Auburn University
Department of Computer Science and Software Engineering

COMP 5970/ 6970/ 6976– Midterm Exam #2
Information Retrieval: Spring 2020

Name: Mousumi Akter

NetID: 904095347

This exam contains 13 pages (including this cover page) and 8 questions. Total of points is 100.
Good luck!

Distribution of Marks

Question	Topic	Points	Score
1	Evaluation	10	
2	Language Models	10	
3	EM Algorithm	20	
4	Mixture Model	20	
5	Feedback	10	
6	Distributed Semantics	20	
7	Learning to Rank	5	
8	Recommender Systems	5	
Total		100	

1. **[10 points] Evaluation**

Questions (a)-(f) refer to the following scenario of two systems:
Suppose a query has a total of 4 relevant documents in a collection with 100 documents. System A and System B have each retrieved 10 documents, and the relevance status of the two ranked lists of results is:

System A: [+,+,-,-,-,-,-,-,-,-]
System B: [+,-,+,-,-,-,-,-,-,-]

where a "+" (or "-") indicates that the corresponding document is relevant (or non-relevant). For example, the first document from System B is relevant, while the second is non-relevant, etc.

(a) **[1/10 point]** What is the precision of System A? (A) 2/4 (B) 2/8 [✓ (C) 2/10] (D)4/10

(b) **[1/10 point]** What is the recall of System A? [✓ (A) 2/4] (B) 2/8 (C) 2/10 (D) 4/10

(c) **[3/10 points]** What is the average precision of System A? (A) 2/4 (B) 2/8 [✓ (C) 2/10] (D) 4/10

(d) **[1/10 points]** If we compare System A and System B based on precision at top five documents (i.e., prec@5doc), which system is better?
(A) A is better than B (B) B is better than A [✓ (C) A and B have the same prec@5doc]

(e) **[1/10 points]** If we compare System A and System B based on average precision, which system is better?
[✓ (A) A is better than B] (B) B is better than A (C) A and B have the same average precision.

(f) **[3/10 points]** Which of the following general statements about Mean Average Precision (MAP) and NDCG are true?

(A) When binary relevance judgments are made, both MAP and NDCG can be used to measure ranking accuracy. (True. NDCG also can perform for multilevel judgements but MAP cant)
(B) Because NDCG is normalized per query, it can better ensure comparability between different queries than MAP. (True)
(C) When more than two levels of relevance judgments are made, we cannot use MAP for evaluation. (True)
(D) Both NDCG and MAP are more sensitive to small differences in ranking than precision at k documents. (True but NDCG ensures more comparability as it has been normalized)

2. **[10 points] Language Models for Retrieval**

Let $q = q_1 \ldots q_m$ be a query, d be a document, $p(q_i\ d)$ be the probability of query word q_i according to a smoothed document language model estimated based on d, and $p(w\ C)$ be the collection (background) language model.

(a) **[3/10 points]**

Ans: The ranking remains same even if we take the logarithm of the scoring function.

(b) **[7/10 points]**

Ans: The query vector, $q = w_1, w_2, \ldots, w_n$ such that n = |V|.
The query vector is a vector of all words in the vocabulary which contains number of times a particular word present in a query.
The document vector d= $w_1, w_2, \ldots w_n$ such that n=|V|.
The document vector is a vector of all words in the vocabulary which contains number of times a particular word present in a document.
The similarity function is a sum over all the matched query terms along with the extra term in right side of equation.
For JM smoothing,

$$p(w \mid d) = (1-\lambda)\frac{c(w,d)}{|d|} + \lambda p(w \mid C)$$

For JM Smoothing, the scoring function is,

$$score(Q,D) = \sum_{w \in Q \cap D} c(w,Q) * log(1 + \frac{(1-\lambda)*c(w,D)}{\lambda * P(w|REF)*|D|})$$

We see very clearly the TF weighting in the numerator, which is scaled sublinearly. We also see the IDF-like weighting, which is the p(w | REF) term in the denominator; the more frequent the term is in the entire collection, the more discounted the numerator will be. Finally, we can see the |D| in the denominator is a form of document length normalization, since as |D| grows, the overall term weight would decrease, suggesting that the impact in this case is clearly to penalize a long document.

3. **[20 points] EM Algorithm**

   (a) **[4/20 points]** Write down the formula to compute the probability of generating a word w_i in document D_2?

   $$\lambda p(w_i \mid C) + (1-\lambda) p(w_i \mid \theta)$$

   (b) **[2/20 points]** Write down the log-likelihood of the whole document D_2, i.e., the probability of observing all the words in D_2 being generated from the mixture model.

   $$\log p(D \mid \theta) = \sum_{d \in D} \sum_{i=1}^{|d|} \log \{\lambda p(w_i \mid C) + (1-\lambda) p(w_i \mid \theta)\}$$

   (c) **[6/20 points]** How many binary hidden variables in total do we need for computing this maximum likelihood estimate?

   Ans: A binary hidden variable z has been introduced to indicate whether a word has been generated from the background model (z = 1) or the topic model (z = 0).

(d) **[8/20 points]** Write down the E-step and M-step formulas for estimating λ.

E-step: Since we established earlier that maximizing $F(q, \theta_F^{(n)})$ with $\theta_F^{(n)}$ fixed means minimizing the KL-divergence between $q(Z)$ and $p(Z \mid D, \theta_F^{(n)})$, the main computation here is to compute

$$p(z_{i,j} = 0 \mid D, \theta_F^{(n)}) = \frac{\lambda p(w_{i,j} \mid C)}{\lambda p(w_{i,j} \mid C) + (1-\lambda) p(w_{i,j} \mid \theta_F^{(n)})}$$

which can be rationalized using Bayes' rule. We can then simply let

$$q(z_{i,j} = 0) = p(z_{i,j} = 0 \mid D, \theta_F^{(n)})$$

and

$$q(z_{i,j} = 1) = 1 - q(z_{i,j} = 0)$$

to complete the setting of q to maximize $L(q, \theta_F^{(n)})$.

M-step: Now we need to maximize $L(q, \theta_F^{(n+1)})$, holding q fixed to the value we computed in the E-step. We note that the entropy term $H(q)$ is a constant and thus we can ignore it. We can then set about maximizing the first term in the expression. We can do this analytically by introducing Lagrange multipliers and taking derivatives with respect to each parameter $p(w \mid \theta_F)$.

Let's let $n_{w,F}$ be the number of times we expect to see word w drawn from the feedback distribution given our data D and our latent variable distribution q. We can see that

$$n_{w,F} = \sum_{i=1}^{N} \sum_{j=1, d_{i,j}=w}^{|d_i|} q(z_{i,j} = 1) = \sum_{d \in D} q(z_w = 1) c(w, d).$$

where we've relabeled z to be indexed by w by noting that $z_{i,j}$ depends only on the specific word type $d_{i,j} = w$.

Since we know how to estimate a multinomial distribution given count data, we can use these numbers directly to re-estimate our parameters for $p(w \mid \theta_F^{(n+1)})$. Specifically, we have

$$p(w \mid \theta_F^{(n+1)}) = \frac{n_{w,F}}{\sum_{w' \in V} n_{w',F}} = \frac{\sum_{i=1}^{N} q(z_w = 1) c(w, d_i)}{\sum_{i=1}^{N} \sum_{w' \in V} q(z_{w'} = 1) c(w', d_i)}$$

4. **[20 points] Mixture Models**

   (a) **[4/20 points]** Write down the likelihood for the described two-component mixture model.

   (i) $p(d|\theta_0, \theta_1, \lambda)$ =

$$: \prod_{i=1}^{N} \prod_{j=1}^{|d_i|} \left(\lambda p(d_{i,j} = w \mid D) + (1-\lambda) \sum_{k=1}^{K} p(z_{i,j} = k \mid \pi_i) p(d_{i,j} = w \mid \theta_k) \right)$$

   (ii) This expression is: **(A)** The complete likelihood **(B)** The incomplete likelihood

   (b) **[4/20 points]**

   (i) $p(d, z|\theta_0, \theta_1, \lambda)$ =

$$\frac{(1-\lambda) \sum_{k=1}^{K} p(z_{i,j} = k \mid \pi_i^{(n)}) p(w \mid \theta_k)}{\lambda p(w \mid D) + (1-\lambda) \sum_{k=1}^{K} p(z_{i,j} = k \mid \pi_i^{(n)}) p(w \mid \theta_k)}$$

   (ii) This expression is: **(A)** The complete likelihood **(B)** The incomplete likelihood

(c) **[4/20 points]** Give the following formulas in terms of the parameters λ, $p(w|\theta_0)$ and $p(w|\theta_1)$

$$p(z=k \mid \pi_d^{(n+1)}) = \frac{n_{d,k}}{\sum_{k'=1}^{K} n_{d,k'}} = \frac{\sum_{w\in d} c(w,d) q_y(y_{d,w}=1) q_{z|y}(z_{d,w}=k)}{\sum_{k'=1}^{K}\sum_{w\in d} c(w,d) q_y(y_{d,w}=1) q_{z|y}(z_{d,w}=k')}$$

for k = 0, we will find $p(z_{w_i}=0|w_i)$ and k= 1, we will find $p(z_{w_i}=1|w_i)$

(d) **[6/20 points]** Write down the maximization step of the EM algorithm for θ_0 andθ_1

$$p(w \mid \theta_k^{(n+1)}) = \frac{n_{w,k}}{\sum_{w'\in V} n_{w',k}} = \frac{\sum_{d\in D} c(w,d) q_y(y_{d,w}=1) q_{z|y}(z_{d,w}=k)}{\sum_{w'\in V}\sum_{d\in D} c(w',d) q_y(y_{d,w'}=1) q_{z|y}(z_{d,w'}=k)}$$

For k=0, we will find $p(w|\theta_0)$ and k=1, we will find $p(w|\theta_1)$.

(e) **[1/20 points]** The expectation maximization (EM) algorithm cannot be guaranteed to converge to a global optimum unless there is only one maximum value (only one local maximum).

[✓ **true**] **false**

(f) **[1/20 points]** Briefly describe one way of determining when to stop running iterations of the EM algorithm.

Stop when likelihood doesn't change

5. **[10 points] Feedback in Retrieval**

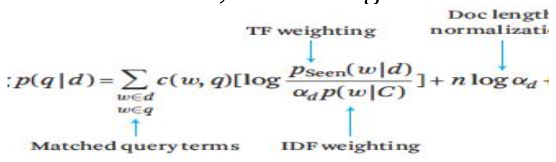
(a) **[5/10 points]** What is a major problem with this way of estimating θ_Q from retrieval perspective? Can you propose a way to solve this problem?

Ans: The major problem is if c(w, Q) = 0 there will be math error in the computation of AKL. We can use smoothing here. Like,

$$P(w|\theta_Q) = \frac{c(w,Q)+1}{|Q|+|V|}$$

(b) **[5/10 points]** After you fix this problem, analyze how well various retrieval heuristics (e.g., TF weighting, IDF weighting, and document length normalization) are implemented in this retrieval function and analyze whether this AKL retrieval function would work as well as the regular KL-divergence retrieval function.

Ans: It captures TF-weighting (for the presence of c(w,D) & c(w,Q)) and doc length normalization for the presence of |D| but it doesn't capture IDF weighting. But the regular KL-divergence retrieval function captures TF weighting, IDF weighting, and document length normalization. So, I think regular KL-divergence retrieval function will perform better.


TF weighting
Doc length normalizati
$p(q|d) = \sum_{w\in d, w\in q} c(w,q)[\log \frac{p_{Seen}(w|d)}{\alpha_d p(w|C)}] + n \log \alpha_d$
Matched query terms
IDF weighting

6. **[20 points] Distributed Semantics**

   (a) **[5/20 points]** What is Distributional Hypothesis? How does CBOW and Skipgram cap- ture this Hypothesis?

   Ans: The distributional hypothesis is:

   Words that are high similar occur in the same contexts as one another.

   The distribution hypothesis incorporated into CBOW because a context ought to be able to predict its missing word. For an example, if a sentence is said "Professor gives --- in the University", the missing word "Lecture" should be predicted.

   The distribution hypothesis incorporated into Skip-Gram because a word ought to be able to predict its context. For an example, if word "University" is used it should predict word like "Professor, Lecture, Class, Exam, Assignments, Research etc."

   (b) **[5/20 points]** Let $X(w)$ be the embedding vector for word w learned using a word embedding algorithm such as Glove or Word2Vec. Consider three vectors X("Apple"), X("Steve Jobs"), X("Microsoft"), which represent the three word vectors representing words "Ap- ple", "Steve Jobs" and "Microsoft", respectively. How would you use these three vectors to answer the following analogical question?

   "Who is the founder of Microsoft?"

   Ans: Among these three vectors, we will measure which word is found most frequently or found in same context with word "Microsoft". That word will be used to answer the analogical question.

(c) **[5/20 points]** This questions refers to the paper by Tomas Mikolov et al. "*Distributed Representations of Words and Phrases and their Compositionality*". In: Advances in Neural Information Processing Systems 26.2013, pp. 3111–3119.

As a result of the Negative Sampling approach described in the paper for training word embedding, is it possible for a pair of word (a, b), where a is the target word and b is the context word, that, pair (a, b) is considered as both a positive and negative example during training? If possible, how? If not possible, why?

Ans: If pair (a,b) both occurs in the data, they both will be considered as positive example.
On the other hand, if both words are from same unigram distribution U(w) where U(w) has been raised to the 3/4rd power. They both may be considered as negative example. So, its possible.

(d) **[5/20 points]** Both Skipgram and Glove paper replace the original (hard) learning problem into a much simpler and computationally feasible alternative machine learning problem. Which of the following alternative learning problem was formulated by Skipgram? What was done for the same in case of Glove? Justify your answer.

Clustering	Learning to Rank
Regression	Binary Classification
Multi-Class Classification	Forecasting a time series

Ans: In word2vec, it converts the problem to a binary classification problem. It gives output 1 for the words that are neighbor and 0 for the words that are not neighbor.
In glove, probability ratios have been taken and it has been assumed probability ratios are more informative. For the neighboring words, probability ratios will be higher. For glove, it is a multi-class classification.

7. **[5 points] Learning to Rank**

Learning-to-rank (LETOR) methods can be categorized into the follow three types:

- Pointwise LETOR method
- Pairwise LETOR method
- Listwise LETOR method

Among these three methods, which one do you think is more appropriate for practical scenario? Why?

Ans:

Point-wise approach regress the relevance score of a single object.

Pair-wise approach predict partial ranking of two documents.

List-wise approach is based on a ranked list of items.

I think pair-wise approach is more appropriate for practical scenario. Because, it gives us the partial ranking of documents and tells us which one is better. On the other hand, pointwise approach gives only the score. But in reality, we dont' care about the score. Listwise approach works only for ranked list of items which is a generalized version of pairwise LETOR method.

8. **[5 points] Recommender Systems**

Collaborative filtering is essentially making filtering decisions for an individual user based on the judgments of other users. It heavily relies on the availability of large number of historical user preferences. However, when a Collaborative Filtering based Recommender Systems is deployed for the first time, such historical data is not immediately available. This is called a "cold start" problem.

Describe one possible solution to the "cold start" problem.

Ans: We assume an unknown function f (., .), that maps a user and object to a rating. In the matrix X, we have observed there are some output values of this function and we want to infer the value of this function for other pairs that don't have values. This is very similar to other machine learning problems, where we would know the values of the function on some training data and we hope to predict the values of this function on some unseen test data. We can use linear regression (if the predicted value is numeric) or logistic regression (if the predicted value is a class). Also, there are many other techniques to solve this problem.

This page is intentionally left blank to accommodate work that wouldn’t fit elsewhere and/or scratch work.

This page is intentionally left blank to accommodate work that wouldn’t fit elsewhere and/or scratch work.