Midterm sample questions

CS 585, Brendan O'Connor and David Belanger

October 12, 2014

1 Topics on the midterm

Language concepts

- Translation issues: word order, multiword translations
- Human evaluation
- Parts of speech
- Morphology

Non-structured models

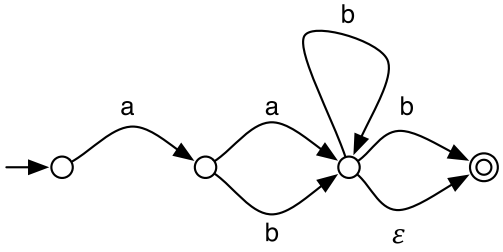
- Probability theory: Marginal probs, conditional probs, law(s) of total probability, Bayes Rule.
- Maximum likelihood estimation
- Naive Bayes
- Logistic regression (both binary and multiclass).
    - Understanding the model formula and likelihood equation is important. The gradient-based learning algorithm will not be covered.
- N-gram Markov language models
- IBM Model 1
- Relative frequency estimation and pseudocount smoothing
- The expectation-maximization algorithm
    - Besides its usage in Model 1, you need to know how to apply it to a different model if necessary.
- Classifier evaluation: precision and recall

Structured models

- Hidden Markov models, Viterbi algorithm
- Finite state automata, regular languages
- Finite state transducers

2 Finite-state things

Consider this FSA:



Question 1. For each of the following strings, is it accepted or not accepted by this FSA?

1. `ab`
2. `a`
3. `aabbb`
4. `aba`
5. `aa`

Question 2. Write a regular expression that corresponds to the same regular language represented by this FSA.

Question 3. The FSA above is bigger than it needs to be. There exists an FSA with a smaller number of edges that represents the same regular language. Please draw this FSA (or some FSA with fewer edges, in case there are others.)

Question 4. Draw an FST that removes repeated *s*'s on the end of a word. For example, it would translate *yesssss* to *yes*. Otherwise, it passes through the string unchanged. It should pass through strings without *s*'s on the end.

3 Bayes Rule and EM

You are in a noisy bar diligently studying for your midterm, and your friend is trying to get your attention, using only a two word vocabulary. She has said a sentence but you couldn't hear one of the words:

$$(w_1 = \text{hi}, w_2 = \text{yo}, w_3 = ???, w_4 = \text{yo})$$

Question 5. At first, you guess that your friend was generating words from this bigram LM:

$$\begin{array}{llll} p(\text{hi}|\text{hi}) & = 0.7 & p(\text{yo}|\text{hi}) & = 0.3 \\ p(\text{hi}|\text{yo}) & = 0.5 & p(\text{yo}|\text{yo}) & = 0.5 \end{array}$$

Given these parameters, what is the posterior probability of whether the missing word is "hi" or "yo"?

Question 6. Now you want to learn your friend's transition model, based on this utterance. Zero pseudocounts. Since there's missing data, you must use the EM algorithm. Show the results of one step of the EM algorithm, where the previous question was the first E-step. So just do an M-step and show the resulting parameters. For this question, don't bother with START and END transitions. Use zero pseudocounts.

4 Language Models

We consider a language over the three symbols 'A', 'B', and 'C'.

Question 7. Consider the training 'corpus'

$$(A, B, C, B, A, A, B, C)$$

- Under a bigram language model with zero pseudocounts, what is the probability of the observation (A, C, B)?
- Under a bigram language model with zero pseudocounts, what is the probability of the observation (A, B, C)?

Question 8. The following questions concern the basic pseudocount smoothing estimator we used in problem set 1.

1. Pseudocounts should only be added when you have lots of training data. True or False?
2. Pseudocounts should be added only to rare words. The count of common words should not be changed. True or False?
3. What happens to Naive Bayes document posteriors (for binary classification), if you keep increasing the pseudocount parameter really really high?
   (a) They all become either 0 or 1.
   (b) They all become 0.5.
   (c) Neither of the above.

Question 9. The following True/False questions concern the use of an OOV tokens.

1. The use of OOV tokens is necessary for ngram language models. It is not necessary in machine translation and document classification.

2. The OOV token should be assigned pseudocounts.

Question 10. Here is a bigram LM.

$$\begin{aligned}
p(a|\text{START}) &= 1.0 & p(b|\text{START}) &= 0 & p(\text{END}|\text{START}) &= 0\\
p(a|a) &= 0.7 & p(b|a) &= 0.3 & p(\text{END}|a) &= 0\\
p(a|b) &= 0.2 & p(b|b) &= 0.5 & p(\text{END}|b) &= 0.3
\end{aligned}$$

Write a regular expression whose corresponding regular language contains all strings with non-zero probability under the LM, and does not include any zero probability strings. Hint: it may be useful to draw a diagram depicting the weighted FSA corresponding to this Markov model.

5 Classification

We seek to classify documents as being about sports or not. Each document is associated with a pair $(\vec{x}, y)$, where $\vec{x}$ is a feature vector of word counts of the document and y is the label for whether it is about sports ($y = 1$ if yes, $y = 0$ if false). The vocabulary is size 3, so feature vectors look like $(0, 1, 5)$, $(1, 1, 1)$, etc.

5.1 Naive Bayes

Consider a naive Bayes model with the following conditional probability table:

word type	1	2	2
$P(w \mid y = 1)$	1/10	2/10	7/10
$P(w \mid y = 0)$	5/10	2/10	3/10

and the following prior probabilities over classes:

$P(y = 1)$	$P(y = 0)$
4/10	6/10

Question 11. 1. What is the probability that the document $\vec{x} = (1, 0, 1)$ is about sports?

2. What is the probability that it is not about sports?

Question 12.

1. Suppose that we know a document is about sports, i.e. $y = 1$. True or False, the Naive Bayes model is able to tell us the probability of seeing $x = (0, 1, 1)$ under the model.
2. If True, what is the probability?

Question 13. Now suppose that we have a new document that we don't know the label of. What is the probability that a word in the document is wordtype 1?

5.2 Logistic Regression

Question 14. Consider a logistic regression model with weights $\beta = (0.5, 0.25, 1)$. A given document has feature vector $x = (1, 0, 1)$. 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$\exp(-1)$.

1. What is the probability that the document is about sports?
2. What is the probability that it is not about sports?

Question 15. Consider a logistic regression model with weights $\beta = (-ln(4), ln(2), -ln(3))$. A given document has feature vector $x = (1, 1, 1)$. Now, please provide your answer in the form of a fraction $\frac{a}{b}$.

1. What is the probability that the document is about sports?

Question 16. Consider a logistic regression model with weights $\beta = (\beta_1, \beta_2, \beta_3)$. A given document has feature vector $x = (1, 0, 1)$.

1. What is a value of the vector β such that the probability of the document being about sports is 1 (or incredibly close)?
2. What is a value of the vector β such that the probability of the document being about sports is 0 (or incredibly close)?

5.3 Evaluation

You run a classifier on a test set, with the following results:

	Labeled sports	Labeled non-sports
Predicted sports	10	20
Predicted non-sports	5	2000

Question 17. What is the precision of your classifier?

Question 18. What is the recall of your classifier?

Question 19. What is the accuracy of your classifier? Is this a useful way to evaluate your classifier here? Why or why not?

6 Viterbi

Question 20. Here's a proposal to modify Viterbi to use less memory: for each token position t, instead of storing all $V_t[1]..V_t[K]$, instead store one probability, for the best path so far. Can we compute an optimal solution in this approach? Why or why not?

Question 21. Consider the Eisner ice cream HMM (from J&M 3ed ch 7, Figure 7.3), and a sequence of just one observation, $\vec{w} = (3)$. There are only 2 possible sequences, (HOT) or (COLD). Calculate both their joint probabilities ($p(w, y)$). Which sequence is more likely?

Question 22. Now consider the observation sequence $\vec{w} = (3, 1, 1)$. Perform the Viterbi algorithm on paper, stepping through it and drawing a diagram similar to Figure 7.10. What is the best latent sequence, and what is its probability? To check your work, try changing the first state; is the joint probability better or worse? (To really check your work you could enumerate all 8 possibilities and check their probabilities, but that is not fun without a computer.)

Question 23. Compare how the Viterbi analyzed this sequence, in contrast to what a greedy algorithm would have done. Is it different? Why? Why is this a different situation than the previous example of $\vec{w} = (3)$?

7 Language stuff

Question 24. Each of the following sentences has an incorrect part-of-speech tag. Identify which one and correct it. (If you think there are multiple incorrect tags, choose the one that is the most egregious.) We'll use a very simple tag system:

- NOUN – common noun or proper noun
- PRO – pronoun
- ADJ – adjective
- ADV – adverb
- VERB – verb, including auxiliary verbs
- PREP – preposition
- DET – determiner
- X – something else

1. Colorless/ADV green/ADJ clouds/PRO sleep/VERB furiously/ADV ./X
2. She/PRO saw/VERB herself/PRO through/PREP the/ADJ looking/ADJ glass/NOUN ./X
3. Wait/NOUN could/VERB you/PRO please/X ?/X

Question 25. The Penn Treebank's tokenization convention splits "auxiliary+not" contractions in English, i.e. "I can't see it" is tokenized as a sequence of length 5, [[*I ca n't see it*]]. Give an argument whether this is a good idea, from the perspective of being able to train part-of-speech taggers.

Question 26. Can or should contraction splitting be done for English-language Twitter data? Why or why not?

Question 27. IBM Model 1 has many shortcomings as a model for machine translation. Pick two of them and explain them, in one or two sentences for each.

8 EM derivation

[Note: this is much longer than questions you should expect to see on the midterm. We're including it if you want to understand EM more in-depth.]

EM is used when you have a model with observed variables x and hidden variables z; for example, in the IBM Models, x are words and z are alignment variables. In this problem we will show that the weighted counting that we do in the M-step corresponds to maximizing a *weighted* log-likelihood of the data, where the weights are the posterior values from the last E-step.

Say we're estimating the probability of the word "ctfu" (see urbandictionary.com for semantic details); call this parameter θ. Every $i = 1..N$ is a token position, and the observed variable x_i is $x_i = 1$ if that token is the word "ctfu", and $x_i = 0$ otherwise. In the simple case without hidden variables, we've already seen that from the likelihood function

$$p(\vec{x}|\theta) = \prod_i p(x_i|\theta) = \prod_i \theta^{x_i}(1-\theta)^{x_i}$$

we can derive the maximum likelihood estimator of θ by taking the derivative of the log-likelihood and setting it to zero, and finding the value of θ where the derivative is 0, yielding

$$\hat{\theta}^{(MLE)} = \frac{\sum_i x_i}{N}$$

Next: assume there are binary latent variables z_i and that we only want to estimate the probability among tokens that have $z_i = 1$; tokens with $z_i = 0$ should be ignored. (With some care you can frame Model 1 in this way; this is a simplification that's easier to analyze.) We're not going to worry about where the z_i's come from. The likelihood function is:

$$p(x_i|z_i,\theta) = (\theta)^{x_i z_i}(1-\theta)^{(1-x_i)z_i}$$

Question 28. Write $p(x_i = 1|z_i = 1, \theta)$ in terms of θ. You can get this by just plugging in the values of x_i and z_i then simplifying.

Question 29. Write $p(x_i|z_i = 1, \theta)$ in terms of x_i and θ. You can get this by just plugging in the value of z_i and simplifying.

Question 30. Why does this likelihood function ignore cases where $z_i = 0$?

We want to solve $\arg\max_\theta p(x|z, \theta)$, which is equivalent to solving $\arg\max_\theta \log p(x|z, \theta)$. So take the log-likelihood and its derivative with respect to θ, and set the derivative to zero. Since the log-likelihood function is concave, the optimal solution will be there. The first few steps of this is:

$$\begin{align}
\log p(x|z, \theta) &= \sum_i \log p(x_i|z_i, \theta) \tag{1}\\
&= \sum_i z_i \left[x_i \log \theta + (1 - x_i) \log(1 - \theta)\right] \tag{2}\\
\frac{\partial}{\partial \theta} \log p(x|z, \theta) &= \sum_i z_i \left(\frac{x_i}{\theta} + \frac{1 - x_i}{1 - \theta} \frac{\partial (1 - \theta)}{\partial \theta} \right) \tag{3}\\
&= \sum_i z_i \left(\frac{x_i}{\theta} - \frac{1 - x_i}{1 - \theta} \right) \tag{4}
\end{align}$$

The first step of the derivative above was just taking the derivative with respect to θ, and using the fact that $(\log x)' = 1/x$, and then we applied the chain rule to the thing inside the log. This is of course a partial derivative, with respect to θ (and not z_i); from the perspective of θ's derivative, z_i is a constant and the rules of calculus say it can just stay hanging out on the outside there.

If you set the above to zero, it takes just a few more lines of algebra to solve for θ, which should yield the MLE solution that counts how many tokens i have both $x_i = 1$ as well as $z_i = 1$, and normalizes out of instances where $z_i = 1$.

$$\theta^{(MLE)} = \frac{\sum_i z_i x_i}{\sum_i z_i}$$

OK, now for EM. We don't know the z_i's, but we have probabilistic guesses for them. (Don't worry about where those came from, though as you know, in Model 1 it comes from Bayes Rule using last round's parameters.) In the E-step we computed posterior values $p(z_i = 1|x, \theta^{(old)})$ at every token i; call one of these q_i (which is between 0 and 1). We've learned that in EM, we are supposed to do weighted counting instead of simple counting for the relative frequency estimator:

$$\theta^{(WMLE)} = \frac{\sum_i q_i x_i}{\sum_i q_i}$$

Question 31. Derive that this is the solution to the *weighted* log-likelihood maximization problem problem, where the q_i terms are already set and we want to learn θ,

$$\arg\max_\theta \sum_i q_i \log p(x_i|z_i = 1, \theta) + (1 - q_i) \log p(x_i|z_i = 0, \theta)$$

You can use the same argument strategy as in the simple MLE case: find the value of θ where the derivative of the log-likelihood with respect to θ is equal to zero.

Before you take the derivative, it might be easier to first plug in the likelihood values to the above equation and simplify it a bit.