University of Illinois at Urbana-Champaign

# Midterm Examination

CS410 Text Information Systems
Professor ChengXiang Zhai
TA: Maryam Karimzadehgan

Time: 2:00–3:15pm, Mar. 14, 2008
Place: Room 1105, Siebel Center

Name:__________________________

NetID:________________

1. **[10 points] Evaluation**

   (a) **[5/10 points]** Suppose we have a topic (i.e., a query) with a total of 10 relevant documents in the whole collection. A system has retrieved 8 documents whose relevance status is

   $$[+,+,+,-,+,-,-,-]$$

   in the order of ranking. A "+" (or "-") indicates that the corresponding document is relevant (or non-relevant). For example, the first three documents are relevant, while the fourth is non-relevant, etc. Compute the precision, recall, and the (non-interpolated) mean average precision for this result.

   **Solution:**
   Recall: 0.4
   Precision: 0.5
   MAP: 0.38

   (b) **[5/10 points]** In Normalized Discounted Cumulative Gain (NDCG), we normalize the Dicounted Cumulative Gain (DCG) for each topic with a normalizer. What is this normalizer? Why do we need to do this normalization step? We do we *not* need to do this normalization step for the Mean Average Precision (MAP)?
   **Solution**:

   - The idea for normalization is to compute the maximum DCG at rank $n$ one can possibly achieve. This value can be obtained by assuming the ideal ranking of results for the topic, i.e., putting the documents with the highest ratings on the top. $MaxDCG = R1 + \sum_{i=2}^{n} \frac{R_i}{\log_2^i}$.

   - The reason for having such a normalizer is that: When averaging over these DCG values, it might be a case that a topic with more relevant documents would dominate the average value. So, we normalize it and the value can then be used to compare two different ranking methods.

   - For MAP, the normalizer is 1.0, so the MAP value is already normalized.

2. **[20 points] Retrieval Heuristics**

(a) What are the three major term weighting heuristics in the vector-space retrieval model?

**Solution:**
TF
IDF
Document length normalization

(b) In the vector space retrieval model, if a query has just one term, which term weighting heuristic will be ineffective? Use no more than two sentences to explain why.

**Solution:**
IDF, all documents would have the same IDF.

(c) What are the two different roles of smoothing in the query likelihood retrieval method?

**solution:**
i) Explain unseen words
ii) Explain noise in query

(d) What are the two scores of a Web page that the HITS algorithm computes?

**solution:**
i) Hub
ii) Authority

(e) What would be the PageRank results if there is a directional link between every two documents?

**solution:**
It would be the same for all documents.

3. **[25 points]** Consider the following statistics collected in an imaginary search engine which only receives queries from two domains – ".EDU" and ".COM". Suppose our vocabulary has only three words $V = \{the, algorithm, computer\}$ (i.e., every query is a subset of $V$). The following table shows the counts of query words in a very small sample of queries from the two domains.

| **Domain** | number of queries | number of queries with a particular word | | |
|---|---|---|---|---|
| | | "the" | "algorithm" | "computer" |
| EDU | 10 | 10 | 10 | 4 |
| COM | 10 | 10 | 2 | 8 |

That is, we observed altogether 10 queries in each domain. Among the 10 queries from ".EDU", "the", "algorithm", and "computer" occurred in 10, 10, and 4 of them, respectively. And among the 10 queries from ".COM", "the", "algorithm", and "computer" occurred in 10, 2, and 8 of them, respectively.

We introduce the following probabilisties to model the data:

- $p(D)$ where $D \in \{EDU, COM\}$: probability that a query is from a particular domain.
- $p(X_t)$ where $X_t \in \{0, 1\}$: probability whether the word "the" is present ($X_t = 1$) or absent ($X_t = 0$) in any query.
- $p(X_a)$ where $X_a \in \{0, 1\}$: probability whether the word "algorithm" is present ($X_a = 1$) or absent ($X_a = 0$) in any query.
- $p(X_c)$ where $X_c \in \{0, 1\}$: probability whether the word "computer" is present ($X_c = 1$) or absent ($X_c = 0$) in any query.

(a) (6 points) Use the maximum likelihood estimator to compute the following probabilities:
**Solution:**

- $p(D = EDU) = 0.5$
- $p(X_t = 1) = 1$
- $p(X_a = 1) = 0.6$
- $p(X_a = 0) = 0.4$
- $p(X_a = 1|D = EDU) = 1$
- $p(X_a = 0|D = EDU) = 0$

(b) (5 points) Consider the entropies of the three binary random variables $X_t$, $X_a$, and $X_c$ (i.e., $H(X_t), H(X_a), H(X_c)$). Without calculating the entropies, can you tell which two of them have the same entropy? Why?

**Solution:**
$H(X_a), H(X_c)$ since they have the same probability.

(c) (6 points) Now consider the three conditional distributions $p(X_t|D = EDU)$, $p(X_a|D = EDU)$, and $p(X_c|D = EDU)$. We may also compute their entropies (i.e., $H(X_t|D = EDU), H(X_a|D = EDU), H(X_c|D = EDU)$). This time, which two of them have the same entropy? Why? (Once again, you do not really need to actually compute the entropy values.)

**Solution:**
$H(X_a|D = EDU), H(X_t|D = EDU)$ since they have the same probability.

(d) (8 points) Given a new query $Q =$ "the computer algorithm" and assuming that the observation of each query word is independent of each other in each domain, can you use Bayes Rule to infer whether $Q$ is from the ".EDU" domain or ".COM" domain? Show your calculation. (Hint: compute $\frac{p(D=EDU|X_a=1,X_t=1,X_c=1)}{p(D=EDU|X_a=1,X_t=1,X_c=1)}$. Use Bayes rule to convert $p(D|X_a = 1, X_t = 1, X_c = 1)$ to $p(X_a = 1, X_t = 1, X_c = 1|D)$ and then decompose it into $p(X_a = 1|D)p(X_t = 1|D)p(X_c = 1|D)$)

**Solution:**

$\frac{p(D=EDU|X_a=1,X_t=1,X_c=1)}{p(D=EDU|X_a=1,X_t=1,X_c=1)} =$

$\frac{p(X_a=1,X_t=1,X_c=1|D=EDU)p(D=EDU)}{p(X_a=1,X_t=1,X_c=1|D=COM)p(D=COM)} =$

$\frac{p(X_a=1|EDU)p(X_t=1|EDU)p(X_c=1|EDU)p(D=EDU)}{p(X_a=1|COM)p(X_t=1|COM)p(X_c=1|COM)p(D=COM)} =$

$\frac{1*1*0.4*0.5}{0.2*1*0.8*0.5} = 2.5$

So, query is from EDU domain.

4. **[25 points] Dirichlet prior smoothing**

Suppose we have an extremely small vocabulary with only 8 words $w_1$, ..., $w_8$.

**Solution:**

| Word | Ref. Model $p(w\|REF)$ | $c(w,d)$ | $p_{ml}(w\|d)$ | $p_\mu(w\|d)$ |
|---|---|---|---|---|
| $w_1$ | 0.3 | 2 | 0.2 | |
| $w_2$ | 0.15 | 1 | 0.1 | |
| $w_3$ | 0.1 | 2 | 0.2 | |
| $w_4$ | 0.1 | 3 | 0.3 | 0.2 |
| $w_5$ | 0.05 | 2 | 0.2 | |
| $w_6$ | 0.1 | 0 | 0 | 0.05 |
| $w_7$ | 0.1 | 0 | 0 | |
| $w_8$ | 0.1 | 0 | 0 | |

The second column shows the reference/collection language model $p(w|REF)$ estimated using the entire collection. The third column $c(w,d)$ shows the counts of words in document $d$.

(a) (5 points) Fill in the estimated probabilities using the non-smoothed maximum likelihood estimate for all the eight words (column 4).

(b) (5 points) Assume that we use Dirichlet prior smoothing and set the smoothing parameter $\mu = 10$. Fill in the probabilities of words $w_4$ and $w_6$ after applying the smoothing. Show your calculation in the space below.

**Solution:**
$p(w_4|d) = \frac{3}{(10+10)} + \frac{(10*0.1)}{(10+10)} = 0.2$

$p(w_6|d) = 0 + \frac{(10*0.1)}{(10+10)} = 0.05$

(c) (7 points) Let $q = w_4 w_6$ be a query. Assume that we use Dirichlet prior smoothing method with $\mu = 10$ to smooth the document language model for document $d$. What is the probability of $q$ according to this smoothed language model?

**Solution:**

$p(q|d) = p(w_4|d)p(w_6|d) = 0.01$

(d) (8 points) Can you give two two-word queries $q_1$ and $q_2$ such that if we increase the value of the smoothing parameter $\mu$, $p(q_1|d)$ would monotonically increase while $p(q_2|d)$ would monotonically decrease?

**Solution:**

$q_1 = w_1 w_2$
$q_2 = w_3 w_4$

OR

$q_1 = w_6 w_7$
$q_2 = w_4 w_5$

5. **[10 points] Efficiency**

(a) **(5 points)** Write down the gamma code for the integer 7.

**Solution:**

11011

(b) **(5 points)** Would removing a few words with *low* IDF values from an inverted index help reduce the size of the index substantially? Why?

**Solution:**

Yes, low IDF means that a word occurs in many documents, so the posting of such words in the index is very large. So removing those words will reduce the size of the index.

5. **[10 points] Collaborative Filtering** In collaborative filtering, given an active (target) user's known preferences on some items, we exploit the preference pattens of a group of users to help predict what *additional* items that the active user may also like. The same idea can also be exploited to expand a query for retrieval in a way similar to pseudo feedback by treating the query as an "active user". Explain how the memory-based collaborative filtering method can be used to predict what *additional* terms can be used to expand an existing query. Specifically, what are the other "users"? What are the "items"? What is the "rating" in this case? Briefly sketch the algorithm, preferrably with some formulas. Assume that $r(Q, D)$ is a retrieval function that can give you a positive similarity value for any query and document.

**Solution:**

There are some possible solutions, here are two of them:

**Solution1:**
Queries are other users, documents are items and rating is the similarity measure between a query and a document.

Algorithm:
Find all documents for an active query based on a retrieval function, r(Q,D)
Find other queries that also retrieved these documents which are called similar users in memory-based approach
Find other documents which are retrieved by these similar queries, these documents might be of interest to the active query.
Return these sets of documents

$r(Q_i, D_j)$ is the rating of document $j$ by user $i$ (similarity measure)
$n_i$ the average rating of all documents by user i
Normalized ratings: $S_ij = r(Q_i, D_j) - n_i$
$S' = k \sum_{i=1}^{m} W(q,i) S_ij$
$k = \frac{1}{\sum_{i=1}^{m} W(q,i)}$

**Solution 2:**
Treating top-ranked documents as "other users", "terms" as "items". Thus the current user would be the query which has terms, and we will recommend the terms in the top ranked documents (items rated by other users) to the query (our current user).

6. **[Extra credit: 10 points] Adaptive Filtering**

In adaptive filtering, for each document in a stream, the system makes a binary decision regarding whether the document is interesting to a user. The results are thus a sequence of "yes/no" values. From the user's perspective the results are whatever documents the system delivers to the user. The filtering performance is often evaluated using a linear utility function $LU = \alpha R^+ - \beta N^+$ that rewards a correctly delivered document and penalizes an incorrectly delivered one.

Now suppose we can estimate the probability that a document is relevant accurately, i.e., we know $p(Rel|d)$ for each document. We can then use a probability threshold $\theta$ to make the filtering decision such that we will deliver $d$ iff $p(Rel|d) > \theta$. How should we set $\theta$ to optimize our utility function $LU$? Can you derive a formula for computing $\theta$ based on $\alpha$ and $\beta$?

**Solution:**

Given a document d, the expected utility of delivering d is $p(Rel|d) * \alpha - (1 - p(Rel|d)) * \beta$

The expected utility of rejecting d is 0. So, the optimal decision rule is to deliver $d$ iff

$p(Rel|d) * \alpha - (1 - p(Rel|d)) * \beta > 0$

$p(Rel|d) > \frac{\beta}{(\alpha+\beta)}$

scratch

scratch

scratch