Trial exam solutions

November 15-16, 2016

Part I

2 Summary statistics (5x2p)

Compute the range and variance of the following data: 17 5 3 9 49 53 11.

For variance use the formula $var(x) = \frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{x})^2$ and round the result to one decimal place.

Range: [50]
Median: [11]
Mean: [21]
Variance: [378.3]
Absolute Average Deviation (AAD): [17.1]

3 Attribute types (2x2p)

Classify the attribute *student ID* as binary, discrete, or continuous.

- Binary
- Discrete
- Continuous

Also classify it as qualitative (nominal or ordinal) or quantitative (interval or ratio).

- Qualitative
- Quantitative

4 Similarity (2x2p)

$$\begin{aligned}\mathbf{x} &= (1,1,0,1,0,0,1,0)\\ \mathbf{y} &= (1,0,0,1,0,0,1,1)\end{aligned}$$

Calculate the similarity of the above two data binary vectors.

Jaccard similarity: [0.6]
Cosine similarity: [0.75]

5 Classification (5x2p)

Train a Naive Bayes classifier given the document-term matrix and class labels in the table above. Use Laplace smoothing for computing term probabilities.

Notice that the description says **Laplace smoothing**! This is not to be confused with Jelinek-Mercer smoothing, what we typically use for Language Modeling retrieval. Laplace smoothing is computed using

$$P(t|c) = \frac{n(t,c)+1}{\sum_{t'} n(t',c) + |V|}, \tag{1}$$

where $|V|$ is the size of the vocabulary (here: 5).

- What is the prior class probability for C2? $P(C2) = \frac{2}{8} = \mathbf{0.25}$

	t1	t2	t3	t4	t5	class
Doc 1	1	0	2	2	0	C1
Doc 2	3	0	0	2	3	C3
Doc 3	0	4	2	0	0	C2
Doc 4	1	0	0	1	1	C1
Doc 5	0	0	0	2	1	C3
Doc 6	0	1	1	4	3	C3
Doc 7	0	2	1	0	0	C2
Doc 8	1	0	1	2	3	C3

Table 1: Document-term matrix.

- What is the (smoothed) probability of term “t4” belonging to C2? The term count for “t4” and C2 $n(t4, C2) = 0$. The sum term count for C2 $\sum_{t'} n(t', C2) = 9$. Substituting these back into Eq. 1: $\frac{0+1}{9+5} = \mathbf{0.071}$ (There was an error here in the original inspera answer.)
- What is the probability of a new document “t1” belonging to C1? $P(C1|t1) = P(C1) * P(t1|C1)$. The class probability is just the number of instances belonging to C1 divided by the total number of instances: $P(C1) = \frac{2}{8}$. $P(t1|C1)$ is computed using Eq. 1: $\frac{2+1}{8+5} = 0.23$. Multiplying the two gives **0.057**.
- What is the probability of a new document “t1 t4 t5” belonging to C3? $P(C3|t1\ t4\ t5) = P(C3) * P(t1|C3) * P(t4|C3) * P(t5|C3) = \frac{4}{8} * \frac{4+1}{27+5} * \frac{10+1}{27+5} * \frac{10+1}{27+5} = \mathbf{0.009}$
- Which class will document “t4 t4 t5” be classified to (i.e., for which class is $P(c|\text{“}t4\ t4\ t5\text{”})$ the highest?) $P(C1|t4\ t4\ t5) = 0.003$, $P(C2|t4\ t4\ t5) = 0.000$, $P(C3|t4\ t4\ t5) = 0.02$. The probability is highest for **C3**.

6 Classification (3p)

Assume a multiclass classification problem with 5 categories. Using the one-against-one strategy, how many binary classifiers are needed in total?

Answer: [$\frac{5*4}{2} = \mathbf{10}$]

7 Classification Evaluation (3x2p)

	actual label	predicted label
Instance 1	N	Y
Instance 2	Y	Y
Instance 3	Y	Y
Instance 4	N	Y
Instance 5	N	N
Instance 6	N	N
Instance 7	Y	Y
Instance 8	Y	N
Instance 9	N	N
Instance 10	N	Y

Table 2: Actual class labels vs. predictions.

Given the actual class labels and the predicted class labels for 10 instances in the table above, evaluate the classifier in terms of Precision, Recall, and F1-measure.

TP = 3, FN = 1, FP = 3, TN = 3

Precision: [$\frac{TP}{TP+FP} = \mathbf{0.5}$]
Recall: [$\frac{TP}{TP+FN} = \mathbf{0.75}$]
F1-measure: [$\frac{2RP}{R+P} = \mathbf{0.6}$]

8 Classification (4p)

Explain with your own words what overfitting means in practical terms for a decision tree classifier. How can it be avoided?
(Note: Copy-paste answers from the slides or from the book will not be accepted.)

In practical terms, overfitting for a decision tree classifier means that many of the leaf nodes will have only a few records belonging to them. This way the tree becomes "overspecialized" in the training examples, and it may not generalize well for instances not seen before.

Two main techniques help to avoid overfitting in a decision tree classifier are *pre-pruning* and *post-pruning*. The former stops growing the tree before it would overfit (e.g., if there are less than X records for a node then it will not be split further). The latter applies after the full tree is obtained, and tries to generalize in a bottom-up way, by trimming subtrees into leaf nodes.

Scoring: overfitting (2p), pre-pruning (1p), post-pruning (1p).

9 Clustering (10p)

	x_1	x_2	x_3	x_4
P1	2	0	5	2
P2	3	4	5	9
P3	1	5	8	1
P4	7	3	1	2

Table 3: Data points.

The table shows the vector representation of four data points that we want to cluster using the K-means method with k = 2. Assume that we use the dot product between vectors as the similarity measure between them. If we select points 3 and 4 as the initial centroids, what will be the cluster centroids in the next iteration?
Mind that the dot product is similarity and not a distance metric!

Centroid 1 is P3, Centroid 2 is P4. We compute the similarity of each point to both centroids, using the dot product.

	Centroid 1 (1,5,8,1)	Centroid 2 (7,3,1,2)
P1	2*1+0*5+5*8+2*1=44	2*7+0*3+5*1+2*2=23
P2	3*1+4*5+5*8+9*1=72	3*7+4*3+5*1+9*2=56
P3	1*1+5*5+8*8+1*1=91	1*7+5*3+8*1+1*2=32
P4	7*1+3*5+1*8+2*1=32	7*7+3*3+1*1+2*2=63

Table 4: Similarity of data points to centroids.

Each point gets reassigned to the closest cluster. Since we use a similarity metric it means that to the cluster with the highest similarity (which is equivalent to assigning it to the cluster with the smallest distance). So P1,P2,P3 ⇒ Cluster 1, P4 ⇒ Cluster 2. We need to recompute the centroid for Cluster 1. For cluster 2 it remains the same (since it only contains P4, as before).

Centroid 1: $(\frac{2+3+1}{3}, \frac{0+4+5}{3}, \frac{5+5+8}{3}, \frac{2+9+1}{3}) = \mathbf{(2,3,6,4)}$
Centroid 2: **(7, 3, 1, 2)**

10 Coding (3p)

```
# Compute the distance between two clusters.
# - sim is the similarity matrix between the data points.
#   For any two data points i and j 0 <= sim[i][j] <= 1.
# - c1 and c2 are the list of data points (indices)
#   belonging to each cluster; c1 and c2 are both non-empty.
```

```
    def cdist(sim, c1, c2):
        s = 1
            for i1 in c1:
                    for i2 in c2:
                            if sim[i1][i2] < s:
                                    s = sim[i1][i2]
        return 1-s
```

The above code computes the distance between two clusters based on a similarity matrix of data points. Which linkage function does it implement?

- Single link
- Complete link
- Group average
- None of the above

Part II

2 Similarity (3p)

We are given two documents, A and B, with term vectors, and we compute their cosine similarity. Then, we multiply all values by 2 in the term vector of A, and divide all values by 2 in the term vector of B. How will cosine similarity change?

- It will be 4 times the original
- It will be 2 times the original
- It will be 0.5 times the original
- It will be 0.25 times the original
- It will not change

3 Indexing (10p)

Doc 1	There are many interesting things to do in winter.
Doc 2	The weather this winter is not so great.
Doc 3	Do you prefer winter or summer?
Doc 4	Stop complaining about the weather!

Table 5: Documents.

Given the above set of documents, create an inverted index with position information.

- Apply standard tokenization, lowercasing, and stopword removal (but no stemming).
- Use a standard English stopword list; submit the list of words you identified as stopwords.
- Stopwords do not get indexed, but their positions count. For example, if you have ”word1 stopword word2”, then the position of word1 is 1 and the position of word2 is 3.
- Show one posting list per line. Use : to separate the payload. For example: x⇒y1:z1, y2:z2, ... (You should now what x, y, and z stand for.)

Stopwords (taken from the slides from Lecture 7): are, in, is, not, or, the, there, this, to

(We write x ⇒ y:z, where x is a term, y is a doc ID and z is a payload, i.e., term position in the document.)

```
about ⇒ 4:3
complaining ⇒ 4:2
do ⇒ 1:7, 3:1
interesting ⇒ 1:4
great ⇒ 2:8
many ⇒ 1:3
prefer ⇒ 3:3
so ⇒ 2:7
summer ⇒ 3:6
stop ⇒ 4:1
things ⇒ 1:5
you ⇒ 3:2
weather ⇒ 2:2, 4:5
winter ⇒ 1:9, 2:4, 3:4
```

Scoring: tokenization (2p), lowercasing (1p), stopword list submitted and looks standard (1p), stop word removal (1p), index in required format (1p), right index word positions (4p).

4 Retrieval (12p)

	term 1	term 2	term 3	term 4	term 5	term 6	length
Document 1	3	0	5	2	10	5	25
Document 2	4	3	5	2	1	5	20
Collection	100	50	80	93	100	25	1000

Table 6: Term statistics

$$BM25(q,d) = \sum_{t \in q} \frac{f_{t,d} \cdot (1+k_1)}{f_{t,d} + k_1(1 - b + b\frac{|d|}{avgdl})} \cdot idf_t \qquad idf_t = \log \frac{N}{n_t}$$

Compute retrieval scores using the BM25 algorithm.

- The collection row shows the number of documents that contain the given term; the collection contains 1000 documents in total.
- The average document length in the collection is 50.
- The BM25 parameters are k1 = 1.2 and b = 0.75.
- Use base-10 logarithm for the computations!

The query is a single term, "term 2".
The IDF value for term 2: $\log \frac{1000}{50} = 1.301$

- Doc 1 score
  $\frac{0\cdot(1+1.2)}{0+1.2(1-0.75+0.75\frac{25}{50})} \cdot 1.301 = \mathbf{0}$
- Doc 2 score
  $\frac{3\cdot(1+1.2)}{3+1.2(1-0.75+0.75\frac{20}{50})} \cdot 1.301 = \mathbf{2.346}$

The query is "term2 term2 term5".
We have already computed the score for term 2. The score for term 5 is as follows.
IDF: $\log \frac{1000}{100} = 1$
Doc 1: $\frac{10\cdot(1+1.2)}{10+1.2(1-0.75+0.75\frac{25}{50})} \cdot 1 = 2.046$
Doc 2: $\frac{1\cdot(1+1.2)}{1+1.2(1-0.75+0.75\frac{20}{50})} \cdot 1 = 1.325$

- Doc 1 score 0+0+2.046=**2.046**
- Doc 2 score 2.346+2.346+1.325=**6.017**

5 Retrieval Evaluation (10p)

	Query 1	Query 2
Algorithm A	1, 2, 6, 5, 9, 10, 7, 4, 8, 3	1, 2, 4, 5, 7, 10, 8, 3, 9, 6
Algorithm B	10, 9, 8, 7, 5, 4, 6, 2, 1, 3	1, 3, 2, 4, 5, 6, 8, 7, 10, 9
Ground truth	1, 4, 5	3, 6

Table 7: Retrieval evaluation.

The table shows, for two queries, the document rankings produced by ranking two different algorithms along with the list of relevant documents according to the ground truth. We assume that relevance is binary.

Answer the questions below. (5x2p)

- What is P5 (precision at rank 5) of Algorithm A on Query 1? [0.4]
- What is the Average Precision of Algorithm A on Query 1? [0.625]
- What is the Reciprocal Rank of Algorithm B on Query 2? [0.5]
- What is the Mean Reciprocal Rank of Algorithm B? [0.35]
- Which algorithm has higher Mean Average Precision? [A/B/the same]

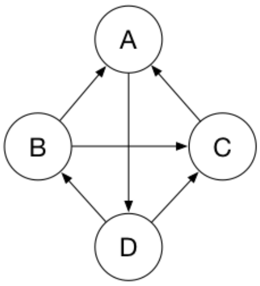
6 Retrieval (4p)

Explain the role of smoothing in Language Modeling. Also explain what would the effect be of setting the smoothing parameter in Jelinek-Mercer smoothing to 0 or to 1.

- The main role of smoothing is to avoid multiplying with zero term probability if the given term does not appear in the document (i.e., $P(t|d) = 0$ for any query term would result in $P(q|d) = 0$ if no smoothing is applied). (2p)
- If the smoothing parameter is set to 0, then no smoothing is applied; documents that do not contain all the query terms will be assigned a zero score. (1p)
- Setting the smoothing parameter to 1 will result in all documents being scored using the collection language model, i.e., all documents will get the same score. (1p)

7 PageRank (10p)

Compute the PageRank values for the following graph for two iterations.



The probability of a random jump (i.e., the parameter q) is 0.2.

	Iteration 0	Iteration 1	Iteration 2
A	0.25	0.35	0.31
B	0.25	0.15	0.15
C	0.25	0.25	0.21
D	0.25	0.25	0.33

8 PageRank (2p)

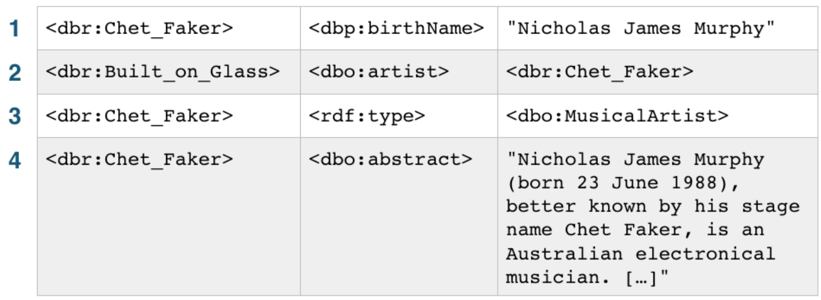
Assume that Page A has 10 in-links and Page B has 2 in-links. Which one has higher PageRank?

- Page A
- Page B
- They have the same
- It's not possible to tell

9 Entity retrieval (8p)

We want to create a fielded document representation for the entity "Chet Faker" given the information associated with him in a knowledge base. We use three fields:

- names: literal objects that contain the name of the entity
- attributes: all literal objects that are not already in names
- inlinks: all incoming relations (subjects where the given entity stands as object)



1	<dbr:Chet_Faker>	<dbp:birthName>	"Nicholas James Murphy"
2	<dbr:Built_on_Glass>	<dbo:artist>	<dbr:Chet_Faker>
3	<dbr:Chet_Faker>	<rdf:type>	<dbo:MusicalArtist>
4	<dbr:Chet_Faker>	<dbo:abstract>	"Nicholas James Murphy (born 23 June 1988), better known by his stage name Chet Faker, is an Australian electronical musician. […]"

Select for each RDF triple from the above image the field that it should be mapped to (or NONE if that triple is not mapped to any of the three fields).
Each correct answer is 2p, each incorrect answer is -1p.

	names 0	attributes	inlinks	NONE
1	[X]	[ ]	[ ]	[ ]
2	[ ]	[ ]	[X]	[ ]
3	[ ]	[ ]	[ ]	[X]
4	[ ]	[X]	[ ]	[ ]

10 Entity linking (4p)

We have an entity linking system that only returns entity annotations above a given confidence threshold. First, we run this system on some input text using 0.1 as the threshold. Then, we change the threshold to 0.9 and run the system on the same input. How will precision and recall change?

By increasing the threshold we restrict the system to return only high confidence annotations, i.e., it will return less, but higher quality annotations than before. Therefore, precision will increase (2p) and recall will drop (2p).