[handwritten margin, top right] feature selection / bigram, trigram / ↓ sparse data problem.

Discussion

[handwritten] Sentiment analysis, Author ID, Auto-fact check.

1. What is **text classification**? Give some examples.

   (a) Why is text classification generally a difficult problem? What are some hurdles that need to be overcome?

   [handwritten] BOW — document representations, features

   (b) Consider some (supervised) text classification problem, and discuss whether the following (supervised) machine learning models would be suitable:

   [handwritten left margin] high dimensionality issue.

   [handwritten] NB assumption — conditional independence. Sensitive to large data set. relax

      i. k-Nearest Neighbour using Euclidean distance — *[handwritten]* classify based on length which is really bad.
      ii. k-Nearest Neighbour using Cosine similarity
      iii. Decision Trees using Information Gain — *[handwritten]* choose rare features / mostly useless features.
      iv. Naive Bayes
      v. Logistic Regression — *[handwritten]* Can handle
      vi. Support Vector Machines — *[handwritten]* Multiple classes.

2. For the following "corpus" of two documents:

```
1.  how much wood would a wood chuck chuck if a wood chuck
    would chuck wood
2.  a wood chuck would chuck the wood he could chuck if
    a wood chuck would chuck wood
```

   (a) Which of the following sentences: `a wood could chuck`; `wood would a chuck`; is more probable, accoding to:

      i. An unsmoothed uni-gram language model?
      ii. A uni-gram language model, with Laplacian ("add-one") smoothing?
      iii. An unsmoothed bi-gram language model?
      iv. A bi-gram language model, with Laplacian smoothing?
      v. An unsmoothed tri-gram language model?
      vi. A tri-gram language model, with Laplacian smoothing?

[handwritten equations]

$$P_{uni}(\text{---}) = \frac{4}{34} \times \frac{1}{34} \times \frac{5}{34} \times \frac{8}{34}$$

$$P(B) = \cdots$$

$$\frac{5}{45} \times \frac{10}{65} \times \frac{2}{65} \cdots$$

$$P_{bi}(\cdots) = \frac{1}{2} \times \frac{4}{4} \times \frac{0}{8} \times \frac{1}{1} \times \frac{0}{9}$$

$$P_L = \frac{1 + 9}{|V| + 0}$$

[handwritten top right] $P(\,</s> \mid (\text{chuck}))$

$$P_{Tri} = P(a \mid <s><s>) * P(\text{wood} \mid <s>\,a) * \cdots$$

(b) Based on the "corpus", the vocabulary = {a, chuck, could, he, how, if, much, the, wood, would, </s> }, and the **continuation counts** of the following words are given as follows:

- a = 2
- could = 1
- he = 1
- how = 0
- if = 1
- much = 1
- the = 1
- would = 2
- </s> = 1

[handwritten: $\frac{C(w_i) - D}{\sum C(w_i)}$ — way to distribute D.]

[handwritten: chuck = 4, wood = 4, wood, would, could, chuck.]

i. What is the continuation probability of chuck and wood?

3. What does **back–off** mean, in the context of smoothing a language model? What does **interpolation** refer to?

[handwritten: redistributed based on lower order model (unigram) probability]

Programming

[handwritten: $\chi + \lambda_1 P_1 + \lambda_2 P_2 + \cdots$]

1. In the `03-classification` notebook, observe how different tokenisation regimes alter the text classification performance of the various classifiers on the given Reuters dataset problem.

   (a) Alter the tokenisation strategy so that it incorporates other stages, for example, punctuation, or stemming/lemmatisation.

   (b) Does performance increase or decrease? Are some classifiers affected more than others? Why do you think that is?

[handwritten: linear Combine three order of prob...]

2. Using the iPython notebook `04-ngram`, randomly generate some sentences based on the bi-gram models of the Gutenberg corpus and the Penn Treebank. What do you notice about these sentences? Are there any sentences which might get returned for both corpora? Why?

3. Find a sentence with a higher probability than *revenue increased last quarter.*, according to:

   (a) The Gutenberg corpus, using bi-grams smoothed with Laplacian smoothing

   (b) The Gutenberg corpus, using bi-grams smoothed with Interpolation

   (c) The Penn Treebank corpus, using bi-grams and Laplacian smoothing

   (d) The Penn Treebank corpus, using bi-grams and Interpolation

4. Find the perplexity of the above (smoothed) language models for a number of sentences. Why does Interpolation generally have better perplexity?

Catch-up

- What is a **language model**? What is an n-**gram language model**? Why are language models important?

- What do **uni-gram**, **bi-gram**, **tri-gram**, etc. signify?

- Why is **smoothing** important? *[handwritten: → zero frequency problem]*

- Why do we usually use **log probabilities** when finding the probability of a sentence according to an n-gram language model? *[handwritten: product → numeric underflow]*

- How might one evaluate a language model? *[handwritten: evaluation]*

Get ahead

- Adjust the `03-classification` iPython notebook, so that the supervised machine learning model attempts to solve the **multi-class** problem, rather than the **single-class** problem (for `acq`). Does your assessment of the relative utility of the given classifiers change?

- Using the (short) "corpus" from Discussion Q2, generate all of the sentences of length 3. Choose an n-gram language model, and find the most probable sentence. What about length 4? 5? 6? What do you notice about these sentences? Does smoothing (or not) change this?

- Modify the iPython notebook so that it uses back–off smoothing. How does this change the probability of the given sentence? Why? Is the perplexity of this model better than Laplacian smoothing? Interpolation? Why?

- Perform the Programming experiments above using different corpora.