Unit 1 Review: Topics in Big Data

March 1, 2021

Exercise 1

You learned about an algorithm called ‘Minimum Hash‘ which was an efficient way to approximate the Jaccard Similarity of two strings. In the Minimum Hash algorithm, you first break up a string into words, and then return the minimum hashed value over all words.

In this problem, you will work on a variant called ‘Median Hash‘ which will: (1) break a string into words, (2) take the set of distinct words, and (3) return the median hashed value over all words.

(a) From lecture, we know that the probability that the Minimum Hash of two sets of words is equal is the same as the Jaccard Similarity of the sets. What about the probability that two sets have the same Median Hash? Explain your answer.

◇ Always equal to the Jaccard Similarity of the sets.

◇ Not necessarily equal to the Jaccard Similarity of the sets.

(b) Under which of the following conditions would the Median Hash be an exact estimator of Jaccard Similarity (meaning probability of the same median hash is equal to the Jaccard Similarity):

◇ Always equal to the Jaccard Similarity of the sets.

◇ If all strings have an odd number of words.

◇ If all strings have an even number of words.

◇ If the hash functions used are “order preserving”.

◇ None of the above

Exercise 2

You are given a dataset of 100 dates represented as strings in the format ‘MM-DD’. The resulting dataset looks as follows:

```
`NOV-11',
`DEC-12',
...
`NOV-13',
`MAR-02'
```

Assuming that each of the 100 given dates are unique, select the answer that best describes the effects of the following encoding schemes on the above dataset:

(a) Applying standard dictionary encoding [1] to the collection of dates will:

◇ Always reduce the size of the given dataset.

◇ Sometimes reduce the size of the given dataset.

◇ Will never reduce the size of the given dataset.

(b) Suppose, we calculate an optimal prefix-free code (like Huffman coding) [2] to the collection of dates will:

◇ Arbitrarily return different code lengths for each of the dates.

◇ Return an encoding with the same code length for each of the dates.

◇ Will return a failure because it assumes no tied item probabilities.

(C) Consider the following encoding algorithm:

Step 1. We split each of the strings into two collections, MM and DD:

```
[`NOV', `DEC', ..., `NOV', `MAR']
[`11', `12', ... `13', `02']
```

Step 2. We apply dictionary encoding independently to each collection.

Apply this encoding algorithm to the collection of dates will:

◇ Always reduce the size of the given dataset.

◇ Sometimes reduce the size of the given dataset.

◇ Will never reduce the size of the given dataset.

[1] Treating each full date as a data item

[2] Treating each full date as a data item

Exercise 3

Suppose, you are given a *single* hash function `hash(st)` that takes in a string as an argument and returns an integer from 1 to H. You are also given a list of N distinct strings.

We know the following about the hash function:

1. The set of possible distinct codes is less than the number of strings: $H < N$
2. Every possible code $1, ..., H$ appears at least once.
3. Every string is of length > 10.
4. You may assume the hash function is ideal over the set of all possible strings, i.e., simple uniform hashing assumption.

Based on these assumptions, evaluate the following statements.

(a) It is guaranteed that at least two different strings from the list have the same hash code:

__ True

__ False

(b) Suppose, I defined a new hash function based on the given hash function:

```
def h2(st):
    val = str(hash(st)) #casts the output to a string
    return hash(val)
```

the maximum value that `h2` could return is:

__ Always equal to H

__ Possibly less than H

__ Possibly greater than H

(c) Suppose, I defined a new hash function based on the given hash function:

```
def h2(st):
    st1 = st[1:] #remove the first letter from st
    return hash(st1)
```

under the assumptions above (Mark all that apply):

__ `h2` is independent from `hash`

__ Every possible code $1, ..., H$ appears at least once if you apply `h2` to each of the N strings.

__ `hash` and `h2` can be combined to make a code space of H^2.

__ `h2` is independent from `hash`, but will no longer satisfy the simple uniform hashing assumption.

Exercise 4

Let P and Q be discrete probability distributions over the same set of symbols $\mathcal{X}$. Assume, that $P(x) > 0, Q(x) > 0$ for all $x \in \mathcal{X}$. Let $H(P)$ denote the standard entropy measure of a probability distribution:

$$H(P) = -\sum_{x \in X} P(x) \cdot \log P(x)$$

Prove that:

$$H(P) \leq \sum_{x \in X} \frac{P(x)}{Q(x)}$$

Exercise 5

In class, we derived the optimal bias-variance tradeoff for a histogram w.r.t a continuous probability density function. In this problem you will derive the optimal solution for a discrete random variable. You may assume the following:

- Let X be a discrete random variable with D possible contiguous integer values, e.g, $0, ..., D - 1$.
- Observe n i.i.d observations of X.
- Equi-width partitioning of the histogram into M buckets each with $\frac{D}{M}$ values. You may assume for convenience that D is a multiple of M.
- The difference in probability mass between neighboring values (e.g., $P(i)$ and $P(i + 1)$) is bounded by a known parameter Δ.

Under the assumptions above derive an optimal value for M.