Data 101/CS 187: Data Engineering Fall 2026 Discussion 9 – November 13th, 2026

MapReduce, Sampling

Instructor: Aditya Parameswaran *Covers Lecture(s): 19 – 20*

I MapReduce

A MapReduce-style system, such as Apache Hadoop and Apache Spark, requires each stage of computation to be composed of a *map* function and a *reduce* function. The functions have the following type signatures:

$$Map(K, V) \rightarrow \langle K', V' \rangle^*$$
$$Reduce(K', \langle V' \rangle^*) \rightarrow \langle K'', V'' \rangle^*$$

1. *Word count.* Write a MapReduce program to count the appearance of each word in a set of sentences. Specifically, implement the following two functions:
   - The map function should take a sentence as input, split it into words, then output a dictionary of words and their counts in the sentence.
   - The reduce function should take a word, and a list of counts from the map tasks. It should output the total count of this word in all sentences.

```
def map_fn(document: str) -> dict[str, int]:



def reduce_fn(word: str, counts: list[int]) -> dict[str, int]:
```

2. Let us reason about how this program will be executed in a distributed MapReduce environment. Suppose the input strings are:
   - hello world
   - hello world mapreduce
   - mapreduce example
   - hello hello hello

   (a) There will be 4 mappers, each processing one sentence. What are their outputs?

(b) There will be 4 reducers. The *shuffle* stage, handled by the execution system, will dispatch all the mapper outputs to their corresponding reducers. What will be the inputs to each reducer? For each input value, which mapper will it come from?

(c) What will be the outputs of the reducers?

3. **[Challenge]** What if we wanted to compute the relative frequency of each word (its count divided by the total number of words across all sentences)? How might we change our MapReduce program? (Hint: Consider multiple maps/reduces)

4. Consider the following SQL query to compute the total number of page views of each Wikipedia article. The `daily_page_views` table stores the total views of an article by day, so we want to aggregate them all. How can we implement this in MapReduce style? First write in English what the map and reduce functions should do, then implement these functions.

```
SELECT title, SUM(num_views) FROM daily_page_views GROUP BY title;
```

II Reservoir Sampling

In this section, we will revisit the reservoir sampling algorithm introduced in lecture. Feel free to open the demo to follow along.

5. What is the first step of reservoir sampling assuming the reservoir is of size k?

6. Once the reservoir is filled with the initial k elements, how does the algorithm decide whether to replace an existing element when a new element is encountered?

7. After processing n elements, what is the probability that each element is in the reservoir?

8. **[Optional]** Theorem Statement: We are trying to prove that, at any step, each element in the dataset has an equal probability $\frac{k}{n}$ of being in the reservoir, where n is the total number of elements processed so far.

   (a) Base Case: Prove that when $n \leq k$, the probability holds.

(b) Induction Hypothesis: Assume that for a reservoir of size k, after n elements have been processed, each element in the dataset has an equal probability of being in the reservoir. What is this probability?

(c) Induction Step (considering the $(n+1)$-th element): Show that after processing $n+1$ elements, each element in the dataset still has an equal probability $\frac{k}{n+1}$ of being in the reservoir. Consider an element j that is already in the reservoir after n steps. What are the two cases for element j to remain in the reservoir, and what are their respective probabilities?

(d) What's the total probability that j remains in the reservoir after processing the $(n+1)$-th element? Hint: combine the probabilities from the previous cases.

(e) Combine the induction step with the induction hypothesis to find the probability that element j is still in the reservoir after $n+1$ steps.

III Stratified Sampling

9. You are estimating the average commute time for the Bay Area. The population has two groups:

   - Berkeley: 60% of population, mean = 25 mins.
   - SF: 40% of population, mean = 50 mins.

   You collect a sample of $n = 40$ people using the following two approaches:

   - Random Sample: 38 Berkeley, 2 SF
   - Stratified Sampling

   (a) Compute the sample mean for the random sample.

   (b) Compute the sample mean for the stratified sample.