CS674/INFO 630: Advanced Language Technologies, Fall 2007
11/29/07: Final exam guide

The final exam for the Fall 2007 semester is Thursday, Dec 13, 2:00 - 4:30 pm, Thurston 202.

You may bring to the exam up to five sheets of notes (8”×11”, both sides OK), but no other reference material. Use of calculators, laptops, etc. is not permitted.

Note that the exam is comprehensive, in the following sense. *All* material from lectures 1 (8/28/07) onwards, including lectures for which no lecture guides are available[1], is fair game. On the other hand, the exam will also be held to approximately four to six questions to allay time pressure; hence, it is easily inferred that the coverage will be very heavily weighted towards material not tested on the midterm.

It bears repeating that what is most of interest is whether one understands how the various methods and models we have discussed were developed. This is because we have introduced fundamental concepts and tools that one simply must be familiar with to understand current research in human-language technologies, and because such understanding should help enable one to develop new methods and models. As has been previously mentioned, an argument can be made that one really understands a concept when one can understand the implications when some assumption or other aspect of the setting is changed. My favorite kind of question is based on this principle.

I will be holding drop-in office hours on Thursday Nov 29th, Thursday Dec 6, and Tuesday December 11: all 3-4pm. You can, of course, also make an appointment (with advance notice).

The five questions from the Spring 2006 final exam appear on the following pages. Some notation differs from this year, but this shouldn’t pose any particular problem.

[1] Those that are available are/will be posted at the course homepage, http://www.cs.cornell.edu/courses/cs674/2007fa/ .

(1) [11 points] Figure 1 below represents hypothetical results produced by Google in response to two different queries. We have indicated whether or not the summaries were clicked on; we have also indicated what "class" the URLs of the corresponding documents belong to, where we assume some predefined set of possible classes (e.g., trustworthy vs. unknown vs. untrustworthy). For simplicity's sake, assume that the summaries shown below are actually the full text of the corresponding documents (hence the use of "*d*" instead of "*s*" to label the summaries).

		q$_a$: "cats versus dogs"			**q$_b$: "garfield versus snoopy"**
click	d_1^a:	"dogs outclass cats" [URL class 0]	no click	d_1^b:	"garfield is snoopy" [URL class -1]
no click	d_2^a:	"cats are great" [URL class 1]	click	d_2^b:	"critics prefer snoopy" [URL class 1]

Figure 1: Results for two queries.

Finally, assume the following term index:

$v^{(1)}$=dogs $v^{(2)}$=outclass $v^{(3)}$=cats $v^{(4)}$=are $v^{(5)}$ =great
$v^{(6)}$=garfield $v^{(7)}$=versus $v^{(8)}$=snoopy $v^{(9)}$=is $v^{(10)}$=critics $v^{(11)}$=prefer

a) Suppose that Figure 1 represents the training data for Joachims' 2002 system (which ignores query chains). Furthermore, assume that the query-document representation scheme is:

$$\Phi(q,d) = \begin{bmatrix} \cos(\vec{q},\vec{d}) \\ \text{URL class of } d \end{bmatrix},$$

where, for simplicity, we use tf weighting (instead of tf-idf weighting) to create $\vec{q}$ and $\vec{d}$.

Is $\vec{w} = (0,1)^T$ a valid choice for Joachims' algorithm based on the above training data? Justify your response. Your answer should include

- an intuitive explanation of the kinds of items that would be preferred if we did indeed use $\vec{w} = (0,1)^T$ for ranking (sample, probably incorrect answer: "documents containing exactly one word of the query");
- an explicit but brief explanation of what constraints, if any, are inferred from the data above, as well as what potential constraints *aren't* inferred, if any; and,
- explicit numerical computation of those $\Phi(q,d)$ that you use in checking whether the proposed weight vector is valid.

b) Suppose instead that we treat the clicks and non-clicks in Figure 1 as *explicit* relevance feedback. If we apply the Rocchio algorithm upon the results of q_b with $\alpha = 1$ and $\gamma = 0$, what is the minimum value of β, if any, that will result in ranking d_2^b above d_1^b with respect to new version of q_b (i.e., will cause the Rocchio algorithm to "do the right thing")? Be sure to justify your answer, showing all steps and providing brief but clear explanations of them.

To simplify your calculations, use *non-normalized tf weighting* to form document vectors.

(2) [6 points] **Fall 2007 note: One should not necessarily expect a midterm question to be repeated. In the Spring 2006 semester, a question from the midterm was repeated due to unusual circumstances.**

This question, which also appeared in extremely similar form on the midterm, modifies the setting that resulted in our second derivation of the LM-based approach.

Assume a finite set of document-topic language models $t_1, t_2, \ldots, t_n$, where the parameters for each t_i are known and where we assume that each document was generated by *exactly one* of models t_i. Suppose that the system is issued a query whose semantics is, "A document is relevant if it was generated by t_1 **or** by t_2". You should consider the query to be fixed and to be not a term sequence and hence not "generatable" by an LM (for instance, perhaps the system gets information requests through the user clicking on some checkboxes).

Derive (with adequate explanation of your steps) a scoring function that results from expanding $P(R = y|D = d)$ based on the information just given, where it is required that most, if not all, of the quantities in your function can be directly estimated in a reasonable way. For each quantity, be sure to explain either how you would estimate it, justifying your choice, or why a problem arises (despite good-faith effort on your part) in estimating it.

Note: topic LMs are allowed to "generate" documents that are not in the corpus.

---

(3) [11 points]

a) In the following three-matrix product, a, b, c, d, and e are variables and V is a matrix:

$$\begin{bmatrix} \sqrt{2}/2 & 0 \\ \sqrt{2}/2 & a \\ 0 & b \end{bmatrix} \begin{bmatrix} 5 & c \\ d & e \end{bmatrix} \boxed{\quad V^T \quad}$$

(the size of the box surrounding V^T is not meant to indicate anything about V's dimensionality). Suppose someone asserts the following statement:

The above represents a singular-value decomposition for some matrix.

Give the values or most specific ranges of values for the variables a through e and the dimension of V that can be inferred from this statement. Be sure to give the reason(s) for each of your inferences.

b) Suppose we have a corpus consisting of just two document vectors, $\overrightarrow{d^{(1)}} = (x, y)^T$ and $\overrightarrow{d^{(2)}} = (x, -y)^T$. Prove that for any real-valued x and y such that $x > y > 0$, the first left singular vector $\overrightarrow{u_1}$ of the corresponding term-document matrix does not lie along either $\overrightarrow{d^{(1)}}$ or $\overrightarrow{d^{(2)}}$. (Diagrams of vectors, ellipses, etc. can serve as useful explanatory aids, but they generally do not suffice as full proofs.)

Hint: Do *not* attempt to explicitly compute $\overrightarrow{u_1}$. Rather, consider the lengths of the mappings of coefficient vectors $(1, 0)^T$, $(0, 1)^T$, and one that "splits the difference"; and recall how left singular vectors relate to lengths of various entities.

(4) [6 points] Consider the following simplification of EM-based learning of PCFG rule probabilities. We have some fixed CFG, and only wish to learn probabilities for the n rewrite rules that expand the non-terminal "NP" (which, we note, is likely to appear several times in one parse tree); probabilities for all other rules are considered fixed. The free parameters of our model are thus $\theta = (\rho_1, \ldots, \rho_n)$, where ρ_j is the parameter corresponding to the probability of the jth rewrite rule for "NP".

Iterations of the EM algorithm for this setting consist of computing

$$\theta_i = \arg\max_{\theta} \sum_t P_{\theta_{i-1}}(t|w) \log \left[K(t) \prod_{j=1}^{n} \rho_j^{\#(\text{rule } j \text{ in } t)} \right]$$

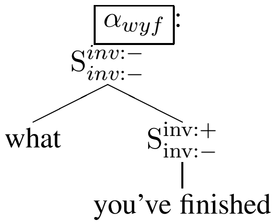
subject to certain constraints, where the term $K(t)$, which accounts for the probabilities of the non-NP-expansions in t, does not depend on θ.

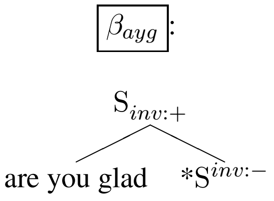
Suppose that we now erase the "log" (but not its argument!) from the above equation. Employ the same method as in class to (attempt to) solve for the value of component ρ_j of θ_i when this new optimization criterion is used. Be sure to show all work and provide (brief) adequate justifications of your steps; also, don't forget to incorporate relevant constraints on the ρ_j's.

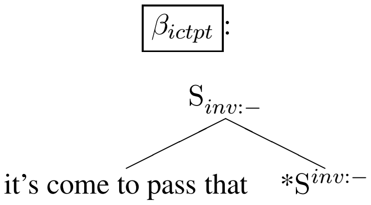
Note: there is a point at which it is not possible to advance the computations any further; answers should correctly identify this point and the obstacle(s) to proceeding in order to receive full credit. The "secret" motivation behind this question is to examine a reason why the optimization criterion used in the EM algorithm employs log-likelihood rather than "plain" likelihood. (The "non-secret" motivation is to test the ability to apply EM to different models.)

---

(5) [6 points] Consider the following proposal for a feature-based TAG:







Your task:

a) show that this FBTAG correctly generates "what are you glad you've finished";

b) explain why it correctly *doesn't* generate "what you've finished" as a complete sentence; and

c) demonstrate that it (unfortunately) allows some other ungrammatical sentence to be generated.

As always, provide (brief) adequate explanations of your reasoning.

Note: the "secret" motivation behind *this* problem is to provide partial evidence as to why one might want to specify that adjunction is *not* allowed at certain nodes.