SNLP 2016
Exercise 3

Submission date: 22.05.2016, 23:59

Entropy

1) (7 points) For this exercise, you should select three texts, from a source such as NLTK or Project Gutenberg. Choose two in the same language (for example, English), and one in a different language (for example, German). We will consider the token-level distribution, using the maximum likelihood estimation with Lidstone smoothing*.

- (1 point) First, pre-process (lowercase, remove puncutation) and tokenize the text. You may use any functions from NLTK for this task.
- (3 points) Implement a function[1] that calcuates the entropy of a given text, using the formula from the slides. Run it on the three texts that you have selected, and report the results. Comment briefly on how to interpret the numbers from the results.
- (3 points) Implement a function that takes two texts as arguments, and calculates the KL divergence $D(p||q)$ between them. For this task, you will need to use the probabilities after applying Lidstone smoothing. Calculate and report the KL divergence between the two texts of the same language, and between two of the texts in different languages. Comment on any difference in the results.

*Smoothing is a technique used in language modeling to account for words that are missing from the training sample, so that they don't have a probability of 0. It works by taking a small amount of probability mass from other words, and giving them to new, unseen words. The formula for one method, Lidstone smoothing, is given below:

$$P_{lidstone}(w) = \frac{count(w) + \alpha}{N + \alpha V} \tag{1}$$

where N is the total number of tokens, and V the size of the vocabulary. We'll use $\alpha = 0.1$ for this example.

Text Compression

2) (3 points) In this task, we will determine an encoding for the following string of characters:

aaabacddbabacdd

- (2 points) Use the Huffman coding algorithm (explained here: `https://en.wikipedia.org/wiki/Huffman_coding#Informal_description`) to calculate, by hand, the encoding for the string above. Report the code obtained for each character, and use it to encode the string above.
- (1 point) Using the formula in the slides, calculate the optimal length of the code. Comment on how this relates to the length of the code you found in the previous question.

[1] Implementation hint: You may want to begin both of these tasks by writing a function that takes a single text as an arugment, and returns a dictionary with words as keys, and their probabilities as values.

Bonus

3) (2 points) The joint entropy $H(X, Y)$ measures the entropy of a joint probability distribution, and is given by the following formula:

$$H(X, Y) = -\sum_{x}\sum_{y} P(x, y) log_2[P(x, y)] \tag{2}$$

Prove the following identity between the joint and conditional entropy: $H(X, Y) = H(X) + H(Y|X)$.

1 Submission Instructions: Read carefully

- You can form groups of maximum 3 people.
- Submit only 1 archive file in the ZIP format with name containing the MN of all the team members, e.g.:

Exercise_01_MatriculationNumber1_MatriculationNumber2_MatriculationNumber3.zip

- Provide in the archive:
  - your code, accompanied with sufficient comments,
  - a PDF report with answers, solutions, plots and brief instructions on executing your code,
  - a README file with the group member names, matriculation numbers and emails,
  - Data necessary to reproduce your results [2]
- The subject of your submission mail must contain the string [SNLP] (including the braces) and explicitly denoting that it is an exercise submission, e.g:

[SNLP] Exercise Submission 01

- Depending on your tutorial group, send your assignment to the corresponding tutor:
  - Sedigheh Eslami: *eslami@mpi-inf.mpg.de*
  - Naszdi Kata: *b.naszadi@gmail.com*
  - Stephanie Lund: *stflund@gmail.com*

2 General Information

- In your mails to us regarding the tutorial please add the tag [SNLP] in the subject accompanied by an appropriate subject briefly describing the contents.
- Feel free to use any programming language of your liking. However we strongly advise in favor of Python, due to the abundance of available tools (also note that Python3 comes with an excellent native support of UTF8 strings).
- Avoid using libraries that solve what we ask you to do (unless otherwise noted).
- Avoid building complex systems. The exercises are simple enough.
- Do not include any executable files in your submission, as this may cause the e-mail server to reject it.

[2] If you feel that these files are beyond reasonable size for an email submission and also reasonably convenient, please provide a means for us to access them online

- **In case of copying, all the participants (including the original solution) will get 0 points for the whole assignment. Note: it is rather easy to identify a copied solution. Plagiarism is also not tolerated.**
- **Missing the deadline even for a few minutes, will result in 50% point reduction. Submission past the next tutorial, is not corrected, as the solutions will already be discussed.**
- **Please submit in your solutions necessary to support your claims. Failure to do so, might results in reduction of points in the relevant questions.**
- Each assignment has 10 points and perhaps some bonus points (usually 2 or 3). In order to qualify for the exams, you need to have 2/3 of the total points. For example, in case there are 12 assignments, you need to collect at least 80 out of the 120 points to be eligible for the exams. A person that gets 10 plus 2 bonus points in every exercise, needs to deliver only 7 assignments in order to be eligible for the exams, since 7*12=84.
- Attending the tutorial gives 2 points increase for the corresponding assignment.
- Exercise points (including any bonuses) guarantee only the admittance to the exam, however have no further effect on the final exam grade.