CSCI 6351 Data Compression

Term Project Proposal:
Comparative Analysis of Probability Models for Arithmetic Coding

Oscar Fang
G42568236

November 10, 2025

1 Project Overview

1.1 What: Comparative Analysis of Probability Models for Arithmetic Coding

This project will implement and compare different probability models used with arithmetic coding for data compression. Specifically, I will evaluate:

- **Markov Models:** 1st, 2nd, and 3rd order models that predict symbols based on previous context
- **Finite State Machine (FSM) Models:** State-based models that adapt to patterns like runs of symbols
- **Neural Network Models:** RNN/LSTM-based predictors that learn complex dependencies

Each model will be integrated with an arithmetic coder implemented in MATLAB and tested on diverse datasets including text files, binary data, and structured sequences (e.g., DNA).

1.2 Why: Motivation and Objectives

Arithmetic coding can achieve near-optimal compression when paired with an accurate probability model. However, different models have different strengths:

- Simple models (1st order Markov) are fast but may miss patterns
- Complex models (3rd order, neural networks) can capture more dependencies but require more memory and computation
- Specialized models (FSM) may excel on specific data types

Key Questions:

1. How does model complexity affect compression ratio and speed?
2. Which models work best for different data types?
3. What are the practical trade-offs between compression quality and computational cost?

Expected Outcomes:

- Quantitative comparison of compression performance across models and datasets
- Analysis of memory usage, encoding/decoding time, and compression ratios
- Practical guidelines for choosing models based on application requirements
- Open-source MATLAB implementation for educational use

1.3 How: Implementation Approach

All models will be implemented in MATLAB with a common arithmetic coding framework:

1. **Arithmetic Coder:** Base encoder/decoder with interval arithmetic and E1/E2/E3 scaling
2. **Probability Models:** Each model provides probability distributions for the next symbol given context
3. **Adaptive Updates:** Models update as data is processed (online learning)

Datasets:

- Text: English text, source code (10KB - 1MB files)
- Binary: Executable files, images
- Structured: DNA sequences

Evaluation Metrics:

- Compression ratio (compressed size / original size)
- Encoding and decoding time
- Memory usage
- Comparison against baselines (Huffman, gzip, entropy)

2 Timeline and Deliverables

2.1 Timeline

Week	**Tasks**
1	Implement arithmetic coder and 1st/2nd order Markov models
2	Implement 3rd order Markov and FSM models
3	Implement neural network models and run experiments
4	Analyze results and write final report

2.2 Deliverables

1. Unified comparison of traditional and neural probability models for arithmetic coding
2. Quantitative analysis of compression-complexity trade-offs
3. Practical recommendations for model selection
4. Educational MATLAB implementation with documentation