Probability theory: practicum problems

LING83800

1 Maximum likelihood estimation: part 1

Problem Compute the maximum likelihood distribution for all possible rolls of a single ("fair") 6-sided die (or a `d6` in tabletop gaming dice notation).

2 Maximum likelihood estimation: part 2

Problem Estimate the maximum likelihood distribution for adding the values of two ("fair") 6-sided dice (`2d6`). For instance, the roll $\{2, 3\}$ gives the value $2 + 3 = 5$.

Hint Make a 6×6 table giving the sum of the dice after each of the 36 equiprobable rolls:

	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

Then, count the number of times each sum appears in the table to get the numerator.

3 Language modeling, part 1

Problem Using the chain rule, write out the probability for the sentence *colorless green ideas sleep furiously.* Then, write out the formula using a second-order Markov approximation.

Hint The formulas are a product of several probabilities. You haven't been told possible values for $P(\text{colorless})$, etc., so you can only give formulas, not an actual probability.

4 Language modeling, part 2

Problem Daft Punk's 2001 single "Harder, Better, Faster, Stronger" has the following lyrics (repeated several times):

```
work it make it do it makes us
harder better faster strong
work it harder make it better
do it faster makes us stronger
more than ever hour after hour
work is never over
```

1. Compute the maximum likelihood probability of the phrase *work is never over* using a first-order Markov approximation.
2. Compute the maximum likelihood probability of the phrase *work it harder make it better* using a second-order Markov approximation.

5 Avoiding underflow

Problem Use negative logarithms to compute $.00002 \cdot .3$.

Hint You can use a calculator or Python for this.

6 Bayes' rule

Problem A *hidden Markov model* part-of-speech tagger is conceptually quite similar to a speech recognizer as described in the lecture. Let $\mathbf{W} = w_1, \ldots, w_n$ be a sequence of words and $\mathbf{T} = t_1, \ldots, t_n$ be the corresponding sequence of tags. Our goal in tagging is to compute:

$$\hat{\mathbf{T}} = \arg\max_{\mathbf{T}} P(\mathbf{T} \mid \mathbf{W})$$

where $P(\mathbf{T} \mid \mathbf{W})$ is the probability of the tag sequence given the observed word sequence. Use Bayes' rule to rewrite the equation for $\hat{\mathbf{T}}$ in terms of $P(\mathbf{W} \mid \mathbf{T})$ instead.