CSCI-GA 2590: Natural Language Processing

Representing and Classifying Text

Hemant Ramawat
hr2378

Collaborators:
By turning in this assignment, I agree by the honor code of the College of Arts and Science at New York University and declare that all of this is my own work.

Welcome to your first assignment! The goal of this assignment is to get familiar with basic text classification models through both mathematical analysis and hands-on feature engineering. **Before you get started, please read the Submission section thoroughly**.

Submission

Submission is done on Gradescope.

Written: When submitting the written parts, make sure to select **all** the pages that contain part of your answer for that problem, or else you will not get credit. You can either directly type your solution between the `shaded` environments in the released `.tex` file, or write your solution using pen or stylus. A `.pdf` file must be submitted.
Programming: Questions marked with “coding” next to the assigned to the points require a coding part in `submission.py`. Submit `submission.py` and we will run an autograder on Gradescope. You can use functions in `util.py`. However, please do not import additional libraries (e.g. `numpy`, `sklearn`) that aren’t mentioned in the assignment, otherwise the grader may crash and no credit will be given. You can run `test.py` to test your code but you don’t need to submit it.

Problem 1: Naive Bayes classifier

In this problem, we will study the *decision boundary* of multinomial Naive Bayes model for binary text classification. The decision boundary is often specificed as the level set of a function: $\{x \in \mathcal{X} : h(x) = 0\}$, where x for which $h(x) > 0$ is in the positive class and x for which $h(x) < 0$ is in the negative class.

1. [2 points] Give an expression of $h(x)$ for the Naive Bayes model $p_\theta(y \mid x)$, where θ denotes the parameters of the model.

At decision boundary, $P(y=1|x) = P(y=0|x)$

$\Rightarrow \dfrac{P(y=1|x)}{P(y=0|x)} = 1$ $\Rightarrow$ taking log both side $\Rightarrow \log(P(y=1|x)) - \log(P(y=0|x)) = 0$

$$\boxed{\therefore\ h(x) = \log(P(y=1|x)) - \log(P(y=0|x))}$$

2. [3 points] Recall that for multinomial Naive Bayes, we have the input $X = (X_1, \dots, X_n)$ where n is the number of words in an example. In general, n changes with each example but we can ignore that for now. We assume that $X_i \mid Y = y \sim \text{Categorical}(\theta_{w_1,y}, \dots \theta_{w_m,y})$ where $Y \in \{0,1\}$, $w_i \in \mathcal{V}$, and $m = |\mathcal{V}|$ is the vocabulary size. Further, $Y \sim \text{Bernoulli}(\theta_1)$. Show that the multinomial Naive Bayes model has a linear decision boundary, i.e. show that $h(x)$ can be written in the form $w \cdot x + b = 0$. [**RECALL:** The categorical distribution is a multinomial distribution with one trial. Its PMF is

$$p(x_1, \dots, x_m) = \prod_{i=1}^{m} \theta_i^{x_i} \ ,$$

where $x_i = \mathbb{1}[x = i]$, $\sum_{i=1}^m x_i = 1$, and $\sum_{i=1}^m \theta_i = 1$.]

$$\log P(x^i, y^i; \theta, \alpha) = \log P(x^i | y^i; \theta) \ P(y^i, \alpha)$$

and $P(x^i | y^i; \theta) \sim \text{Categorical}(\theta_{w_1,y}, \dots, \theta_{w_m,y})$

$$\sim p(x_1, \dots, x_m) = \prod_{i=1}^{m} \theta_i^{x_i}$$

$$h(x) = \log(P(y=1|x)) - \log(P(y=0|x))$$

$$\Rightarrow h(x) \sim \log\left(\prod_{i=1}^{m} \theta_{i_1}^{x_i} \cdot P(y_i=1, \alpha)\right) - \log\left(\prod_{i=1}^{m} \theta_{i_0}^{x_i} \cdot P(y_i=0, \alpha)\right)$$

$$= \log\left(\prod_{i=1}^{m} \theta_{i_1}^{x_i}\right) - \log\left(\prod_{i=1}^{m} \theta_{i_0}^{x_i}\right) + \log(P(y_i=1, \alpha)) - \log(P(y_i=0, \alpha))$$

$$= \sum_{i=1}^{m} x_{i_1}\left[\log(\theta_{i_1}) - \log(\theta_{i_0})\right] + \left[\log P(y_i=1, \alpha) - \log(P(y_i=0, \alpha))\right]$$

$$= x_i \cdot \sum_{i=1}^{m} \left[\log(\theta_{i_1}) - \log(\theta_{i_0})\right] + \left[\log P(y_i=1, \alpha) - \log(P(y_i=0, \alpha))\right]$$

$$\Rightarrow h(x) = x \cdot w + b$$

where $w = \sum_{i=1}^{m} \left[\log(\theta_{i_1}) - \log(\theta_{i_0})\right]$

$$b = \left[\log P(y_i=1, \alpha) - \log(P(y_i=0, \alpha))\right]$$

3. [2 points] In the above model, X_i represents a single word, i.e. it's a unigram model. Think of an example in text classification where the Naive Bayes assumption might be violated. How would you allieviate the problem?

4. [2 points] Since the decision boundary is linear, the Naive Bayes model works well if the data is linearly separable. Discuss ways to make text data more separable and its influence on model generalization.

3) Example :
sentence 1 → I like bacon but not eggs
sentence 2 → I like eggs but not bacon
the problem is that we miss out on the Contentual use of the word.
This could possibly be solved by combining more words in tokens.
Like using a bi-gram model using 2words

4) Data could be made more separable by adding more dimensions to it and then applying Feature Selection or Feature extraction to improve accuracy and reduce the risk of overfitting.

Problem 2: Skip-gram model

In this problem, we will gain some insights into the skip-gram model. In particular, we will show that it encourages the inner product between two word embeddings to be equal to their pointwise mutual information (PMI) (up to a constant).

Recall that to evaluate the objective function of the skip-gram model we need to enumerate over the entire vocabulary $\mathcal{V}$, which can be quite expensive in practice. Note that the reason we need to enumrate the vocabulary is to obtain a probability distribution of neighboring words.

Instead of modeling the distribution of words, we can model the distribution of an indicator: whether two words are neighbors. Let Y be the indicator random variable. We model it by a Bernoulli distribution

$$p_\theta(y = 1 \mid w, c) = \frac{1}{1 + e^{-u_c \cdot v_w}} \ ,$$

where $c \in \mathcal{V}$ is within a window centered at $w \in \mathcal{V}$, u_c, v_w represent embeddings of c and w, and θ is the model parameter, i.e. word embedding v's and context embedding u's.

What about the negative observations ($y = 0$)? A naive way is to consider all words that do not occur in the neighborhood of w, however, this is as expensive as our original objective. One solution is **negative sampling**, where we only consider a small number of non-neighboring words.

Let $D = \{(w, c)\}$ be the set of all word-neighbor pairs observed in the training set. For each pair (w, c), we generate k negative neighbors c_N by sampling from a distribution $p_N(\cdot)$ over the vocabulary.

1. [3 points] Let L be the learning objective that maximizes the log-likelihood of D and the *expected* log-likelihood of the negative examples. Note that C_N is the random variable in the expectation. Show that

$$L = \max_\theta \sum_{(w,c)\in D} \log \frac{1}{1 + e^{-u_c \cdot v_w}} + k \sum_{(w,\cdot)\in D} \sum_{c_N \in \mathcal{V}} p_N(c_N) \log \frac{1}{1 + e^{u_{c_N} \cdot v_w}} \tag{1}$$

From softmax function, $p(w|c) = \dfrac{e^{h^T v_w'}}{\sum_{w, \in V} e^{h^T v_m'}}$

The loss function, $j(\theta) = -\log \dfrac{e^{vh}}{\sum e^{vh}}$

$$= -h^t v + \log \sum e^{(h^t v)}$$

$$L = -J(\theta) = \sum_{w \in V} \left(\log P(y=1|w,c) \right) - k E [P(y=0|w,c]$$

$$= \max_{\theta} \sum_{(w,c) \in D} \log \frac{1}{1+e^{-u_c v_w}} + k \sum_{(w,\cdot) \in D} \sum_{c_N \in V} P_N(c_N) \log \frac{1}{1+e^{u_{c_N} v_w}}$$

2. [2 points] Let

$$\sigma(\alpha) = \frac{1}{1+e^{-\alpha}} \,.$$

Show that

$$\frac{d}{d\alpha} \log \sigma(\alpha) = \frac{1}{1+e^{\alpha}} \,.$$

$$\frac{d}{d\alpha} \log \sigma(\alpha) = \frac{1}{\sigma(\alpha)} \frac{d}{d\alpha} \sigma(\alpha)$$

$$= \frac{1}{1+e^{-\alpha}} \frac{d}{d\alpha} \left(\frac{1}{1+e^{\alpha}} \right)$$

$$= \frac{1}{1+e^{-\alpha}} (-1) \frac{d}{d\alpha} \left(1+e^{-\alpha} \right)$$

$$= \frac{-1}{1+e^{-\alpha}} \left[0 + \frac{d}{d\alpha} \left(e^{-\alpha} \right) \right]$$

$$= \frac{-1}{1+e^{-\alpha}} \left[-e^{-\alpha} \right]$$

$$= \frac{e^{-\alpha}}{1+e^{-\alpha}}$$

$$= \frac{1}{e^{\alpha}+1}$$

$$\Rightarrow \frac{d}{d\alpha} \log \sigma(\alpha) = \frac{1}{1+e^{\alpha}}$$

3. [3 points] Let's consider one pair $(w = b, c = a)$. Let $\alpha = u_a \cdot v_b$. Show that the optimal solution of α is
$$\alpha^* = \log \frac{A}{B} ,$$
where
$$A = \sum_{(w,c)\in D} \mathbb{1}[w = b, c = a] \tag{2}$$
$$B = k \sum_{(b,\cdot)\in D} p_N(a) . \tag{3}$$
[**HINT:** Let $\ell(\theta)$ be the objective. Solve for α in $\frac{\partial \ell}{\partial \alpha} = 0$, carefully considering what the partial derivative of objective w.r.t. α will be. You can use results from previous questions.]

We know,

$$L = \max_{\theta} \sum_{(w,c)\in D} \log \frac{1}{1+e^{-u_c \cdot v_w}} + K \sum_{(w,\cdot)\in D} \sum_{c_N \in V} P_N(c_N) \log \frac{1}{1+e^{u_{c_N} \cdot v_w}}$$

for one pair $(w = b, c = a)$ and $\alpha = u_a \cdot v_b$

$$\ell(\theta) = \sum_{(b,c)\in D} \log \frac{1}{1+e^{-u_a \cdot v_b}} + K \sum_{(b,\cdot)\in D} P_N\, a \log \frac{1}{1+e^{u_a \cdot v_b}}$$

$$\Rightarrow \ell(\theta) = 1 \cdot \log \frac{1}{1+e^{-\alpha}} + K \sum_{(b,\cdot)\in D} P_N\, a \log \frac{1}{1+e^{\alpha}}$$

Solving for α in $\frac{\partial \ell}{\partial \alpha} = 0$

$$\Longrightarrow \frac{1}{1+e^{\alpha}} + K \sum_{(b,\cdot)\in D} P_N\, a \left(- \frac{1}{1+e^{-\alpha}} \right) = 0$$

$$\Rightarrow \frac{1}{1+e^{\alpha}} = K \sum_{(b,\cdot)\in D} P_N\, a \left(\frac{1}{1+e^{-\alpha}} \right)$$

$$\Rightarrow \frac{1}{1+e^{\alpha}} = K \sum_{(b,\cdot)\in D} P_N(a) \left(\frac{e^{\alpha}}{1+e^{\alpha}} \right) \Rightarrow e^{\alpha} = \frac{1}{K \sum_{(b,\cdot)\in D} P_N(a)}$$

$$\Rightarrow \alpha = \ln \frac{1}{K \sum_{(b,\cdot)\in D} P_N(a)} = \ln \frac{A}{B}$$

4. [2 points] Suppose the distribution of negative words $p_N(w)$ for $w \in \mathcal{V}$ is a categorical distribution given by

$$p_N(w) = \frac{\text{count}(w, \cdot, D)}{|D|} ,$$

where $\text{count}(w, \cdot, D) = \sum_{c \in \mathcal{V}} \text{count}(w, c, D)$. Note that $p_N(w)$ is simply the fraction of unigram w in D. Show that

$$\alpha^* = \text{PMI}(b, a) - \log k ,$$

where the PMI score of word co-occurence in D is defined as

$$\text{PMI}(b, a) \overset{\text{def}}{=} \log \frac{\text{count}(b, a, D)|D|}{\text{count}(b, \cdot, D)\text{count}(a, \cdot, D)} .$$

[**HINT:** Think about how to express A and B using count.]

From last part, $\alpha^* = \log \frac{A}{B}$, where $A = \sum 1 \, [w = b, v = a]$ and
$B = K \sum_N P_N(a)$

$$\Rightarrow \quad \alpha^* = \log \frac{A}{K \sum P_N(a)}$$

$$= \log A - \log \sum_N P_N(a) - \log K$$

$$= \log(\text{count}(b, a, D)) - \log\left(\sum_N \frac{\text{count}(a, \cdot, D)}{|D|}\right) - \log K$$

$$= \log(\text{count}(b, a, D)) - \log\left(\text{count}(b, \cdot, D) . \frac{\text{count}(a, \cdot, D)}{|D|}\right) - \log K$$

$$= \log\left[\frac{\text{count}(b, a, D) \times |D|}{\text{count}(b, \cdot, D) \times \text{count}(a, \cdot, D)}\right] - \log K$$

$$\Rightarrow \alpha^* = \text{PMI}(b, a) - \log K$$

where,

$$\text{PMI}(b, a) = \log\left[\frac{\text{count}(b, a, D) \times |D|}{\text{count}(b, \cdot, D) \times \text{count}(a, \cdot, D)}\right]$$

5. [2 points] Let's denote the unigram model defined above by $p_{\text{unigram}}(w)$. In practice, we often prefer to use $p_N(w) \propto p^{\beta}_{\text{unigram}}(w)$ where $\beta \in [0, +\infty]$. Describe the desired range of β and explain why.

The desired range for β is $(0,1)$. Because we would want to assign higher values to non significant negative words.

Basically, we are trying to reduce the noise as much as possible which will help us to identify target words better

Problem 3: Natural language inference

In this problem, you will build a logistic regression model for textual entailment. Given a *premise* sentence, and a *hypothesis* sentence, we would like to predict whether the hypothesis is *entailed* by the premise, i.e. if the premise is true, then the hypothesis must be true.
Example:

label	premise	hypothesis
entailment	The kids are playing in the park	The kids are playing
non-entailment	The kids are playing in the park	The kids are happy

1. [1 point] Given a dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ where $y \in \{0, 1\}$, let ϕ be the feature extractor and w be the weight vector. Write the maximum log-likelihood objective as a *minimization* problem. [**HINT:** Start with $\min_w - \sum_{i=1}^{n} \log p(y^{(i)} \mid x^{(i)}; w)$]

$$L(w) = \sum_{i=1}^{N} \log p(y^{(i)} | x^{(i)}, w)$$

max = -min

$$= \sum_{i=1}^{N} y^{(i)} \log \frac{1}{1+e^{-w\phi(x^{i})}} + (1-y^{(i)}) \log \left[1 - \frac{1}{1+e^{-w\phi(x^{i})}}\right]$$

2. [3 point, coding] We first need to decide the features to represent x. Implement `extract_unigram_features` which returns a BoW feature vector for the premise and the hypothesis.

3. [2 point] Let $\ell(w)$ be the objective you obtained above. Compute the gradient of $\ell_i(w)$ given a single example $(x^{(i)}, y^{(i)})$. Note that $\ell(w) = \sum_{i=1}^{n} \ell_i(w)$. You can use $f_w(x) = \frac{1}{1+e^{-w\cdot\phi(x)}}$ to simplify the expression.

$$L_i(w) = \sum_{i=1}^{N} y^{(i)} \log \frac{1}{1+e^{-w\,\phi(x)}} + (1-y^{i}) \log \left[1 - \frac{1}{1+e^{-w\,\phi(x)}}\right]$$

if $f_w(x) = \dfrac{1}{1+e^{-w\cdot\phi(x)}}$

$$L_i(w) = \sum_{i=1}^{W} y^{(i)} \log\left(f_w(x)\right) + (1-y^{i}) \log\left[1 - f_w(x)\right]$$

$$\frac{d}{dw}\left[\log\left(1 - f_w(x)\right)\right] = \frac{d}{dw}\left[\log\left(\frac{1}{1+e^{-w\,\phi(x)}}\right)\right] = \left(1+e^{w\phi(x)}\right) \frac{d}{dw}\left(\frac{1}{1+e^{w\phi(x)}}\right)$$

$$= \left(1+e^{w\phi(x)}\right)(-1)\left(\frac{1}{1+e^{w\phi(x)}}\right) \cdot e^{w\phi(x)} \cdot \phi(x)$$

$$= -\frac{e^{w\phi(x)} \cdot \phi(x)}{1+e^{w\cdot\phi(x)}} = -\phi(x) \frac{1}{e^{w\phi(x)}+1} = -\phi(x)\, f_w(x)$$

$$\frac{d}{dw} \log\left(f_w(x)\right) = \phi(x)\left(1 - f_w(x)\right)$$

For given single example, $L_i(w) = \cdot 1\left[y_i = 1\right] \log\left(f_w(x)\right) + 1\left[y_i = 0\right] \log\left(1 - f_w(x)\right)$

$$\frac{\partial\left(\ell_i(w)\right)}{\partial w} = 1\left[y_i = 1\right]\left(1 - f_w(x)\right)\phi(x) + \left[1 \cdot \left[y_i = 0\right) \cdot -\phi(x)\, f_w(x)\right]$$

$$= 1\left(y_i = 1\right)\left(1 - f_w(x)\right)\phi(x) - 1\left(y_i = 0\right) f_w(x)\,\phi(x)$$

4. [5 points, coding] Use the gradient you derived above to implement `learn_predictor`. You must obtain an error rate less than 0.3 on the training set and less than 0.4 on the test set to get full credit *using the unigram feature extractor*.

5. [3 points] Discuss what are the potential problems with the unigram feature extractor and describe your design of a better feature extractor.

Unigram feature extractor lacks dimensions and also misses out on content. Another potential problem is encountering unigrams

Potential Problems with unigram feature extractor

→ Lacks dimensions

→ misses out on content

→ Dealing with unknown unigrams

For the lack of dimensions and Content we add below ngrams to the vocabulary:

- Bigrams, trigrams or higher count ngrams depending on the language
- Add information about the source of the word, for example, is it from hypothesis
- Add roots of the words to the vocabulary. For example, eating → eat

For dealing with unknown unigrams/ngrams

→ We can keep one single default token for all new/unknown ngrams

→ we could have default value for any new ngram

6. [3 points, coding] Implement your feature extractor in `extract_custom_features`. You must get a lower error rate on the dev set than what you got using the unigram feature extractor to get full credits.

7. [3 points] When you run the tests in `test.py`, it will output a file `error_analysis.txt`. (You may want to take a look at the `error_analysis` function in `util.py`). Select five examples misclassified by the classifier using custom features. For each example, briefly state your intuition for why the classifier got it wrong, and what information about the example will be needed to get it right.

```
1. P: A man in a denim shirt and pants is smoking a cigarette while playing a cello for money .
H: A man in a denim shirt is smoking .
label=1, predicted=0, wrong
f=a, v=4, w=-0.052640342871420234, v*w=-0.21056137148568094
f=for, v=1, w=-0.8329942931414186, v*w=-0.8329942931414186

2. P: A group of people walk up a ramp on a cloudy day .
H: The people are walking .
label=1, predicted=0, wrong
f=a, v=2, w=-0.052640342871420234, v*w=-0.10528068574284047
f=are, v=1, w=-0.18688313226308337, v*w=-0.18688313226308337
f=The, v=1, w=-0.4043669422071161, v*w=-0.4043669422071161

3. P: There is not an image here .
H: There is only a single color displayed instead of an image .
label=0, predicted=1, wrong
f=image, v=2, w=-0.031599260739242574, v*w=-0.06319852147848515

4. P: A man and a woman are sitting at a table outside , next to a small flower garden .
H: The man and woman are on a date .
label=0, predicted=1, wrong
f=a, v=4, w=-0.052640342871420234, v*w=-0.21056137148568094
f=date, v=1, w=-0.30274550326002836, v*w=-0.30274550326002836
f=are, v=2, w=-0.18688313226308337, v*w=-0.37376626452616674
f=to, v=1, w=-0.4034652611792073, v*w=-0.4034652611792073
f=The, v=1, w=-0.4043669422071161, v*w=-0.4043669422071161

5. P: A man with blond-hair and tattoos strolls behind bleachers on the cellphone .
H: A man talks on his cell phone .
label=1, predicted=0, wrong
f=his, v=1, w=-0.39929418737301636, v*w=-0.39929418737301636
```

For examples 1, 2, 4 and 5 we can higher values (-ve or +ve) for insignificant words like 'a', 'are', 'The', etc which badly impacts the final prediction. These insignificant words could be handled better by ignoring them or scoring them lower

Similarly, for example 3, we can see that a significant word "image" has a lower value thereby contributing less than what its should be. Such significant words should be scored higher.

8. [3 points] Change `extract_unigram_features` such that it only extracts features from the hypothesis. How does it affect the accuracy? Is this what you expected? If yes, explain why. If not, give some hypothesis for why this is happening. Don't forget to change the function back before your submit. You do not need to submit your code for this part.

By only extracting features from the hypothesis we observe that the accuracy improves. Ideally this would not be an expected outcome as we reduced some information from the features.

This could have happened because of the nature and quality of our data. This means this observation would be a case of an exception where removing information resulted in a more accurate outcome.