Data Science of Text Generation

Università della Svizzera italiana

1. Taking your Chance

1.1. Probability Rules

PROBABILITY (of an event): **ratio of number of cases favorable to number of all cases possible**, when nothing leads us to expect that any one of these cases should occur more than any other, which renders them, for us, equally possible.
Key concepts: *SAMPLE SPACE*, *EVENT*, *COMBINING EVENTS*, *INDEPENDENCE*

1.2. Conditional Probabilities

CONDITIONAL PROBABILITY (of an event A given an event B):

$$\mathrm{P}(\mathrm{A}|\mathrm{B}) = \frac{\mathrm{P}(\mathrm{A} \cap \mathrm{B})}{\mathrm{P}(\mathrm{B})}$$

Key concepts: *DEPENDENCE*, *CONDITIONAL PROBABILITY*

1.3. Bayes' Theorem

BAYES' THEOREM:

$$\mathrm{P}(\mathrm{A}|\mathrm{B}) = \frac{\mathrm{P}(\mathrm{B}|\mathrm{A}) \cdot \mathrm{P}(\mathrm{A})}{\sum_{i=1}^{n} P(B|A_i) \cdot \mathrm{P}(A_i)}$$

Key concepts: *TOTAL PROBABILITY RULE*, *SENTIMENT ANALYSIS*, *BAYES' THEOREM*

2. Are you able to Recognize your Sentiment from your Sentence?

2.1. Birthday Paradox

TASKS:

- **TASK 1&2**: Create and apply the function `generate_birthdays(n)` to one of the following values of n:

$$[5, 10, 15, 25, 30, 35, 40, 45, 50, 75, 100]$$

- **TASK 3** (*Optional*) Apply the function `generate_birthdays(n)` to all the sizes above.
- **TASK 4&5**: Create and apply the function `has_duplicate(birthdays)` for the chosen group size.
- **TASK 6**: Conduct a simulation study for the chosen group size.
- **TASK 7** (*Optional*) Conduct a **simulation study** for all the sizes above.
- **TASK 8**: Answer the questions!

Key concepts: *FOR LOOP*, *CONDITIONALS*, *SIMULATION STUDY*

2.2. Conditional Probabilities & Videogames

*In the digital realm of Cybervale, Sarah, a brave hacker, ventures into the treacherous Darknet. This network is fraught with dangers, from encrypted traps to lurking digital predators. As she journeys through, Sarah discovers that **40% of the pathways lead to regions infested with malicious code**, posing imminent threats (there, the probability of **encountering a malware is 90%**). In a secure region **only 5% of encountered objects are malware**. Additionally, **within secure sectors**, there's a **25% chance that Sarah will find a powerful antivirus program**, offering her crucial defense. Unfortunately, in **infected areas**, the **probability** of finding such a program **drops to a mere 15%**. **Given the type of region** (secure or infected), finding the antivirus and encountering a malware can be considered independent events.*

TASKS:

- **TASK 9**: Translate the videogame in probability terms
- **TASK 10**: Answer the questions!
- **TASK 11&12**: Design your videogame by writing and applying the function `sarahs_journey()`;

Key concepts: *EXPLAIN REAL PROBLEMS VIA PROBABILITIES*

2.3. Sentiment Analysis

SENTIMENT ANALYSIS: aimed at computing **probability of positive (or negative) sentiment given a sentence of n words**.
TASKS:

- **TASK 13-16**: Load datasets using `load_dataset(file_path)`; merge difference sources; divide positive & negative;
- **TASK 17&18**: Build your sentiment dictionary (probability of the word in a positive/negative context);
- **TASK 19&20**: Create and apply `calculate_sentiment_probability` to compute the sentiment given a sentence;
- **TASK 21**: Have fun with the last function: Are you able to Recognize your Sentiment from your Sentences?

Key concepts: *SENTIMENT RANDOM VARIABLE, WORD RANDOM VARIABLE, SENTIMENT DICTIONARY*

3. Markov Chains

TASK: Make a haiku by each choosing one word after hearing what previous person has chosen.

3.1. Markov Chain: Definition

MARKOV CHAIN: stochastic process that only looks at last state to decide where to go next.

$$\mathrm{P}(X_{n+1} = x_n | X_0 = x_0, X_1 = x_1, \ldots, X_n = x_n) = \mathrm{P}(X_{n+1} = x_n | X_n = x_n)$$

Key concepts: *STATE, MARKOV CHAIN, MARKOV PROPERTY, CHESS AS MARKOV CHAIN, RANDOM WALK*

3.2. Transition Probabilities

TRANSITION MATRIX: matrix $\mathrm{P} = [p_{ij}]$ such that:

$$p_{ij} = (X_{n+1} = j | X_n = \mathrm{i})$$

The rows of a transition matrix adds up to 1.

Key concepts: *TRANSITION PROBABILITI*ES, *STATE SPACE*, *TIME STRUCTURE*

3.3. Limiting Probabilities

CHAPMAN-KOLMOGOROV THEOREM:

$$P_{ij}{}^{(n)} = (P^n)_{ij}$$

Key concepts: *INFECTION DISEASES MODEL, FINANCIAL MARKET MODEL, LEHRER-WAGNER MODEL, LIMITING DISTRIBUTION, RECURRENCE, FOREST FIRE SIMULATION*

3.4. Text Generation using Markov Chains

$$\mathrm{p_{ij}} = \frac{\#\text{ instances of word i followed by word j}}{\#\text{ instances of word i}}$$

Key concepts: *MODEL TRAINING, TEXT GENERATION, ADVANCED TEXT GENERATION*

4. Are you Able to Write like... Oscar Wilde?

4.1. Who do you want to Resemble as a Writing Style?

TASKS:

- **TASK 1&2**: Create and apply the function `read_text_file(filename)` to load the Oscar Wilde (OW)'s book and *Green Eggs and Ham* by Dr. Seuss.
- **TASK 3**: Read and apply the function `preproc(canon, letter, delimit, remov)` to above mentioned texts.

Key concepts: *LOADING DATA, PRE-PROCESSING STEP*

4.2. Time for Markov Chains

- **TASK 4&5**: Create and apply the function `transition(canon)` to the text *Green Eggs and Ham*. Use the function `transition_row_for_word(word, transition_matrix, words)` to answer the questions.
- **TASK 6**: Read the code to exploit networks to visualize the transition probabilities.
- **TASK 7**: Answer the questions!

Key concepts: *TRANSTION MATRIX IN PRACTICE, NETWORK VISUALIZATION OF TRANSITION PROBABILITIES, IRREDUCIBLE MARKOV CHAINS*

4.3. Let us work... Step by Step!

Our final goal (if we opt for a depth equal to 2):

$$P(W_3 = w | W_1 = w_1, W_2 = w_2)$$

By means of a function named `writer`.

- **TASK 8&9**: Create and apply the function `initialize_novel(canon, depth, seed, n_tot)` to the OW text with depth equal to 3, without or with a seed provided.
- **TASK 10&11**: Create and apply the function `get_candidates(canon, novel, temp_depth, n_tot)` to the output obtained before.
- **TASK 12&13**: Create and apply the function `choose_next_word(canon, select)` to the output obtained before.
- **TASK 14&15**: Create and apply the function `generate_sentence(canon, novel, depth, n_tot)` to the output obtained before.
- **TASK 16**: Read the function `format_sentences(novel, delimit)` to deal with punctuation.
- **TASK 17**: Put the pieces together by writing the function function `writer(canon)`.

Key concepts: *FUNCTION COMPOSITION*

4.4. Space for Imagination!

Use the function `writer(canon)` to write like... Oscar Wilde!