Midterm sample questions

UMass CS 585, Fall 2015

October 16, 2015

1 Midterm policies

The midterm will take place during lecture next Tuesday, 1 hour and 15 minutes.

It is closed book, EXCEPT you can create a 1-page "cheat sheet" for yourself with any notes you like. One page front and back. Feel free to collaborate to create these notes. You will probably find the studying implicit in the act of creating the notes is even more useful than actually having them.

2 Topics on the midterm

Language concepts

- Parts of speech
- The Justeson-Katz noun phrase patterns

Probability / machine learning

- Probability theory: Marginal probs, conditional probs, law(s) of total probability, Bayes Rule.
- Maximum likelihood estimation
- Naive Bayes
- Relative frequency estimation and pseudocount smoothing
- Logistic regression (for binary classification)
- Perceptron
- Averaged Perceptron

Structured models

- Hidden Markov models
- Viterbi algorithm
- Log-linear models and CRFs
- Structured Perceptron

3 Bayes Rule

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 1. Assume that your friend was generating words from this first-order Markov model:

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

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

Question 2. 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? [HINT: you can try to do this intuitively. It may help to focus on the $P(w|y)$ terms. A rigorous approach is to use L'Hospital's rule.]

   (a) They all become either 0 or 1.

   (b) They all become 0.5.

   (c) Neither of the above.

4 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.

4.1 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 3.
Consider the document with counts $\vec{x} = (1, 0, 1)$.

1. Which class has highest posterior probability?
2. What is the posterior probability that the document is about sports?

Question 4. Consider the document with counts $\vec{x} = (2, 0, 1)$. Is it the case that $P(y = 1 \mid \vec{x} = (2, 0, 1)) = P(y = 1 \mid \vec{x} = (1, 0, 1))$? If not, please calculate for $(2, 0, 1)$.

Question 5. In lectures, and in the JM reading, we illustrated Naive Bayes in terms of TOKEN generation. However, $\vec{x}$ is WORD COUNTS, i.e. the BOW vector. Please rewrite the unnormalized log posterior $P(y = 1 \mid doc)$ in terms of $\vec{x}$, instead of in terms of each word token as in lecture.

Question 6.

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 7. 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?

Question 8. True or False: if the Naive Bayes assumption holds for a particular dataset (i.e., that the feature values are independent of each other given the class label) then no other model can achieve higher accuracy on that dataset than Naive Bayes. Explain.

Question 9. Can Naive Bayes be considered a log linear model? If so, explain why; if not, example why not.

Question 10. Show that for Naive Bayes with two classes, the decision rule $f(x)$ can be written in terms of $\frac{\log[P(y=1|x)]}{\log[P(y=0|x)]}$. Can the decision rule be formulated similarly for multiclass Naive Bayes?

Question 11. In terms of exploratory data analysis, why might it be interesting and important to compute the log odds of various features?

4.2 Logistic Regression

Question 12. 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 13. 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 14. 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)?

Question 15. Consider the following two weight vectors for logistic regression:

- $w = (10000, -2384092, 24249, 284924, -898)$
- $w' = (1.213, -.123, 2.23, 3.4, -2)$

For which of these weight vectors is small changes between test instances likely to make large changes in classification? Which of these models do you think generalizes better and why?

5 Language stuff

Question 16. 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

6 Perceptron

Question 17. In HW2 we saw an example of when the averaged perceptron outperforms the vanilla perceptron. There is another variant of the perceptron that often outperforms the vanilla perceptron. This variant is called the **voting perceptron**. Here's how the voting perceptron works:

- initialize the weight vector
- if the voting perceptron misclassifies an example at iteration i, update the weight vector and store it as w_i.
- if it makes a correct classification at iteration i, do not update the weight vector but store w_i anyway.
- To classify an example with the voting perceptron, we classify that example with each w_i and tally up the number of votes for each class. The class with the most votes is the prediction.

Despite often achieving high accuracy, the voting perceptron is rarely used in practice. Why not?

Question 18. [NOTE: we won't ask for any proofs by induction on the test]
Recall that the averaged perceptron algorithm is as follows:

- Initialize $t = 1, \theta_0 = \vec{0}, S_0 = \vec{0}$
- For each example i (iterating multiples times through dataset),
  - Predict $y^* = \arg\max_{y'} \theta^\mathsf{T} f(x_i, y')$
  - Let $g_t = f(x_i, y_i) - f(x_i, y^*)$
  - Update $\theta_t = \theta_{t-1} + r g_t$

- Update $S_t = S_{t-1} + (t-1)rg_t$
- $t := t + 1$

- Return $\bar{\theta}_t = \theta_t - \frac{1}{t}S_t$

Use proof by induction to show this algorithm correctly computes the average weight vector for any t, i.e.,

$$\frac{1}{t}\sum_{i=1}^{t}\theta_t = \theta_t - \frac{1}{t}S_t$$

Question 19. For the case of the averaged perceptron, why don't we make predictions during training with the averaged weight vector?

Question 20. Why wouldn't we want to use the function below to update the weight vector when training a percetron?

```
def update_weights(weight_vec, gradient):
  updated_weights = defaultdict(float)
  for feat, weight in weight_vec.iteritems():
    updated_weights[feat] += weight
  for feat, weight in gradient.iteritems():
    updated_weights[feat] += weight
  return updated_weights
```

7 HMM

Consider an HMM with 2 states, A, B and 2 possible output variables $\Delta, \square$, with transition and emission probabilities from HW2. All probabilities statements are implicitly conditioning on $s_0 = START$.

Question 21. Explain the difference between
$P(s_1 = A \mid o_2 = \Delta)$ versus $P(s_1 = A \mid o_2 = \Delta, s_3 = END)$

Question 22. Rewrite $P(s_1 \mid o_2)$ so that you could calculate it for any particular values of s_1 and o_2. (This is like in HW2, except you should be able to do it abstractly without the numbers or particular values and swap in the numbers only at the end.)

Question 23. Rewrite $P(s_1 \mid o_2, s_3 = END)$ so that you could calculate it for any particular values of s_1 and o_2.

Question 24. Why does the END state matter?

Question 25. (Here's what HW2 1.3 was supposed to be.)
Is it the case that $P(o_2 = \Delta \mid s_1 = A) = P(o_2 = \Delta \mid s_1 = A, s_3 = A)$?

Question 26. Write an expression that computes the probability of the HMM emitting the sequence $\Delta, \square$ given that the first state is A and the length of the sequence is 2 (remember to consider the start and end states).

8 Viterbi

Question 27. 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 28. Here's an erroneous version of the (multiplicative-version) Viterbi algorithm. The line in the inner loop had

- BUGGY: $V_t[k] := \max_j V_{t-1}[j] P_{trans}(k \mid j) P_{emit}(w_t \mid j)$
- CORRECT: $V_t[k] := \max_j V_{t-1}[j] P_{trans}(k \mid j) P_{emit}(w_t \mid k)$

Please describe one specific issue that the buggy version of this code would have. For example, describe an important thing in the data that the buggy version ignores.

Question 29. 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 30. 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 31. 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 $\vec{w} = (3)$?