Name, Vorname: Wang, Jiahui	**Fach:** B.Sc.Mathematik	**Martriknr.**2992080
Name, Vorname: Zhao, Jiaqi	**Fach:** M.Sc.Autonome System	**Martriknr.** 3470190
Name, Vorname: Wang, Ziyin	**Fach:** M.Sc.Informatik	**Martriknr.** 3435397

Assignment 2

Task 1: Simple Bayes

1. Calculate the probability:

$$\begin{aligned} P(yellow) &= P(yellow \mid 1994) \cdot P(1994) + P(yellow \mid 1996) \cdot P(1996) \\ &= \frac{30}{100} \cdot \frac{1}{2} + \frac{16}{100} \cdot \frac{1}{2} \\ &= \frac{23}{100} \\ P(1994 \mid yellow) &= \frac{P(yellow \mid 1994) \cdot P(1994)}{P(yellow)} \\ &= \frac{\frac{30}{100} \cdot \frac{1}{2}}{\frac{23}{100}} \\ &= \frac{15}{23} \end{aligned}$$

2. Calculate the probability:

$$\begin{aligned} P(apple) &= P(apple \mid box = 1) \cdot P(box = 1) + P(apple \mid box = 2) \cdot P(box = 2) \\ &= \frac{4}{14} \cdot \frac{1}{2} + \frac{6}{14} \cdot \frac{1}{2} \\ &= \frac{5}{14} \end{aligned}$$

Task 2: Fake News Classification with Naive Bayes

Please see the assignment2_Jiahui_Jiaqi_Ziyin.ipynb file.

Task 3. k-NN for Text Classification

Since the meaning of a piece of text is determined by its words, it is naturally to represent the text by its words. However, the set of all often-used words are on one hand too complicated to analyze and on the other hand not necessarily helpful for solving the certain problem. The approach introduced here is to collect the problem related set of words from training dataset, and use this set of words to measure the distance of different pieces of text.

PROBLEM SETTING:

- Given pieces of text $\mathbf{T_1, T_2, \ldots, T_n}$, with predefined labels $\mathbf{y_1, y_2, \ldots, y_n}$ (Training dataset)
- Making Prediction of label on a new piece of text **T** (Testing Data)

APPROACH:

I. Define the problem related set of words **W** as the union of all words which occur in testing data $\mathbf{T_1, T_2, \ldots, T_n}$.
II. Enumerate $\mathbf{W = \{w_1, w_2, \ldots, w_m\}}$ and calculate the frequency of each word $\mathbf{f_1, f_2, \ldots, f_m}$.
III. Represent each training text $\mathbf{T_i}$ by a vector $\mathbf{t_i}$ with length **m**, whose **j-th** entry is the frequency of $\mathbf{w_j}$ appears in $\mathbf{T_i}$ divided by the reference frequency $\mathbf{f_j}$.
IV. Represent the testing text **T** in the same way in **III**.
V. Define the distance of two text $\mathbf{T_i}$ and **T** as the Euclidean distance of the representing vectors $\mathbf{t_i}$ and **t**.
VI. Conduct k-NN algorithm to predict the label of **T**.

EXAMPLE:

The comments of a pet website are to be classified into two categories: "cat related" and "dog related".

The given pieces of training text are:

- $\mathbf{T_1}$ ="I love cats, they are so cute.", $\mathbf{y_1}$ = "cat related";
- $\mathbf{T_2}$="OMG, my dog shit in the room again!", $\mathbf{y_2}$ = "dog related".

The testing text is:

- **T** ="I really want to adopt a long-haired dog, but my wouldn't allow me :(" .

Set of words is:

- **W** = {"I", "love", "cats", "they", "are", "so", "cute","OMG", "my", "dog", "shit", "in", "the", "room", "again"}, totally 15 words, with the same frequency 1.

The representing vectors are:

- $\mathbf{t_1}$ = [1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0];
- $\mathbf{t_2}$ = [0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1];
- **t** = [1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0].

The distances of **t** with $\mathbf{t_1}$ and $\mathbf{t_2}$ are:

- d(**t**, $\mathbf{t_1}$) = $2\sqrt{2}$.
- d(**t**, $\mathbf{t_2}$) = $\sqrt{7}$

According to the 1-NN algorithm, the test data is to be labeled as "dog related", which is correct when we check it manually.

ADVANTAGES AND DISADVANTAGES:

Pros:

- The construction is easy to understand and the algorithm is easy to implement.
- It dose not rely on prior probabilities and dosen't need any additional linguistic data base.

Cons:

- Many important information of text is ignored, e.g. the order of words.
- Many redundant words like "the", "a", "of", etc, are taken into consideration.
- It could be the case, that some relevant words used in the testing text dose not appear in **W**, which has no impact to the result.
- Since all words appearing in the training text is collected in the set W, the dimension of input data grows rapidly with the number and the length of training text. However, k-NN algorithm is very slow in high-dimension case.

Task 4: kNN in High-Dimensional Feature Spaces

4.1 In high-dimensional feature spaces, KNN may suffer from the curse of dimensionality, which means that Euclidean distance is unhelpful in high dimensions, because all vectors are almost equidistant to the search query vector. For example, imagine multiple points lying more or less on a circle with the query point at the center; the distance from the query to all data points in the search space is almost the same.

4.2 In order to solve or circumvent this problem we can reduce the dimensionality of our feature space with different methods of feature transform and feature selection techniques, such as sequential floating forward selection (SFFS), linear discriminant analysis (LDA), principle component analysis PCA.

Method and steps:

• Principle Component Analysis(PCA)

1) Given samples S:

$$S = x_n \in \mathbb{R}^n, 1 \leq n \leq N$$

2) Parameter: reduced number of features $\overline{d} < d$

3) Initial calculation:

•compute sample correlation matrix

$$R = \frac{1}{N} \sum_{n=1}^{N} x_n x_n^\top$$

•compute eigenvalue decomposition

$$R = VDV^\top$$

•choose first $\overline{d}$ eigenvectors:

$$W = \begin{bmatrix} v_1 & \cdots & v_{\overline{d}} \end{bmatrix}$$

4) Compression feature reduction

$$\overline{x}_n = W^\top x_n, 1 \leq n \leq N$$

5) Reconstruction

$$\hat{x}_n = W\overline{x}_n$$

6) Optimization

$$\arg\min_W J(W) = \sum_{n=1}^{N} \left\| x_n - WW^\top x_n \right\|^2$$

s.t. $W^\top \cdot W = I$.