Thesis 7

Anna Lucia Lamacchia
Statistics

January 11, 2024

"Online" Algorithms (Data Streams)

1 Ideas

Online algorithms are designed to process data as it arrives in a sequential manner, without having access to the entire dataset in advance. This is particularly relevant in the context of data streams, where the data is continuously generated or arrives in a rapid, sequential fashion. Online algorithms are characterized by making decisions or computations on-the-fly with limited memory and computational resources.
In the context of data streams, online algorithms are crucial for applications where it is impractical or impossible to store the entire dataset due to its size or dynamic nature. Examples include real-time analytics, network monitoring, and financial data analysis.
Common challenges in the online algorithm paradigm include maintaining accurate statistics, detecting trends, and making predictions using limited memory and processing time. Online algorithms often need to balance trade-offs between accuracy and resource efficiency.
Unlike traditional batch processing, these algorithms adapt dynamically as new data arrives, making them well-suited for applications dealing with large, ever-changing datasets.
Key Ideas for these algorithms are:

1. **One-Pass Processing**: Online algorithms are designed to process data in a single pass, without the need to store the entire dataset. This makes them memory-efficient and suitable for scenarios where storing the entire dataset is impractical.

2. **Dynamic Adaptation**: Online algorithms adapt to changes in the data distribution over time. They make decisions or update statistics based on the current data without revisiting past observations.

3. **Constant Memory Usage**: As data streams through the algorithm, it maintains a constant amount of memory, ensuring scalability for large datasets.

2 Simulation

In this simulation, a data stream is generated, and the online algorithm computes the running average as the data arrives sequentially. The running average provides a continuously updated estimate of the mean of the data stream. This example illustrates the concept of processing data in an online manner, adapting to new information as it becomes available.

```python
import numpy as np
import matplotlib.pyplot as plt

# Simulate a data stream
np.random.seed(42)
data_stream = np.random.normal(loc=0, scale=1, size=1000)

# Online algorithm for computing running average
running_average = np.zeros(len(data_stream))
cumulative_sum = 0

for i in range(len(data_stream)):
    # Update running sum
    cumulative_sum += data_stream[i]

    # Compute running average
    running_average[i] = cumulative_sum / (i + 1)

# Plot the data stream and running average
plt.plot(data_stream, label='Data Stream')
plt.plot(running_average, label='Running Average', linestyle='--', color='red')
plt.xlabel('Time')
plt.ylabel('Value')
plt.legend()
plt.title('Online Algorithm: Running Average of Data Stream')
plt.show()
```

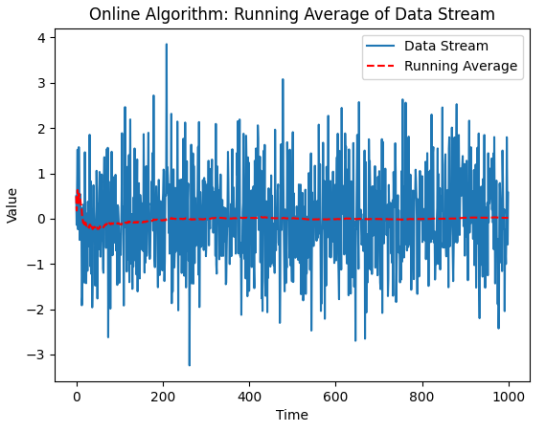

Figure 1: Code