Data Science Toolbox Question Sheet

08.1 Algorithms

Daniel Lawson

Block 8

1. Why do we distinguish between average case and worst case in algorithmic complexity? Describe (with reasons) a situation in which each would be appropriate.
2. What is the name for an algorithm satisfying $x \in \mathcal{X} \rightarrow u \in \mathcal{U}\,[0,r)$?
3. Consider that we are working with a hash function. Under which circumstances would it be useful to consider a) predictability, b) locality, c) collisions, d) compute, and e) families of hash functions?
4. What is a hash table?
5. The error rate of a bloom filter is $(1-\exp(-kn/r))^k$. Given fixed n and r, differentiate this with respect to k. Show that the error rate is minimised when $k=(r/n)\ln(2)$.
6. Explain what Jaccard Similarity means. Why is this slow to compute naively when the feature space is large, and how does hashing help?
7. is $f(n)=4n\log(3n) \in \mathcal{O}(n^2)$?
8. is $2n+5 \in \Theta(n^2)$?
9. Consider the following pseudo-code. What is its time complexity as a function of `a`?

```
input a
algorithm:
    b=0
    while a>1
        a=a/2
        b=b+1
    end
    return b
```

10. There are many formal approaches to solving recusrive algorithm complexities. We will use *substitution*, where we **guess** a bound and demonstrate that it is true.

a. A recursive algorithm for $f(n)$ follows $T(n)=2T(n/2)+n$. Write the first 3 terms (i.e. for $n/8$).

b. Noting that we will have a logarithmic number of terms, we hypothesise that $f(n)=\mathcal{O}(n\log(n))$. State the inequality that must therefore hold, and substitute this into the recursion for $T(n)$. By retaining the inequality, find a constant factor that makes this true.