 \leq \sum_{i=1}^K \theta_i \cdot \theta_{max} = \theta_{max} \sum_{i=1}^K \theta_i = \theta_{max}$$

Strategy 1 is the better choice. When $\theta_i = \frac{1}{K}, i = 1, 2, \dots, K$, both strategies have the same accuracy.

Answer 3.2

Strategy 1 $p_1(x = y) = p(x = x^*) = \phi_{x^*}$

Strategy 2

$$p_2(x = y) = \sum_{i=1}^K p(x = x_i | y = x_i) p(y = x_i) = \sum_{i=1}^K p(x = x_i) p(y = x_i) = \sum_{i=1}^K \theta_i \cdot \phi_i$$

The relative relation between $p_1(x = y)$ and $p_2(x = y)$ is determined by θ and ϕ . If $\arg \max_x \theta_x = \arg \max_x \phi_x$, then Strategy 1 is still better. Otherwise, the relation is not clear.