Assignment 1. Foundations

Prof. Hwanjo Yu
CSED342 - Artificial Intelligence

Contact: Kyumin Lee (qmin2@postech.ac.kr)
Deadline: March 3, 2025, at 23:59 (50% penalty per day of delay)

General Instructions

Assignment 1 consists of five problems where Problems 1-4 are written and Problem 5 is a programming task, with each problem's score indicated alongside it. For the written problems, please submit your answers and solutions in PDF format. You may create the PDF by converting from Overleaf or by taking clear, legible photos of your handwritten solutions and combining them into a single PDF; any issues arising from illegible handwriting are solely your responsibility.

For the programming problem, please note that the assignment has been developed in Python 3.12, so please use **Python 3.12** to implement your code. We recommend using **Conda environment**.[1]. You must submit the programming file as `submission.py` without changing the file name.

Ultimately, your final submission must be a {YOUR_STUDENT_ID}.zip file that contains both {YOUR_STUDENT_ID}.pdf (your written solution) and `submission.py`.

When working on your code, please make your modifications only within the designated section in `submission.py`—that is, between the markers

```
# BEGIN_YOUR_ANSWER
```

and

```
# END_YOUR_ANSWER
```

You can add other helper functions outside the answer block if you want, but do not import other libraries and do not make changes to files other than `submission.py`. You can solve all the problems without any libraries other than *collections* and *math*.

[1] https://docs.conda.io/projects/conda/en/stable/user-guide/install/index.html

Your code will be evaluated on two types of test cases, **basic** and **hidden**, which you can see in `grader.py`. Basic tests, which are fully provided, do not stress your code with large inputs or tricky corner cases. Hidden tests are more complex and do stress your code. The inputs of hidden tests are provided in `grader.py`, but the correct outputs are not. To run all the tests, type

```
python grader.py
```

This will tell you only whether you passed the basic tests. The script will alert you if your code takes too long or crashes on the hidden tests, but does not say whether you got the correct output. You can also run a single test (e.g., `2a-0-basic`) by typing

```
python grader.py 2a-0-basic
```

We strongly encourage you to read and understand the test cases, create your own test cases, and not just blindly run `grader.py`.

Problem 1 [10 points]

You repeatedly roll a fair six-sided die until you get a 1 (in which case you stop).

- Each time you roll a 3, you *gain* a points.
- Each time you roll a 6, you *lose* b points.
- Rolls of 2, 4, or 5 neither add nor subtract points.

What is the expected number of points you have at the time you stop?

Problem 2 [10 points]

We flip a coin 7 times with an unknown bias p (the probability of landing heads). Suppose we observe the sequence $\{T, H, H, H, H, H, H\}$, which contains 6 heads and 1 tail. The likelihood for observing exactly 6 heads (and thus 1 tail) is

$$L(p) \;=\; p^6\,(1-p).$$

Find the value of p that maximizes $L(p)$, prove that this is indeed a maximum, and give an intuitive explanation for the result.

Hint: Instead of differentiating the likelihood, work with the log-likelihood function

$$\ell(p) = \ln L(p) = 6\ln(p) + \ln(1-p).$$

Problem 3 [10 points]

Let A and B be events such that $P\big(A \mid B\big) = P\big(B \mid A\big)$, with $P(A \cup B) = \frac{1}{3}$ and $P(A \cap B) > 0$. Show that $P(A) > \frac{1}{6}$.

Problem 4 [10 points]

Let's practice taking gradients, which is a key operation for being able to optimize continuous functions. For $\mathbf{w} \in \mathbb{R}^d$ and constants $\mathbf{a}_i, \mathbf{b}_j \in \mathbb{R}^d$ and $\lambda \in \mathbb{R}$, define the scalar-valued function

$$f(\mathbf{w}) = \sum_{i=1}^{m} \sum_{j=1}^{n} (\mathbf{a}_i^\top \mathbf{w} - \mathbf{b}_j^\top \mathbf{w})^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2,$$

where $\mathbf{w}$, $\mathbf{a}_i$ and $\mathbf{b}_j$ are column vectors (e.g. $\mathbf{w} = (w_1, ..., w_d)^\top$) and $\|\mathbf{w}\|_2 = \sqrt{\sum_{j=1}^{d} w_j^2}$ is known as the L_2 norm. Compute the gradient $\nabla_{\mathbf{w}} f(\mathbf{w})$.

Recall: the gradient is a d-dimensional vector of the partial derivatives with respect to each w_i:

$$\nabla_{\mathbf{w}} f(\mathbf{w}) = \left(\frac{\partial f(\mathbf{w})}{\partial w_1}, \ldots, \frac{\partial f(\mathbf{w})}{\partial w_d} \right)^\top.$$

If you're not comfortable with vector calculus, first warm up by working out this problem using scalars in place of vectors and derivatives in place of gradients. Not everything for scalars goes through for vectors, but the two should at least be consistent with each other (when $d = 1$). Do not write out summation over dimensions, because that gets tedious.

Problem 5 - Programming [24 points]

If you're new to Python, the following articles provide pointers to various tutorials and examples for the language:

- Python for Programmers:
  `https://wiki.python.org/moin/BeginnersGuide/Programmers`
- Example programs of increasing complexity:
  `https://wiki.python.org/moin/SimplePrograms`

Problem 5a [4 points]

Implement *find_alphabetically_first_word* in `submission.py`.

Problem 5b [3 points]

Implement *euclidean_distance* in `submission.py`.

Problem 5c [4 points]

Implement *sparse_vector_dot_product* in `submission.py`.

Problem 5d [4 points]

Implement *increment_sparse_vector* in `submission.py`.

Problem 5e [3 points]

Implement *find_nonsingleton_words* in `submission.py`.

Problem 5f [6 points]

Implement *mutate_sentences* in `submission.py`.