Dataset

Your dataset is a preprocessed and modified subset of the Twenty Newsgroups Data Set [1]. It is based on 2000 real email messages from a newsgroup mailing list. Emails have been preprocessed in the following ways:

- **Stop word removal:** Words like "and","the", and "of", are very common in all English sentences and are therefore not very predictive in deciding the newsgroup. These words have been removed from the emails.
- **Removal of non-words:** Numbers and punctuation have both been removed. All white spaces (tabs, newlines, spaces) have all been trimmed to a single space character

The data has been already split into two subsets: a 1600-email subset for training and a 400-email subset for testing (consider this as your validation set and imagine there is another test set which is not given to you). The features have been generated for you. You will use the following files:

- `question-4-train-features.csv`
- `question-4-train-labels.csv`
- `question-4-test-features.csv`
- `question-4-test-labels.csv`

The files that ends with `features.csv` contains the features and the files ending with `labels.csv` contains the ground truth labels.

In the feature files each row contains the feature vector for an email. The j-th term in a row i is the number of occurrences of the j-th vocabulary word in the i-th email. The size of the vocabulary is 26507. The label files include the ground truth label for the corresponding email, the order of the emails (rows) are the same as the features file. That is the i-th row in the files corresponds to the same email document. The labels are indicated by 1 or 0, 1 stands for an email coming from space newsgroup and 0 stands for an email belonging to medical newsgroup.

Bag-of-Words Representation and Multinomial Naive Bayes Model

Recall the bag-of-words document representation makes the assumption that the probability that a word appears in email is conditionally independent of the word position given the class of the email. If we have a particular email document D_i with n_i words in it, we can compute the probability that D_i comes from the class y_k as:

$$\mathbf{P}\left(D_i \,|\, Y = y_k\right) = \mathbf{P}\left(X_1 = x_1, X_2 = x_2, .., X_{n_i} = x_{n_i} \,|\, Y = y_k\right) = \prod_{j=1}^{n_i} \mathbf{P}\left(X_j = x_j \,|\, Y = y_k\right) \tag{4.1}$$

In Eq. (4.1), X_j represents the j^{th} position in email D_i and x_j represents the actual word that appears in the j^{th} position in the email, whereas n_i represents the number of positions in the email. As a concrete example, we might have the first email document (D_1) which contains 200 words ($n_1 = 200$). The document might be of space email ($y_k = 1$) and the 15^{th} position in the email might have the word "apollo" ($x_j =$ "apollo").

In the above formulation, the feature vector $\vec{X}$ has a length that depends on the number of words in the email n_i. That means that the feature vector for each email will be of different sizes. Also, the above formal definition of a feature vector $\vec{x}$ for a email says that $x_j = k$ if the j-th word in this email is the k-th word in the dictionary. This does not exactly match our feature files, where the j-th term in a row i is the number of occurrences of the j-th dictionary word in that email i. As shown in the lecture slides, we can slightly change the representation, which makes it easier to implement:

$$\mathbf{P}\left(D_i \,|\, Y = y_k\right) = \prod_{j=1}^{V} \mathbf{P}\left(X_j \,|\, Y = y_k\right)^{t_{w_j,i}} \tag{4.2}$$

,where V is the size of the vocabulary, X_j represents the appearing of the j-th vocabulary word and $t_{w_j,i}$ denotes how many times word w_j appears in email D_i. As a concrete example, we might have a vocabulary of size of 1309, $V = 1309$. The first email (D_1) might be from space newsgroup ($y_k = 1$) and the 80-th word in the vocabulary, w_{80}, is "moon" and $t_{w_{80},1} = 2$, which says the word "moon" appears 2 times in email D_1. Contemplate on why these two models (Eq. (4.1) and Eq. (4.2)) are equivalent.

In the classification problem, we are interested in the probability distribution over the email classes (in this case space newsgroup and medical newsgroup) given a particular email D_i. We can use Bayes Rule to write:

$$\mathbf{P}\left(Y = y_k | D_i\right) = \frac{\mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{V} \mathbf{P}\left(X_j \,|\, Y = y\right)^{t_{w_j,i}}}{\sum_k \mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{V} \mathbf{P}\left(X_j \,|\, Y = y_k\right)^{t_{w_j,i}}} \tag{4.3}$$

Note that, for the purposes of classification, we can actually ignore the denominator here and write:

$$\mathbf{P}\left(Y = y_k | D_i\right) \propto \mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{V} \mathbf{P}\left(X_j \,|\, Y = y\right)^{t_{w_j,i}} \tag{4.4}$$

$$\hat{y}_i = \arg\max_{y_k} \mathbf{P}\left(Y = y_k \,|\, D_i\right) = \arg\max_{y_k} \mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{V} \mathbf{P}\left(X_j \,|\, Y = y_k\right)^{t_{w_j,i}} \tag{4.5}$$

Question 4.1 [2 points] Why it is that we can ignore the denominator?

Probabilities are floating point numbers between 0 and 1, so when you are programming it is usually not a good idea to use actual probability values as this might cause numerical underflow issues. As the logarithm is a strictly monotonic function on [0,1] and all of the inputs are probabilities that must lie in [0,1], it does not have an affect on which of the classes achieves a maximum. Taking the logarithm gives us:

$$\hat{y}_i = \arg\max_{y} \left(\log \mathbf{P}\left(Y = y_k\right) + \sum_{j=1}^{V} t_{w_j,i} * \log \mathbf{P}\left(X_j \,|\, Y = y_k\right) \right) \tag{4.6}$$

, where $\hat{y}_i$ is the predicted label for the i-th example.

Question 4.2 [3 points] If the the ratio of the classes in a dataset is close to each other, it is a called "balanced" class distribution if not it is skewed. What is the percentage of space emails in the `train.labels.txt`. Is the training set balanced or skewed towards a one of the classes?

The parameters to learn and their MLE estimators are as follows:

$$\theta_{j \,|\, y=0} \equiv \frac{T_{j,y=0}}{\sum_{j=1}^{V} T_{j,y=0}}$$

$$\theta_{j \,|\, y=1} \equiv \frac{T_{j,y=1}}{\sum_{j=1}^{V} T_{j,y=1}}$$

$$\pi_{y=1} \equiv \mathbf{P}\left(Y = 1\right) = \frac{N_1}{N}$$

- $T_{j,0}$ is the number of occurrences of the word j in medical emails in the training set including the multiple occurrences of the word in a single email.
- $T_{j,1}$ is the number of occurrences of the word j in space emails in the training set including the multiple occurrences of the word in a single email.
- N_1 is the number of space emails in the training set.
- N is the total number of emails in the training set.
- $\pi_{y=1}$ estimates the probability that any particular email will be a space email.

- $\theta_{j\,|\,y=0}$ estimates the probability that a particular word in a medical email will be the j-th word of the vocabulary, $\mathbf{P}\left(X_j \,|\, Y=0\right)$
- $\theta_{j\,|\,y=1}$ estimates the probability that a particular word in a space email will be the j-th word of the vocabulary, $\mathbf{P}\left(X_j \,|\, Y=1\right)$

Question 4.3 [5 points] How many parameters do we need to estimate for this model?

Question 4.4 (Coding) [30 points] Train a Naive Bayes classifier using all of the data in the training set (`train-features.csv` and `train-labels.csv`). Test your classifier on the test data (`test-features.csv` and `test-labels.csv`, and report the testing accuracy as well as how many wrong predictions were made. In estimating the model parameters use the above MLE estimator. If it arises in your code, define $0 * \log 0 = 0$ (note that $a * \log 0$ is as it is, that is -inf). In case of ties, you should predict "medical". In the written part of your report what your test set accuracy is? What did your classifier end up predicting? Why is using the MLE estimate is a bad idea in this situation?

Question 4.5 (Coding) [5 points] Extend your classifier so that it can compute an MAP estimate of θ parameters using add-one smoothing technique. This new prior "hallucinates" that each word appears additional 1 time in the train set(In your final submission, your code shall run with these new parameters, not with the parameters in question 4.4).

$$\theta_{j\,|\,y=0} \equiv \frac{T_{j,y=0}+1}{\sum_{j=1}^{V} T_{j,y=0+V}}$$
$$\theta_{j\,|\,y=1} \equiv \frac{T_{j,y=1}+1}{\sum_{j=1}^{V} T_{j,y=1+V}}$$
$$\pi_{y=1} \equiv \mathbf{P}\left(Y=1\right) = \frac{N_1}{N}$$

Train your classifier using all of the training set and have it classify all of the test set and report test-set classification accuracy using add-one smoothing technique.

Question 4.6 (Coding) [10 points] Rank the features by calculating mutual information between the class variable and each feature. Write indices and mutual information scores of features in descending order.

Question 4.7 (Coding) [10 points] Remove features from the full model one-by-one starting from the least informative one. Keep removing until a single feature remains. Plot test-set accuracy as a function of removed number of features and report the maximum accuracy that you get.

References

1. Twenty Newsgroups dataset. https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups
2. "On Discriminative vs. Generative Classifiers: A comparison of logistic regression and Naive Bayes" by Andrew Ng and Michael I. Jordan.
3. CMU Lecture Notes.
http://www.cs.cmu.edu/~epxing/Class/10701-10s/Lecture/lecture5.pdf