VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING


## Natural Language Processing - Exercise (CO3086)

---

## Lab 4

# Math Exercises

**Semester 2, Academic Year 2023 - 2024**

---

Teacher: Bui Khanh Vinh
Students: Nguyen Quang Phu - 2252621

HO CHI MINH CITY, FEBRUARY 2025

# Contents

**1 Problem Description and Solution** **2**

1.1 Problem 1 . . . . . 2

1.1.1 Description . . . . . 2

1.1.2 Solution . . . . . 2

1.2 Problem 2 . . . . . 4

1.2.1 Description . . . . . 4

1.2.2 Solution . . . . . 4

1.3 Problem 3 . . . . . 6

1.3.1 Description . . . . . 6

1.3.2 Solution . . . . . 6

1.4 Problem 4 . . . . . 9

1.4.1 Description . . . . . 9

1.4.2 Solution . . . . . 10

# Chapter 1

# Problem Description and Solution

## 1.1 Problem 1

### 1.1.1 Description

Assume the **following likelihoods** for **each word being part of a positive or negative movie review**, and equal prior probabilities for each class.

| **Word** | **pos** | **neg** |
|---|---|---|
| I | 0.09 | 0.16 |
| always | 0.07 | 0.06 |
| like | 0.29 | 0.06 |
| foreign | 0.04 | 0.15 |
| films | 0.08 | 0.11 |

Table 1.1: Word Probabilities in each Class

What class?

### 1.1.2 Solution

The final equation for the class chosen by a naive Bayes classifier is thus:

$$c_{NB} = \arg\max_{c \in C} P(c) \prod_{f \in F} P(f \mid c)$$

where:

- $C$ is the set of all possible classes.

- $F$ is the set of features in the input.
- $P(c)$ is the prior probability of class $c$.
- $P(f \mid c)$ is the likelihood of feature $f$ given class $c$.

In the Naive Bayes approach, the chosen class $c_{NB}$ maximizes the posterior probability of the class given the features, under the assumption that the features are conditionally independent given the class.

Assume the likelihoods for each word being part of a positive or negative movie review, and equal prior probabilities for each class:

$$P(\text{pos}) = P(\text{neg}) = 0.5$$

For the given review, the posterior probabilities are computed as follows:

$$P(\text{pos} \mid \text{review}) \propto 0.5 \cdot 0.09 \cdot 0.07 \cdot 0.29 \cdot 0.04 \cdot 0.08 = 0.0000029232$$

Similarly, for the negative class:

$$P(\text{neg} \mid \text{review}) \propto 0.5 \cdot P(I \mid \text{neg}) \cdot P(\text{always} \mid \text{neg}) \cdot P(\text{like} \mid \text{neg}) \cdot P(\text{foreign} \mid \text{neg}) \cdot P(\text{films} \mid \text{neg})$$

$$P(\text{neg} \mid \text{review}) \propto 0.5 \cdot 0.16 \cdot 0.06 \cdot 0.06 \cdot 0.15 \cdot 0.11 = 0.000004752$$

Since:

$$P(\text{neg} \mid \text{review}) > P(\text{pos} \mid \text{review})$$

the review is classified as **negative**.

## 1.2 Problem 2

### 1.2.1 Description

Given the following short movie reviews, each labeled with a **genre, either comedy or action**:

1. fun, couple, love, love **comedy**
2. fast, furious, shoot **action**
3. couple, fly, fast, fun **comedy**
4. furious, shoot, shoot, fun **action**
5. fly, fast, shoot, love **action**

and a new document $D$: **fast, couple, shoot, fly**.

Compute the **most likely class for** $D$. Assume a **naive Bayes classifier** and use **add-1 smoothing** for the likelihoods.

### 1.2.2 Solution

To solve the problem, we will use the Naive Bayes classifier with Laplace (add-one) smoothing.

**Naive Bayes Classifier Formula:**

The Naive Bayes classifier chooses the class $c_{NB}$ that maximizes the posterior probability:

$$c_{NB} = \arg\max_{c \in C} P(c) \prod_{f \in F} P(f \mid c)$$

**Likelihood with Laplace Smoothing:**

The likelihood $P(f \mid c)$ is calculated using Laplace smoothing:

$$P(f \mid c) = \frac{\text{count}(f, c) + 1}{\sum_{w \in V} \text{count}(w, c) + |V|}$$

where:

- $\text{count}(f, c)$ is the number of times word $f$ appears in class $c$,
- $|V|$ is the size of the vocabulary.

**Word counts, Laplace-smoothed counts, and probabilities for Comedy and Action classes.**

| **Word** | **Word Count** | | **Count + 1 (Laplace)** | | **Probabilities** | |
|---|---|---|---|---|---|---|
| | **Comedy** | **Action** | **Comedy** | **Action** | **P**(word \| Comedy) | **P**(word \| Action) |
| fun | 3 | 1 | 4 | 2 | $\frac{4}{9+7} = 0.25$ | $\frac{2}{11+7} \approx 0.111$ |
| couple | 2 | 0 | 3 | 1 | $\frac{3}{9+7} = 0.1875$ | $\frac{1}{11+7} \approx 0.056$ |
| love | 2 | 1 | 3 | 2 | $\frac{3}{9+7} = 0.1875$ | $\frac{2}{11+7} \approx 0.111$ |
| fast | 1 | 2 | 2 | 3 | $\frac{2}{9+7} = 0.125$ | $\frac{3}{11+7} \approx 0.167$ |
| furious | 0 | 2 | 1 | 3 | $\frac{1}{9+7} = 0.0625$ | $\frac{3}{11+7} \approx 0.167$ |
| shoot | 0 | 4 | 1 | 5 | $\frac{1}{9+7} = 0.0625$ | $\frac{5}{11+7} \approx 0.278$ |
| fly | 1 | 1 | 2 | 2 | $\frac{2}{9+7} = 0.125$ | $\frac{2}{11+7} \approx 0.111$ |

Table 1.2: Word counts, Laplace-smoothed counts, and probabilities for Comedy and Action classes.

**Prior Probabilities:**

$$P(\text{comedy}) = \frac{2}{5} = 0.4, \quad P(\text{action}) = \frac{3}{5} = 0.6$$

**Compute Posterior Probabilities:**

For the document $D = \{\text{fast, couple, shoot, fly}\}$:

$$P(\text{comedy} \mid D) \propto P(\text{comedy}) \cdot P(\text{fast} \mid \text{comedy}) \cdot P(\text{couple} \mid \text{comedy}) \cdot P(\text{shoot} \mid \text{comedy}) \cdot P(\text{fly} \mid \text{comedy})$$

$$P(\text{comedy} \mid D) \propto 0.4 \cdot 0.125 \cdot 0.1875 \cdot 0.0625 \cdot 0.125 \approx 0.000586$$

$$P(\text{action} \mid D) \propto P(\text{action}) \cdot P(\text{fast} \mid \text{action}) \cdot P(\text{couple} \mid \text{action}) \cdot P(\text{shoot} \mid \text{action}) \cdot P(\text{fly} \mid \text{action})$$

$$P(\text{action} \mid D) \propto 0.6 \cdot 0.167 \cdot 0.056 \cdot 0.278 \cdot 0.111 \approx 0.000173$$

**Conclusion:**

Since $P(\text{comedy} \mid D) > P(\text{action} \mid D)$, the document $D$ is classified as **comedy**.

## 1.3 Problem 3

### 1.3.1 Description

Train two models, **multinomial naive Bayes** and **binarized naive Bayes**, both with add-1 smoothing, on the following document counts for key sentiment words, with positive or negative class assigned as noted.

| doc | good | poor | great | class |
|---|---|---|---|---|
| $d_1$ | 3 | 0 | 3 | pos |
| $d_2$ | 0 | 1 | 2 | pos |
| $d_3$ | 1 | 3 | 0 | neg |
| $d_4$ | 1 | 5 | 2 | neg |
| $d_5$ | 0 | 2 | 0 | neg |

Table 1.3: Document counts for key sentiment words.

Use both naive Bayes models to assign a class (pos or neg) to this sentence:

*A good, good plot and great characters, but poor acting.*

With naive Bayes text classification, we simply ignore (throw out) any word that never occurred in the training document. (We don't throw out words that appear in some classes but not others; that's what add-one smoothing is for.) **Do the two models agree or disagree?**

### 1.3.2 Solution

For the sentence *A good, good plot and great characters, but poor acting*, the vocabulary from that sentence is: *A, good, plot, and, great, characters, but, poor, acting.*

But from the problem description, we only consider those words that appear in the 5 documents. So the remaining vocabulary includes: *good, poor, great.*

Train two models, **multinomial naive Bayes** and **binarized naive Bayes**, both with add-1 smoothing, on the following document counts for key sentiment words.

Here is the document count for multinomial Naive Bayes use:

| doc | good | poor | great | class |
|---|---|---|---|---|
| $d_1$ | 3 | 0 | 3 | pos |
| $d_2$ | 0 | 1 | 2 | pos |
| $d_3$ | 1 | 3 | 0 | neg |
| $d_4$ | 1 | 5 | 2 | neg |
| $d_5$ | 0 | 2 | 0 | neg |

Table 1.4: Document counts for key sentiment words (multinomial Naive Bayes)

Here is the table that shows the count of those words with add-one smoothing already and their corresponding probabilities:

| Word | Count (+) | Count (-) | $\mathbf{P}(\text{word} \mid +)$ | $\mathbf{P}(\text{word} \mid -)$ |
|---|---|---|---|---|
| good | 4 | 3 | 0.333 | 0.176 |
| poor | 2 | 11 | 0.167 | 0.647 |
| great | 6 | 3 | 0.500 | 0.176 |

Table 1.5: Word counts with add-one smoothing and corresponding probabilities (Multinomial Naive Bayes)

Here is the document count for binary Naive Bayes use:

| doc | good | poor | great | class |
|---|---|---|---|---|
| $d_1$ | 1 | 0 | 1 | pos |
| $d_2$ | 0 | 1 | 1 | pos |
| $d_3$ | 1 | 1 | 0 | neg |
| $d_4$ | 1 | 1 | 1 | neg |
| $d_5$ | 0 | 1 | 0 | neg |

Table 1.6: Document counts for key sentiment words (binary Naive Bayes)

For **binary Naive Bayes**, after per-document binarization and add-one smoothing, the binary counts and their corresponding probabilities are:

| Word | Count (+) | Count (-) | $\mathbf{P}(\text{word} \mid +)$ | $\mathbf{P}(\text{word} \mid -)$ |
|---|---|---|---|---|
| good | 2 | 2 | 0.286 | 0.250 |
| poor | 2 | 4 | 0.286 | 0.500 |
| great | 3 | 2 | 0.429 | 0.250 |

Table 1.7: Binary counts with add-one smoothing and corresponding probabilities (Binary Naive Bayes)

The Naive Bayes classifier chooses the class $c_{NB}$ that maximizes the posterior probability:

$$c_{NB} = \arg\max_{c \in C} P(c) \prod_{f \in F} P(f \mid c)$$

**Prior Probabilities:**

$$P(\text{pos}) = \frac{2}{5} = 0.4, \quad P(\text{neg}) = \frac{3}{5} = 0.6$$

**Compute Posterior Probabilities:**

For the document $D =$ "A good, good plot and great characters, but poor acting" with class C, we have the formula:

$$P(\text{c} \mid D) \propto P(\text{c}) \cdot P(\text{good} \mid \text{c}) \cdot P(\text{good} \mid \text{c}) \cdot P(\text{great} \mid \text{c}) \cdot P(\text{poor} \mid \text{c})$$

**For Multinomial Naive Bayes:**

$$P(\text{pos} \mid D) \propto 0.4 \cdot 0.333 \cdot 0.333 \cdot 0.5 \cdot 0.167 \approx 0.0037$$

$$P(\text{neg} \mid D) \propto 0.6 \cdot 0.176 \cdot 0.176 \cdot 0.176 \cdot 0.647 \approx 0.0021$$

So for **Multinomial Naive Bayes**, we conclude that the sentence is in class **pos**

**For Binary Naive Bayes:**

$$P(\text{pos} \mid D) \propto 0.4 \cdot 0.286 \cdot 0.286 \cdot 0.429 \cdot 0.286 \approx 0.0040$$

$$P(\text{neg} \mid D) \propto 0.6 \cdot 0.25 \cdot 0.25 \cdot 0.25 \cdot 0.5 \approx 0.0047$$

So for **Binary Naive Bayes**, we conclude that the sentence is in class **neg**

**Compare the results for the two models, they disagree.**

## 1.4 Problem 4

### 1.4.1 Description

Consider that our document collection $S$ has the following documents: $D_1, \ldots, D_5$:

| **Document** | **Words** |
|---|---|
| $D_1$ | Data Base System Concepts |
| $D_2$ | Introduction to Algorithms |
| $D_3$ | Computational Geometry: Algorithms and Applications |
| $D_4$ | Data Structures and Algorithm Analysis on Massive Data Sets |
| $D_5$ | Computer Organization |

Table 1.8: Document Collection $S$.

Our dictionary **DICT** consists of 8 words: $\{w_1 = \text{data}, w_2 = \text{system}, w_3 = \text{algorithm}, w_4 = \text{computer}, w_5 = \text{geometry}, w_6 = \text{structure}, w_7 = \text{analysis}, w_8 = \text{organization}\}$. We consider that, by stemming, "computer" and "computational" are regarded as the same word, and so are "algorithms" and "algorithm."

1. Let $tf(w, D)$ denote the term frequency of term $w$ in a document $D$. Give the value of $tf(w_i, D_j)$ for all $1 \leq i \leq 8$ and $1 \leq j \leq 5$.

2. Let $idf(w)$ denote the inverse document frequency of term $w$ as defined in our lecture notes. Give the value of $idf(w_i)$ for all $1 \leq i \leq 8$.

3. Convert each document in $S$ into an 8-dimensional point according to the tf-idf model as defined in our lecture notes.

4. Assume that we have received a query with terms "Geometry Algorithm Concepts." Convert the query to an 8-dimensional point.

5. Rank the documents in descending order of their relevance to the query in Problem 4 according to the cosine metric.

### 1.4.2 Solution

**1. Term Frequency ($tf$)**

$$\mathrm{tf}_{t,d} = \begin{cases} 1 + \log_{10}(\mathrm{count}(t,d)) & \text{if } \mathrm{count}(t,d) > 0 \\ 0 & \text{otherwise} \end{cases}$$

The term frequency $tf(w, D)$ is the number of times the term $w$ appears in document $D$.

The word-count table is:

| **Word** | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ |
|---|---|---|---|---|---|
| $w_1$(data) | 1 | 0 | 0 | 2 | 0 |
| $w_2$(system) | 1 | 0 | 0 | 0 | 0 |
| $w_3$(algorithm) | 0 | 1 | 1 | 1 | 0 |
| $w_4$(computer) | 0 | 0 | 1 | 0 | 1 |
| $w_5$(geometry) | 0 | 0 | 1 | 0 | 0 |
| $w_6$(structure) | 0 | 0 | 0 | 1 | 0 |
| $w_7$(analysis) | 0 | 0 | 0 | 1 | 0 |
| $w_8$(organization) | 0 | 0 | 0 | 0 | 1 |

The term frequency table is:

| **Word** | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ |
|---|---|---|---|---|---|
| $w_1$(data) | 1 | 0 | 0 | 1.3 | 0 |
| $w_2$(system) | 1 | 0 | 0 | 0 | 0 |
| $w_3$(algorithm) | 0 | 1 | 1 | 1 | 0 |
| $w_4$(computer) | 0 | 0 | 1 | 0 | 1 |
| $w_5$(geometry) | 0 | 0 | 1 | 0 | 0 |
| $w_6$(structure) | 0 | 0 | 0 | 1 | 0 |
| $w_7$(analysis) | 0 | 0 | 0 | 1 | 0 |
| $w_8$(organization) | 0 | 0 | 0 | 0 | 1 |

**2. Inverse Document Frequency ($idf$)**

The inverse document frequency $idf(w)$ is given by (the formula is written based on the book Speech and Language Processing):

$$idf(w) = \log_{10}\left(\frac{|S|}{df(w)}\right),$$

where $df(w)$ is the number of documents containing $w$.

| **Word** | $idf(w)$ |
|---|---|
| $w_1$(data) | 0.40 |
| $w_2$(system) | 0.70 |
| $w_3$(algorithm) | 0.22 |
| $w_4$(computer) | 0.40 |
| $w_5$(geometry) | 0.70 |
| $w_6$(structure) | 0.70 |
| $w_7$(analysis) | 0.70 |
| $w_8$(organization) | 0.70 |

### 3. TF-IDF Vector Representation

Each document is converted into an 8-dimensional vector using $tf$ and $idf$:

$$p_i[j] = tf(w_j, D_i)) \cdot idf(w_j)$$

The TF-IDF table is:

| | $w_1$ | $w_2$ | $w_3$ | $w_4$ | $w_5$ | $w_6$ | $w_7$ | $w_8$ |
|---|---|---|---|---|---|---|---|---|
| $p_1$ | 0.40 | 0.70 | 0 | 0 | 0 | 0 | 0 | 0 |
| $p_2$ | 0 | 0 | 0.22 | 0 | 0 | 0 | 0 | 0 |
| $p_3$ | 0 | 0 | 0.22 | 0.40 | 0.70 | 0 | 0 | 0 |
| $p_4$ | 0.52 | 0 | 0.22 | 0 | 0 | 0.70 | 0.70 | 0 |
| $p_5$ | 0 | 0 | 0 | 0.40 | 0 | 0 | 0 | 0.70 |

### 4. Query Vector

Given a query $Q =$ "Geometry Algorithm Concepts"

Based on the query, the word Algorithm and Geometry is the ones that appear in the vocabulary of the TF-IDF table, so the query vector is in the form of

$$q = (0, 0, x_3, 0, x_5, 0, 0, 0)$$

Then query vector is (becaue those word frequency in the query is 1, and there tf value is $1 + \log_{10}(\text{count}(t, d)) = 1 + 0 = 1$):

$$q = (0, 0, 0.22, 0, 0.7, 0, 0, 0).$$

### 5. Cosine Similarity

The cosine similarity between $p_i$ and $q$ is:

$$\text{score}(p_i, q) = \frac{p_i \cdot q}{|p_i| \cdot |q|}$$

The scores for all documents are:

| Document | Score |
|---|---|
| $D_1$ | 0 |
| $D_2$ | 0.30 |
| $D_3$ | 0.88 |
| $D_4$ | 0.06 |
| $D_5$ | 0 |

**6. Final Ranking**

The documents are ranked in descending order of their scores:

$$D_3, D_2, D_4, D_1, D_5$$