Midterm practice questions
[with partial solutions]

UMass CS 485, Fall 2023
This version: 11/2

1 Midterm information

The midterm will be in-class during the normal time, on Tuesday 11/7. It's closed-book, except you can bring a "cheat sheet," one page of notes (front and back is fine) that you write for yourself. Typically the act of writing the notes can be a useful study aid!

2 Topics on the midterm

Anything covered in class, readings, homeworks, or exercises can be on the midterm. We're more likely to focus on topics covered in class or referred to in class.

We recommend going through all exercises as practice problems—do them completely, if you haven't before.

Topics include the following:

Language concepts

- Regular expressions
- Text normalization, tokenization

Probability, language modeling, classification

- Relative frequency estimation and pseudocount smoothing
- N-gram (Markov) language models
- Naive Bayes
- Logistic regression, binary classification
- Logistic regression, multiclass classification
- *Note:* gradient derivations will not be included

Evaluation / Annotation

- Classification evaluation metrics: false positives/negatives, precision, recall, F1
- Annotator agreement rates

Syntax / Linguistic structure

- Part of speech tags
- BIO tagging
- Constituency grammars and parses
- Dependency parses
- Parsing (CKY)

3 Classification

Question 3.1. Consider training and predicting with a naive Bayes classifier for two document classes, and without pseudocounts. The word "booyah" appears once for class 1, and never for class 0. When predicting on new data, if the classifier sees "booyah", what is the posterior probability of class 1?
[Solution: 1]

Question 3.2. For a probabilistic classifier for a binary classification problem, consider the prediction rule to predict class 1 if $P(y = 1|x) > t$, and predict class 0 otherwise. This assumes some threshold t is set. If the threshold t is increased,

(a) Does precision tend to increase, decrease, or stay the same? **[Solution:** increase]

(b) Does recall tend to increase, decrease, or stay the same? **[Solution:** decrease]

Classification example

Here's a naive Bayes model with the following conditional probability table (each row is that class's unigram language model):

word type	a	b	c
$P(w \mid y = 1)$	$5/10$	$3/10$	$2/10$
$P(w \mid y = 0)$	$2/10$	$2/10$	$6/10$

and the following prior probabilities over classes:

$P(y=1)$	$P(y=0)$
$8/10$	$2/10$

Naive Bayes

Consider a binary classification problem, for whether a document is about the end of the world (class $y = 1$), or it is not about the end of the world (class $y = 0$).

Question 3.3. Consider a document consisting of 2 `a`'s, and 1 `c`.

Note: In this practice and on the midterm, you do not need to convert to decimal or simplify fractions. You may find it easier to not simplify the fractions. On the midterm, we will not penalize simple arithmetic errors. Please show your work.

(a) What is the probability that it is about the end of the world?

(b) What is the probability it is not about the end of the world?

Question 3.4. Now suppose that we know the document is about the end of the world ($y = 1$).

(a) True or False, the naive Bayes model is able to tell us the probability of seeing the document $\vec{w} = (a, a, b, c)$ under the model.

[Solution: True. The question is asking for $p(w|y)$ (or $p(x|y)$). Naive Bayes assumes the document words are generated according to a 0th order Markov model, a.k.a. a "bag of words" model.]

(b) If True, what is the probability?

[Solution: The numbers come from the first row of the $p(w|y)$ table.

$$p(aabc \mid y = 1) = [p(a|y = 1)]^2 p(b|y = 1)p(c|y = 1) = .5 \times .5 \times .3 \times .2 = 150/10000$$

]

Logistic Regression

Consider a logistic regression model for this same problem ($y = 1$ means the document is about the end of the world), with three features. The model has weights $\beta = (0.5, 0.25, 1)$.

Note: for this problem you will be exponentiating certain quantities. You do not need to write out your answer as a number, but instead in terms of exp() values, e.g., P = 1 + 2$\exp(-1)$.

Question 3.5. A given document has feature vector $x = (1, 4, 0)$.

(a) What is the probability that the document is about the end of the world ?

(b) What is the probability that it is not about the end of the world?

Question 3.6. Now suppose that we know the document is about sports ($y = 1$).

(a) True or False, the logistic regression model is able to tell us the probability of seeing $x = (1, 1, 2)$ under the model.

[**Solution:** False. Logistic regression only defines p $(y|x)$, not $p(x|y)$. So it cannot be used to calculate the probability of seeing a particular value of x under a particular class y (that is, $p(x|y)$).

More complicated answer: there's a way to apply Bayes Rule to logistic regression to reverse it back to the text generation direction as asked for, but you have to introduce new assumptions to make it work out, and the interpretation of what's going on is kinda funny. See our paper here: `https://aclanthology.org/D18-1487/`]

(b) If True, what is the probability? (again, answer in terms of $\exp()$ values).

Question 3.7. Consider a logistic regression model with weights $\beta = (\beta_1, \beta_2, \beta_3)$. A given document has feature vector $x = (1, -2, -1)$.

1. What is a value of the vector β such that the probability of the document being about the end of the world is 1 (or incredibly close)? [**Solution:** $(1000, 0, 0)$ will do. Causes a large $z = \beta' x$ and thus large $p(y = 1 \mid x) = 1/(1 + \exp(-1000)$ and e^{-1000} is really tiny!]

2. What is a value of the vector β such that the probability of the document being about the end of the world is 0 (or incredibly close)? [**Solution:** $(-1000, 0, 0)$ will do.]

Question 3.8. Show the two standard definitions of the logistic sigmoid function are equivalent.

Question 3.9. In Naive Bayes, if you increase the pseudocount hyperparameter, does your model tend to underfit or overfit more? [**Solution:** underfit]

Question 3.10. In logistic regression, if you increase the L2 norm regularization hyperparameter, does your model tend to underfit or overfit more? [**Solution:** underfit. This is referring to the λ when the learning problem is penalized likelihood learning, either $\max_\beta loglik(\beta) - \lambda(\sum_j \beta_j^2)^2$ or $\min_\beta -loglik(\beta) + \lambda(\sum_j \beta_j^2)^2$ and $loglik(\beta) = \sum_i \log p_\beta(y_i \mid x_i)$.]

4 Evaluation

Question 4.1. INLP chapter 4, Exercise 2 [Note: too hard to be a test question]

Question 4.2. INLP chapter 4, Exercise 3

5 Misc

Question 5.1. Please write a regular expression to match any word that has 3 or more instances of the same vowel in a row, like *sooooo* or *haaaaha*. (Assume there are 5 vowels: a, e, i, o, u.)

Question 5.2. Consider training a supervised document classifier for sentiment, and compare it to a lexicon counting classifier. If you have a very low number of labeled documents, which model do you expect to be better? If you have a very high number of labeled documents, which model do you expect to be better? Why?

6 Text preprocessing

Question 6.1. What is the difference between tokenization and word normalization? (Not vector norms or probability normalization.) Please list a few examples of word normalization.

[**Solution:** Tokenization refers to breaking text into word-sized pieces. Strictly speaking, tokenization preserves the original string of each word. Word normalization refers to processing to collapse different tokens into the same word type: for example, lowercasing, replacing numbers with a special symbols, stemming, lemmatization, etc.]

Question 6.2. Why is word normalization used?

[**Solution:** To reduce sparsity]

Question 6.3. (A) What is the difference between lemmatization and stemming? (B) Give a justification why lemmatization may be preferred to stemming. (C) Give a justification why stemming may be preferred to lemmatization.

[**Solution:** Stemming refers to any algorithm that attempts to remove affixes from a word. Lemmatization refers to a smarter stemmer that uses grammatical information to help determine the root word form. Lemmatizers are much more accurate (just look at the output of the Porter stemmer!), but they require more resources, such as a part of speech tagger, and can be slower to run.]

Question 6.4. What's a pro and a con of using n-gram features, as opposed to bag-of-words features, for a classifier?

[**Solution:** It's the usual sparsity tradeoff you see all over NLP. N-grams have more specific meaning so you get stronger information (e.g. "not good" or "social security" are much different than the unigram features they produce), but at the cost of sparsity (the new features may be rare, so there are fewer examples to training on, and they may not occur much at runtime or in test data).]

7 More questions

Question 7.1. What's a useful thing you can calculate with NB, without having to calculate $p(x)$? **[Solution:** most likely MAP class. or, prob of text given a label.]

Question 7.2. What's a useful thing you can calculate with NB, but requires you to calculate $p(x)$? **[Solution:** probability of a particular class. probability of classes.]

Question 7.3. For Naive Bayes with many classes, consider the case where you only care about the ratio between the posterior probabilities for two classes, say, class C and D. Demonstrate (and show your work) that you do not need to calculate the Bayes Rule normalizer $p(x)$ to calculate this posterior ratio,

$$\frac{p(y = C \mid x)}{p(y = D \mid x)}$$

Question 7.4. Say you have NB for a binary classification problem. You retrain the model lots of times, and each time you make the pseudocount hyperparameter higher and higher. With each model you do predictions on new data. What happens to Naive Bayes predicted document posteriors as the pseudocount goes higher? [HINT: you can just do this intuitively. It may help to focus on the $P(w|y)$ terms. A rigorous, if overkill, approach, is to use L'Hospital's rule.]

(a) They all become either 0 or 1.

(b) They all become 0.5.

(c) They all become the class prior.

(d) There is no stable trend in all situations.

[Solution: They all become the prior. the easy way to see this is, imagine a giant alpha like a million or a zillion. For any word w,

$$p(w|y) = \frac{n_{w,y} + \alpha}{n_y + V\alpha} = \frac{n_{w,y} + 1,000,000}{n_y + V1,000,000} \rightarrow \frac{\alpha}{V\alpha} = \frac{1}{V}$$

where $n_{w,y}$ is the number of tokens among doc class y that are wordtype w, and n_y is the number of tokens for doc class y. those two numbers are dominated by the giant α, which causes all words to have the same uniform probability.]

Question 7.5. In the typical case for English, how does the number of parameters compare between BOW versus a model where features are counts of character 10-grams?

(a) BOW has more parameters than character 10-grams

(b) Character 10-grams has more parameters than BOW

(c) They are the same

[**Solution:** Char 10-grams has more. While some English words are longer than length 10, the average is less, and common function words are often very short. For example, "I had a fe" includes more than 3 tokens.]

Question 7.6. What is an issue if you want to apply BOW to Chinese documents?

[**Solution:** Word segmentation—it's a nontrivial processing step in itself. This is why character n-grams are a popular feature method for languages that don't use whitespace tokenization conventions, including Chinese, Japanese, and Korean.]

Question 7.7. Consider an annotation task with 5 items and 2 annotators, for binary classification. Both annotators annotated all items. Draw up a 5×2 matrix of their annotations and fill in any values you like, as long as agreement is less than 100%. For all the following questions, show your work. (A) What is the agreement rate? (B) What is the random chance agreement rate? (Use the overall prevalence of classes among all annotations.) (C) Calculate Cohen's kappa.

Question 7.8. What is the range of possible values of Cohen's kappa?

Question 7.9. Give an example of a task where Cohen's kappa might be high, and one where it might be low. Why the difference?

8 Syntax

Question 8.1. Constituency and dependency trees focus on different aspects of a sentence's syntactic structure. What does a constituency tree focus on? What does a dependency tree focus on?

Question 8.2. Draw a lexicalized constituency tree for an example sentence. (For example, use Figure 12.11 in SLP3.) Draw the unlabeled dependency tree it corresponds to.

Question 8.3. Give a simple CFG that can parse the following POS-tagged sentence, with an analysis conforming to the standard type of grammar used in your readings (broadly similar to the Penn Treebank style). You can exclude unary expansions from POS tags to the lexicon; assume POS tags are given to the parser as input. Draw the corresponding parse tree.

(PRP I) (VB run) (ADJ fast)

Question 8.4. Amend your CFG so it can also parse this sentence. Draw the corresponding parse tree.

(PRP I) (VB run) (ADJ fast) (IN on) (NNPS Mondays)

Question 8.5. CFGs: Eisenstein INLP textbook questions 9.8, 9.9 (pg. 233, pdf pg. 241)

Question 8.6. CFG parsing: Eisenstein INLP textbook questions 10.1–10.4 (pg. 253, pdf pg. 271)

Question 8.7. PCFGs: Eisenstein INLP textbook question 10.7, and possibly 10.8 (pg. 254, pdf pg. 272).

[**Solution:** Hint for 10.7. What is the question asking? It's not so hard once you wade through the definitions.

Σ is the symbol vocabulary so the M-way cross product Σ^M is the set of all strings of length M. The problem is asking about how the PCFG can generate (or analyze...*) a string of length M. This particular PCFG is pretty simple since all terminals symbols (σ) have the same generation probability from nonterminal Y, so you don't have to worry about the actual content of the string. You have to figure out what trees are possible for a length M string, what the probability of each tree is, and what the most probable tree is, and its probability.

To get started consider $M = 1$ or $M = 2$ or something nice and small.

I don't see the start symbol specified; it's reasonable to assume it's X.

*: the most probable way to generate a particular string w is the same thing as the highest posterior probability parse for observed string w, since $[\arg\max_y p(w, y)] = [\arg\max_y p(w \mid y)]$ (because $p(w, y) = p(w \mid y)p(y)$).]