Exercise Sheet II

Submission Deadline: May 10th, 23:59

1 Probability Theory Review

Exercise 1.1: Probabilistic Independence (2 points)

Consider a fair 6-sided die whose sides are numbered from 1 to 6 and each die roll is independent of the other rolls. In an experiment that consists of rolling the die twice, the following events can be defined

A : The sum of the two outcomes is strictly larger than 10
B : At least one of the two rolls resulted in 6
C : At least one of the two rolls resulted in 3
D : The outcome of the 2nd roll was higher than the 1st roll
E : The difference between the two roll outcomes is exactly 1

(a) Compute the probabilities $\mathbf{P}(A)$, $\mathbf{P}(C)$, and $\mathbf{P}(E)$.

(b) Is event A independent of event B?

(c) Is event A independent of event C?

(d) Are events D and E independent?

Justify your answers using the laws of probability and the definition of probabilistic independence.

Exercise 1.2: Communication through a Noisy Channel (8 points)

A binary (0 or 1) message transmitted through a noisy communication channel is received incorrectly with probability ϵ_0 and ϵ_1, respectively (see the figure). At each time step, the channel source transmits a 0 with probability p and transmits a 1 with probability $1 - p$. Assume that symbols are received in the same temporal order as they were transmitted and errors in different symbol transmissions are independent.

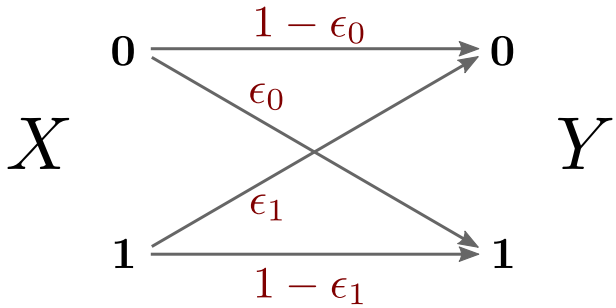

Figure: A schematic of a binary communication channel with error probabilities.

(a) Let X be a random variable over the source symbols $\{0, 1\}$. Express the source probabilities as a probability mass function of X.

(b) Let Y be a random variable over the receiver symbols $\{0, 1\}$. Express the channel's error probabilities as a conditional probability mass function of Y given X.

(c) What is the probability of transmitting the symbol sequence 1001110?

(d) What is the probability that a randomly chosen symbol is received correctly?

(e) Suppose that the symbol sequence 1011 is transmitted. What is the probability that all the symbols in the sequence are received correctly?

(f) What is the probability of receiving the symbol sequence 1101?

(g) The reliability of the communication can be improved by introducing some redundancy in the channel source. Therefore, each symbol is transmitted three times and the received symbol is decoded by majority rule. In other words, a 0 (or 1) is transmitted as 000 (or 111, respectively), and it is decoded at the receiver as a 0 (or 1) if and only if the received three-symbol sequence contains at least two 0s (or 1s, respectively). What is the probability that a transmitted 0 is correctly decoded?

(h) Suppose that the scheme of part (g) is used. What is the probability that a 0 was transmitted given that the received string is 101? (*Hint: use Bayes' rule*)

(i) Would the noisy channel be a good model of human communication with natural languages? Explain why or why not in a few sentences.

In this exercise you are not required to perform any numerical computations. The final answers are mathematical terms of $p, \epsilon_0,$ and, ϵ_1, not numerical values.

2 Character N-grams and Entropy

An n-gram is a sequence of n consecutive symbols. When the symbols are characters in a linear text, a character n-gram is simply a sequence of n adjacent characters. In this exercise, you will investigate character n-gram distributions in two languages: English and German. Download the file `exercise2_corpora.zip` from the course materials and unpack it in a separate directory. For both corpora, lower-case the text and keep only the characters of the English alphabet set $\{\mathrm{a} - \mathrm{z}\}$ for the English corpus, and the characters of the German alphabet set $\{\mathrm{a} - \mathrm{z}, \ddot{\mathrm{a}}, \ddot{\mathrm{o}}, \ddot{\mathrm{u}}, \ss\}$ for the German corpus. Apply white-space tokenization for both corpora.

Exercise 2.1: N-grams Frequency Analysis (2 points)

(a) Implement a Python function `char_ngrams` that takes as an input a word token and an integer m and returns a list of all possible n-grams in the token from $n = 1$ up to $n = m$. The function should behave as follows

```
> char_ngrams('hier', 4)
['hier', 'hie', 'ier', 'hi', 'ie', 'er', 'h', 'i', 'e', 'r']
```

and ...

```
> char_ngrams('und', 5)
['und', 'un', 'nd', 'u', 'n', 'd']
```

Use this function to convert each text corpus into a linear sequence of character n-grams (up to $n = 4$). Collect n-gram frequency counts in a suitable data structure, preferably a `collections.Counter` object.

(b) Generate a table of the top 15 frequent character unigrams (1-grams), bigrams (2-grams), trigrams (3-grams), and 4-grams for each text corpus.

(c) Assume that the identity of the languages and the original word forms in the two corpora are unknown, would you be able to identify the language based on the distribution of frequent n-grams? Justify your answer.

Your solution should contain the source code, a single table for the frequent n-grams in both languages, and the observations you want to include.

Exercise 2.2: N-gram Probability Distributions (8 points)

(a) Given a sequence of preceding characters (or history h), the probability that a specific character c will occur next can be estimated as follows

$$\mathbf{P}(C = c|h) = \mathbf{P}(C = c|c_1, \ldots, c_k) = \frac{N(c_1, \ldots, c_k, c)}{\sum_{s \in S} N(c_1, \ldots, c_k, s)}$$

where $N(c_1, \ldots, c_m)$ is the observed frequency of the n-gram $\langle c_1, \ldots, c_m \rangle$ in the corpus, and S is the alphabet set of the language. For a specific history h, the conditional distribution defined by the probability function above should be a proper probability distribution over the alphabet. More precisely

$$\sum_{s \in S} \mathbf{P}(C = s|h) = 1$$

Implement a Python function `prob_dist` that takes as an input a data structure for n-gram counts $N(.)$, an alphabet S, and a specific history h, and returns as an output a suitable data structure that represents a probability distribution over the symbols of the alphabet, preferably a `collections.defaultdict` object. Write a few test cases with Python's `assert()` function to validate whether probability mass of the distribution for a few arbitrary histories sum to 1 with numerical precision of 1×10^{-7}.

(b) With the German n-gram counts collected in Exercise 2.1 (a), use the function in part (a) to estimate the distributions $\mathbf{P}(C|h = \text{`'})$, $\mathbf{P}(C|h = \text{`n'})$, $\mathbf{P}(C|h = \text{`un'})$, and $\mathbf{P}(C|h = \text{`gun'})$. Plot each of these distributions as a bar chart where the x-axis represents the characters (in the order of the alphabet) and the y-axis represents the probability value. How does the probability distribution change as the history gets longer? Write your observations.

(c) Given a probability distribution over an alphabet S, the expected value of the function $-log\mathbf{P}(C|h)$ is defined as follows

$$\mathbf{E}\Big[-log\mathbf{P}(C|h)\Big] = -\sum_{s \in S} \mathbf{P}(C = s|h) \times log\mathbf{P}(C = s|h)$$

The quantity $\mathbf{E}\big[-log\mathbf{P}(C|h)\big]$ is referred to as the entropy of the distribution $\mathbf{P}(C = s|h)$, which measures the uncertainty or the 'disorder' within the distribution. Implement a Python function `entropy` that takes as an input a data structure of a probability distribution (i.e., the output of `prob_dist` function in part (a)) and an alphabet S and returns as an output the entropy of the distribution in bits. Use this function to compute the entropy of the distributions in part (b) and produce a table of the results. What do you observe?

(d) What is the probability distribution that would give the maximum entropy for the German text in this exercise? What is the entropy of this distribution?.

(e) Repeat part (b) but this time estimate and plot the bigram distributions $\mathbf{P}(C|h = \text{`a'})$, $\mathbf{P}(C|h = \text{`d'})$, $\mathbf{P}(C|h = \text{`z'})$, and $\mathbf{P}(C|h = \text{`c'})$. Compute the entropy of each distribution and briefly explain your observations.

Your solution should contain the figures for both part (b) and (e), two entropy tables of (c) and (e), the source code to reproduce the results, as well as the discussions you want to include.

Exercise 2.3: Cryptographic Analysis (extra credits)

Given what you have learned about the n-gram distributions of English and German, identify the language of the original text and crack the following substitution cipher

```
KOTP OX PSG AIOQVGKT OX PSG WOIVC TPGK XIOK VZBYEZTPZJ
KZTPNHGT NBC TZKAVG KZTEBCGITPNBCZBYT. COB'P GFGI PNHG WOICT
NP XNJG FNVEG. WSGB UOE TPGA ZBPO PSG ROBG OX VOFG, VNBYENYG
NT WG HBOW ZP QGJOKGT OQTOVGPG. PSNP WSZJS JNBBOP QG AEP ZBPO
WOICT JNB OBVU QG YINTAGC PSIOEYS TZVGBJG.
```

The original text is a lower-cased excerpt from contemporary literary fiction. Punctuation marks and spaces are not substituted. To get points for this exercise, submit the substitution key, the source code you developed to crack the ciphertext, and explain how you approached the problem.

Submission Instructions

> The following instructions are mandatory. Please read them carefully. If you do not follow these instructions, the tutors can decide not to correct your exercise solutions.

- You have to submit the solutions of this exercise sheet as a team of 2-3 students.
- NLTK modules are not allowed, and not necessary, for this assignment.
- You do not need to include the distributed corpora within your submission.
- Make a single `ZIP` archive file of your solution with the following structure
  - A `source_code` directory that contains your well-documented source code and a `README` file with instructions to run the code and reproduce the results.
  - A `PDF` report with your solutions, figures, and discussions on the questions that you would like to include.
  - A `README` file with group member names, matriculation numbers and emails.
- Rename your `ZIP` submission file in the format

  `exercise02_id#1_id#2_id#3.zip`

  where `id#n` is the matriculation number of every member in the team.
- Your exercise solution must be uploaded by only one of your team members to the course management system (CMS). Once the grading is done, your assignment grade will be distributed to each team member by one of the tutors.
- If you have any problems with the submission, contact `babdullah@lsv.uni-saarland.de` before the deadline.