Assignment 3 Report

Rishabh Kumar

April 1, 2025

1 Implementation Decisions

The implementation follows a parallel processing approach using CUDA to efficiently handle large matrices. The key steps in the implementation are:

- Computing the histogram of matrix elements using atomic operations in the GPU kernel.
- Calculating the prefix sum on the CPU to avoid race conditions and ensure correctness.
- Using another kernel to rearrange elements based on their computed positions.
- Ensuring that input matrices are not modified directly to maintain integrity checks.

The implementation ensures correctness by verifying that the input and output matrices have the same multiset of elements. Any discrepancies in element counts are flagged as errors.

2 Performance Discussion

The performance is influenced by matrix size and the range of values:

- Small value ranges lead to efficient histogram computation and sorting, reducing execution time.
- Larger value ranges require more memory and increase kernel execution time due to atomic operations.
- As the number of matrices (M) increases, execution time scales linearly.
- CUDA performance degrades for very large matrices with high value ranges due to increased contention in atomic operations and memory bandwidth limitations.

The execution times observed in the test cases confirm that the approach is efficient for small and medium-sized value ranges but becomes slower as the value range grows significantly.