FINAL KEY CSC 483/583 2021

1. **(30 pts) Probability theory**

According to a survey performed at Caffe Luce, 70% of the people entering the coffee shop bought coffee. Of the people who bought coffee, 40% also bought a muffin. On the other hand, the probability that a person did *not* buy coffee *and* bought a muffin is 10%. Let C represent the probability that a person bought coffee, and M indicate the probability that a person bought a muffin. Let C' and M' be the corresponding complements (i.e., did *not* buy coffee; did *not* buy a muffin).

(a) (5 pts) Circle the correct definition for the probability of the people who bought coffee (70%):

i. $P(M|C)$

ii. $P(C)$

iii. $P(C \cap M)$

(b) (5 pts) Circle the correct definition for this probability: "Of the people who bought coffee, 40% also bought a muffin:"

i. $P(M|C)$

ii. $P(C|M)$

iii. $P(C \cap M)$

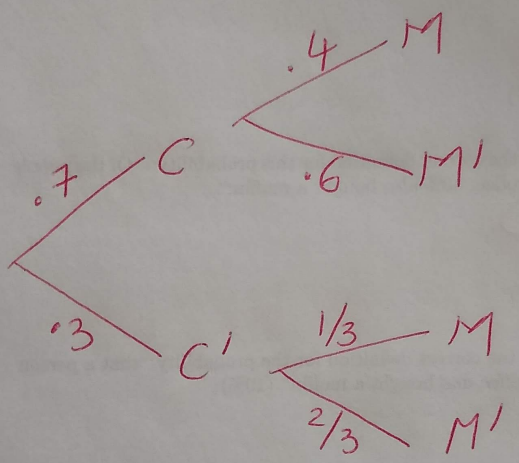
(c) (5 pts) Circle the correct definition for the probability "that a person did *not* buy coffee *and* bought a muffin" (10%):

i. $P(M|C')$

ii. $P(C'|M)$

iii. $P(C' \cap M)$

(d) (5 pts) Draw the probability tree and fill in all the probability values for all branches. If you have to calculate some of them, show the expressions used for each.



$P(C' \cap M) = P(C') \cdot P(M/C')$

$.1 = .3 \cdot P(M/C')$

$P(M/C') = 1/3$

Key: −1 for each incorrect branch
−2 for not showing work

(e) (5 pts) Suppose we know that a person bought a muffin. What is the probability that she bought coffee? Fill in the event in parentheses, and compute the corresponding probability. Show all work.

$$P(C|M) = \frac{P(M|C)\cdot P(C)}{P(M)} = \frac{0.4 \cdot 0.7}{0.3 \times \frac{1}{3} + 0.4 \times 0.7} =$$

$$= \frac{0.28}{0.1 + 0.28} = \frac{0.28}{0.38} = 0.74$$

(f) (5 pts) Are the C and M events independent? Show all work to justify your decision.

$$\left.\begin{array}{l} P(M|C) = .4 \\ P(M) = 0.38 \end{array}\right\} \Rightarrow P(M|C) \neq P(M)$$

$$\Downarrow$$

NOT independent!

2. **(15 pts) Probabilistic IR**

Name three differences between the BM25 model and the vector space *tf.idf* model.

	Vector space	BM25
framework	geometry	probability theory
length normalization?	implicit through cosine normalization	explicit, relative to the length of the average document
hyper parameters	No	Yes

-5 for missing / incorrect difference
- default in Lucene does not count!
- "better results" does not count!

3. **(40 pts undergrads, 60 pts grads) Language models**

Consider the following document collection with two documents:

Doc1: the best movie should be a meaningful movie
Doc2: summer of soul is the best movie of the year

(a) (40 pts) Under a unigram language model, what is $P(\text{movie}|\text{Doc1})$? Use Jelinek-Mercer smoothing with $\lambda = 0.5$. Do not remove stop words.

$$P(\text{movie} \mid \text{Doc1}) = \frac{1}{2} P(\text{movie} \mid \text{Doc1}) + \frac{1}{2} P(\text{Movie} \mid \text{Coll.})$$

$$= \frac{1}{2} \cdot \frac{2}{8} + \frac{1}{2} \cdot \frac{3}{18} = \frac{1}{8} + \frac{1}{12} = \frac{5}{24} \approx 0.21$$

(b) (20 pts - GRAD STUDENTS ONLY) Under a bigram language model, what is $P(\text{movie}|\text{best}, \text{Doc1})$ (interpreted as "movie" immediately follows "best" in the text)? Do not remove stop words. Use Jelinek-Mercer smoothing adapted to work for bigram models; in this problem, back off first to bigram probabilities in the whole collection, and then to unigram probabilities in the collection. Use $\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{3}$.

$\frac{1}{3} P(\text{movie}|\text{best}, \text{Doc1}) + \frac{1}{3} P(\text{movie}|\text{best}, \text{Coll.}) +$

$+ \frac{1}{3} P(\text{movie}|\text{Coll.}) =$ [illegible]

$=$ [illegible]

$= \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot \frac{2}{2} + \frac{1}{3} \cdot \frac{3}{18} = \frac{1}{3} + \frac{1}{3} + \frac{1}{18} = \frac{2}{3} + \frac{1}{18} = \frac{13}{18} = .72$

Key: note that the first back-off term is different than the class exercise!
-5 if not addressed

-15 if no smoothing

4. **(40 pts) Naive Bayes**

Based on the data in the table below: (i) (20 pts) how is the test document classified under a multinomial NB classifier?; and (ii) (20 pts) how is the test document classified under a multinomial NB classifier that uses only bigrams? For example, the "tokens" to use for classification are the bigrams "werewolf_vampire" and "vampire_bat" for document 2. You do not have to estimate parameters that you don't need for classifying the test document, but show all relevant work! For all classifiers use add-one smoothing.

	DocID	Words in document	in $c = Transylvania$
training set	1	dracula vampire vampire	yes
	2	werewolf vampire bat	yes
	3	budapest budapest	no
	4	hungary goulash vampire	no
test set	5	vampire vampire budapest	?

$P(T) = \frac{1}{2}$ $P(T') = \frac{1}{2}$

Vocabulary size = 7

$P(\text{vampire}|T) = \frac{3+1}{6+7} = \frac{4}{13}$

$P(\text{budapest}|T) = \frac{0+1}{6+7} = \frac{1}{13}$

$P(\text{vampire}|T') = \frac{1+1}{5+7} = \frac{2}{12} = \frac{1}{6}$

$P(\text{budapest}|T') = \frac{2+1}{5+7} = \frac{3}{12} = \frac{1}{4}$

$P(T|doc) = \frac{1}{2} \cdot \frac{4}{13} \cdot \frac{4}{13} \cdot \frac{1}{13} = 0.0036$

$P(T'|doc) = \frac{1}{2} \cdot \frac{1}{6} \cdot \frac{1}{6} \cdot \frac{1}{4} = 0.0034$

$\Rightarrow$ Transylvania

incorrect smoothing: −5

show work: −5

incorrect priors: −5

$$P(T \mid \text{vampire\_vampire}) = \frac{1+1}{4+7} = \frac{2}{11}$$

$$P(T \mid \text{vampire\_budapest}) = \frac{0+1}{4+7} = \frac{1}{11}$$

$$P(T' \mid \text{vampire\_vampire}) = \frac{0+1}{3+7} = \frac{1}{10}$$

$$P(T' \mid \text{vampire\_budapest}) = \frac{0+1}{3+7} = \frac{1}{10}$$

$$P(T \mid doc) = \frac{1}{2} \cdot \frac{2}{11} \cdot \frac{1}{11} = \frac{1}{121}$$

$$P(T' \mid doc) = \frac{1}{2} \cdot \frac{1}{10} \cdot \frac{1}{10} = \frac{1}{200}$$

$\Rightarrow$ Transylvania

5. **(40 pts) Link analysis**

Consider a web graph with three nodes 1, 2 and 3. The links are as follows: $1 \to 2$, $3 \to 2$, $2 \to 1$, $2 \to 3$.

(i) (20 pts) Write down the transition probability matrices for the surfer's walk with teleporting, for the following three values of the teleport probability mass α: (a) $\alpha = 0$ (i.e., no teleporting); (b) $\alpha = 0.3$ (i.e., 30% of probability mass is reserved for teleportation probabilities), and (c) $\alpha = 1$.

a)

	1	2	3
1	0	1	0
2	1/2	0	1/2
3	0	1	0

b)

	1	2	3
1	0.1	0.8	0.1
2	0.45	0.1	0.45
3	0.1	0.8	0.1

Teleportation probability = $\frac{0.3}{3} = 0.1$

c)

	1	2	3
1	1/3	1/3	1/3
2	1/3	1/3	1/3
3	1/3	1/3	1/3

(ii) (20 pts) For the configuration with $\alpha = 0.3$, compute the steady state vector after one iteration using the power method, assuming that at the beginning of the random walk the surfer is on node 1 with probability 1.0.

$$\vec{x} = [1, 0, 0]$$

$$P = \begin{bmatrix} 0.1 & 0.8 & 0.1 \\ 0.45 & 0.1 & 0.45 \\ 0.1 & 0.8 & 0.1 \end{bmatrix}$$

$$\vec{x} = \vec{x} \cdot P$$

$$= [0.1, 0.8, 0.1]$$

6. **(30 pts undergrads/40 pts grads) Link analysis for non-textual data**

You are hired by Spotify to revamp their music search engine. Your first task is to produce a *global* ranking of songs using link analysis. (Your colleagues have already used the "most played" heuristic, so you cannot use it again.)

a. (10 pts) What does your music graph look like? What is a node? What is an edge in this graph?

Node: song

Edge: two songs appear on the same playlist

b. (10 pts) Do you need a teleportation probability in this graph? Why or why not?

Yes, to connect songs that never show up in the same playlist.

c. (10 pts) What does the steady-state probability vector mean in this graph? That is, what is the node with the highest steady-state probability?

steady-state prob = indicator of popularity

d. (10 pts - GRAD STUDENTS ONLY) How would you improve this algorithm to make it topic sensitive? That is, your boss requires you to have a different song discovery algorithm for 100 predefined topics (e.g., rock, blues). How would you provide a different song ranking for each individual topic?

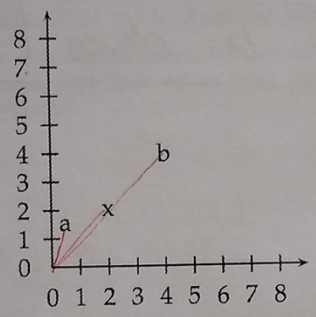
1. Classify songs into one or more topics, using either the Music Genome Project, or a text classifier based on the song's description.
2. Adjust teleportation probabilities to be biased towards songs that belong to the topics of interest to the user.

separate graphs per topic: -5
(PageRank must be joint) on a global graph

7. **(40 pts) Vector space classification**

In the figure below, which of the two vectors a and b is (i) (20 pts) most similar to x according to cosine similarity, and (ii) (20 pts) closest to x according to Euclidian distance? Show all work for computing these similarities and distances!

The vectors are: $a = (0.5 \;\; 1.5)$, $b = (4 \;\; 4)$, and $x = (2 \;\; 2)$.



(i)

$|a| = \sqrt{0.5^2 + 1.5^2} = \sqrt{0.25 + 2.25} = 1.58$

$|b| = \sqrt{4^2 + 4^2} = \sqrt{16 + 16} = 5.65$

$|x| = \sqrt{2^2 + 2^2} = \sqrt{4 + 4} = 2.82$

$\cos(a, x) = \frac{a \cdot x}{|a| \cdot |x|} = \frac{0.5 \cdot 2 + 1.5 \cdot 2}{1.58 \cdot 2.82} = \frac{4}{4.45} = 0.88$

$\cos(b, x) = \frac{b \cdot x}{|b| \cdot |x|} = \frac{4 \cdot 2 + 4 \cdot 2}{5.65 \cdot 2.82} = \frac{16}{15.93} \simeq 1$

b closer to x

(ii)

$$\text{Euclidian}(a, x) = \sqrt{(2-0.5)^2 + (2-1.5)^2} = \sqrt{1.5^2 + 0.5^2}$$

$$= \sqrt{2.25 + 0.25} = \sqrt{2.5} = 1.58$$

$$\text{Euclidian}(b, x) = \sqrt{(4-2)^2 + (4-2)^2} = \sqrt{2^2 + 2^2} = \sqrt{8} = 2.82$$

a is closer

8. **(40 pts) Neural text classification**

Consider the dataset below with three movie reviews:

Doc1: great action movie
Doc2: bad action movie bad action movie
Doc3: great bad bad bad movie

Doc1 is labeled Positive. *Doc2* and *Doc3* are both labeled Negative.

a. (20 pts) Generate the feature matrix for this dataset. The features are individual words, with their values being their frequency in the corresponding document.

	great	action	movie	bad
doc1	1	1	1	0
doc2	0	2	2	2
doc3	1	0	1	3

b. (20 pts) Given the training dataset you constructed above, compute the parameters of a Perceptron model, $\mathbf{w}$ and b, after one training epoch. Both $\mathbf{w}$ and b are initialized with 0s before training. Trace each iteration of the learning algorithm.

$w = [0\ 0\ 0\ 0]$ $b = 0$

score (doc1) $= 0 \Rightarrow$ add doc1:

$w = [1\ 1\ 1\ 0]$ $b = 1$

---

score (doc2) $= 1 \cdot 0 + 1 \cdot 2 + 1 \cdot 2 + 0 \cdot 2 + 1 \cdot 1 = 5 > 0$

$\Rightarrow$ subtract doc2:

$w = [1\ -1\ -1\ -2]$ $b = 0$

---

score (doc3) $= 1 \cdot 1 - 1 \cdot 0 - 1 \cdot 1 - 2 \cdot 3 + 0 \cdot 1 = -6 < 0$

No update!

$\Downarrow$

$w = [1\ -1\ -1\ -2]$ $b = 0$

9. **(30 pts - BONUS GRAD STUDENTS ONLY) Neural IR**

You are tasked by your supervisor with building a neural search system that operates over *very long* texts, which contain more than a hundred thousands words each. Your supervisor is very excited about deep learning, and she mandated that you use a transformer network for the task. That is, your system will score each document using the similarity between the `[CLS]` embedding produced by the transformer for this document and the query embedding (computed in a similar way). (a) What is the problem with this direction? (b) Design a better transformer that mitigates this problem (you are not allowed to change the data in any way).

a) Training is too expensive because the self-attention mechanism is quadratic in the text length.

b) Multiple solutions possible:

1. Change the attention mechanism to be linear, e.g. by considering a small, fixed window for attention.
2. Break documents into fragments, then aggregate all [CLS] embeddings into a single one.