Week 5 lab

Sentiment classification

We are going to explore Naive Bayes classification with a dataset of positive and negative movie reviews. The reviews, collected by Bo Pang and Lillian Lee (distributed under CC-BY licence), can be accessed:

- through NLTK, e.g. `[nltk.corpus.movie_reviews.raw(fileid) for fileid in nltk.corpus.movie_reviews.fileids(categories='neg')]` (Use `nltk.download('movie_reviews')` the first time.)
- from the original publishers: **"polarity dataset v2.0" (http://www.cs.cornell.edu/people/pabo/movie-review-data/)**.

The dataset is distributed as a set of pre-tokenised and lowercased text files in directories `pos` and `neg`.

0. Naive Bayes recap

Bayesian classification is about using Bayes rule to classify an object:

$$\begin{aligned} y^*(x) &= \mathbf{argmax}_y P(y \mid x) \\ &= \mathbf{argmax}_y P(x|y)P(y) \end{aligned}$$

We therefore need an estimate of the prior probability of each class, $P(y)$, and an estimate of the probability of an instance x given a class y. That is, we can model each class separately.

The tricky part is defining $P(x \mid y)$. In Naive Bayes classification, we naively assume x can be represented as a feature vector $< x_1, x_2, \ldots, x_m >$, and that each feature is conditionally independent given y. Thus $P(x \mid y) = \prod_i P(x_i \mid y)$.

Naive Bayes can be an **effective classifier** for text. And it is extremely **easy to train**, since we just estimate a separate model of each feature for the subset of training documents in each class.

1. Multinomial Naive Bayes

The **provided script** (https://github.sydney.edu.au/COMP5046-Natural-Language-Processing/comp5046-labs-2018/blob/master/lab05/classifier-todo.py) will perform bag-of-words Multinomial Naive Bayes classification, except for the important part: calculating the probability of the document conditioned on each class. Complete the script, or write your own, to calculate the joint probability of document and class.

The model hypothesises that the features of a document are drawn from a multinomial distribution such that the probability of seeing each token is fixed (given the class). If a word i occurs c_{iy} times in training, then the MLE for its occurrence is:

$$p_{iy} = \frac{c_{iy}}{\sum_{i'} c_{i'y}}$$

If each x_i is the number of times word i appears in document x the conditional probability of x is:

$$f(x|y) = \prod_i (p_{iy})^{x_i}$$

In practice, we accumulate probabilities in log space for numerical stability:

$$logP(x|y) = \sum_i x_i \log p_{iy}$$

Note that we don't need to explicitly account for the contribution of features with value 0, since $p_{iy} = 1$. (Note: This is not the case for Bernoulli Naive Bayes models!)

You will quickly find documents assigned 0 probability due to maximum likelihood estimation of p_{iy} . Implement Laplace/Lidstone smoothing, which should make your classifier act as if there were a fake document in each class including all features λ times. Note this should not affect the prior distribution of classes $P(y)$ which should remain at the MLE.

$$p_{iy} = \frac{c_{iy}+\lambda}{\sum_{i'}(c_{i'y}+\lambda)}$$

Work in groups to complete the calculation of the log probability of the document given each class.

2. Feature engineering

Working in groups, modify the script to support arbitrary count or binary features (not just bag of all words). Choose from the following list to experiment with features beyond bag of words, and evaluate the effecto on accuracy:

i. just the presence of words "best" and "worst"

ii. use binary presence of words rather than their counts

iii. bag of bigrams (with or without unigrams)

iv. remove the most frequent words or known stop words (function words, e.g. from `nltk_data/corpora/stopwords/english`)

v. bag of words ignoring punctuation / bag of punctuation only

vi. bag of words ignoring infrequent words

vii. adding a feature which is completely random to bag of words

viii. using a **Porter Stemmer** **(https://tartarus.org/martin/PorterStemmer/)** to discard inflectional morphology

ix. only words from the first sentence or last sentence

x. a feature based on how long words or sentences are

2.1. What kinds of features reduce errors? Do they do so by improving the discrimination or the reliability of the model?

2.2. Examine examples of errors produced by bag of words and each further model. Suggest features which may help to further reduce errors.

2.3. Extension: How sensitive is Naive Bayes to noisy and redundant features? Can you make Naive Bayes give too much weight to some features because of its assumption that features are independent given the class? For example, you might include features like {unigram, lowercase unigram, unigram's stem, bigram} altogether in one model. These include redundant information. Thus the features are very non-independent.

3. Evaluating on a held-out set

The script above splits the data into training (80%) and test (20%) portions. This allows us to evaluate a learnt classifier on data that was not used when training the model, estimating its *generalisation error*. We are able to get an estimate of how much generalisation error varies by repeating such training-testing splits in the cross validation process.

Working in groups:

3.1. Try calculating performance on the training set (set `train = test`) as well as the test set. How much does evaluating on the training set over-estimate the model's predictive performance?

3.2. For the same feature extractors, calculate the standard deviation in error count across 10 shuffles of the data? (Do some feature sets or smoothing approaches reduce the variance more than others?)

3.3. Generally, one must be very careful about information from an evaluation dataset being leaked into training the model; such *leakage* produces skewed evaluation results. Usually we want our models to be usable on unseen datasets, which means that even very basic statistics such as document frequency learnt from the test set can be problematic. How can such leakage be avoided?

Copyright © The University of Sydney. Unless otherwise indicated, 3rd party material has been reproduced and communicated to you by or on behalf of the University of Sydney in accordance with section 113P of the Copyright Act 1968 (Act). The material in this communication may be subject to copyright under the Act. Any further

reproduction or communication of this material by you may be the subject of copyright protection under the Act.