Ph.D. Research Proposal: AI-Enhanced Automata-Based Data Compression

Ehsan KhademOlama
Applicant for Ph.D. Position at KTH Royal Institute of Technology

March 26, 2025

1 Introduction

Data compression plays a pivotal role in modern computing, facilitating the efficient storage and transmission of increasingly large datasets across domains such as genomics, image processing, and time-series analysis. Traditional lossless compression algorithms, such as Lempel-Ziv-Welch (LZW) and gzip, rely on dictionary-based techniques to eliminate redundancy by replacing recurring substrings with shorter codes. However, these methods often fail to capture complex, non-linear patterns inherent in many real-world datasets. Recent advancements in artificial intelligence (AI) have demonstrated the ability to detect intricate patterns through machine learning, while automata theory offers a structured framework for modeling and regenerating sequences efficiently.

This research proposes a novel lossless data compression method that integrates AI-driven pattern detection with automata-based pattern reproduction. The approach involves three key steps:

- Employing AI techniques, such as clustering or neural networks, to identify recurring patterns within a dataset and constructing a compact dictionary of these patterns.
- Learning automata, such as finite state machines (FSMs) or cellular automata (CA), capable of regenerating each identified pattern from minimal initial conditions—ideally a zero state.
- Encoding the original data by referencing the learned automata and their initial conditions, thereby achieving high compression ratios with provable theoretical guarantees.

The proposed method aims to push the boundaries of data compression by combining the pattern-learning capabilities of AI with the generative power of automata, offering both practical efficiency and theoretical insights. This research aligns with the focus areas of KTH Royal Institute of Technology, including theoretical foundations, randomized algorithms, and probabilistic data structures, and seeks to contribute to both academic understanding and real-world applications.

2 Background and Motivation

Lossless data compression ensures that compressed data can be perfectly reconstructed, making it essential for applications where data integrity is paramount. Existing techniques can be categorized as follows:

- **Dictionary-Based Methods**: Algorithms like LZW [1] and LZ78 [2] dynamically construct dictionaries of frequently occurring substrings, replacing them with shorter codes. These methods excel with simple, repetitive data but may miss higher-order patterns. - **Statistical Methods**: Huffman coding [3] and arithmetic coding [4] leverage symbol frequency distributions to assign shorter codes to more frequent elements, approaching the entropy limit for statistically predictable data. - **Grammar-Based Compression**: Techniques such as Sequitur [5] and Re-Pair [6] generate context-free grammars to represent data, effectively compressing hierarchical or repetitive structures.

Recent developments have introduced AI and automata into the compression landscape: - **AI-Driven Compression**: Neural networks, including variational autoencoders (VAEs) [7] and generative adversarial networks (GANs) [8], have been used to learn data distributions for compression, particularly in multimedia applications. Learned data structures like Bit-Swap [9] combine latent variable models with entropy coding for competitive results. - **Automata-Based Compression**: Research has explored finite automata [10] and cellular automata [11] for compressing structured data, such as graphs or images. Grammar-based graph compression [12] further extends these ideas to complex structures.

Despite these advances, the integration of AI with automata for lossless compression remains largely unexplored. Current methods lack a unified framework that leverages AI's pattern recognition with automata's sequence generation capabilities. This research aims to address this gap, offering a hybrid approach that could outperform existing techniques in both compression ratio and computational efficiency, while providing theoretical guarantees rooted in information theory and formal language theory.

3 Novelty and Plausibility

The proposed research direction sits at the intersection of AI-driven data analysis and formal language theory applied to data compression. While both AI techniques and automata/grammar-based methods have been independently explored for compression, their specific synthesis as proposed here represents a novel approach.

Novelty: The core novelty lies not in using AI or automata individually, but in the proposed pipeline:

1. *Advanced AI for Pattern Discovery:* Leveraging modern AI (e.g., deep learning embeddings, clustering on complex features) to identify potentially non-linear, approximate, or context-dependent recurring patterns that may elude traditional dictionary or simpler grammar methods.

2. *Explicit Generative Automata Learning:* Crucially, for each class of patterns identified by the AI, the aim is to *explicitly learn a dedicated, potentially minimal, generative automaton* (such as an FSA or CA) capable of reproducing instances of that pattern. This contrasts with typical AI compression methods that often function as probabilistic models or end-to-end transformations, and with grammar methods where the grammar itself is the direct result of pattern discovery.

3. *Encoding via Automata References:* The compressed representation relies on referencing these learned automata and minimal initial conditions, treating them as compact generative subroutines.

This tight coupling of sophisticated pattern recognition with the formal generative power of learned automata is the primary unique contribution explored in this proposal.

Plausibility and Challenges: The conceptual basis is plausible: if significant redundancy exists and can be captured by generative automata whose description size (plus references) is smaller than the original pattern occurrences, compression is achieved.

However, the technical feasibility faces significant research challenges, primarily centered on the automata learning phase:

- *Automata Learning Complexity:* Inferring a minimal automaton (e.g., the smallest DFA) consistent with a set of examples is computationally hard (often NP-hard). Developing efficient and effective learning algorithms that work with the potentially noisy or complex pattern representations derived from AI output is a major research task. Standard algorithms (like L*, RPNI) may require adaptations or new approaches.

- *Generation from Minimal State:* The objective of generating patterns from a minimal (e.g., zero) state adds complexity. It might require the automata to be more intricate or necessitate relaxing this constraint to allow small initial seeds or parameters provided by the AI.

- *Model Selection and Scalability:* Choosing the appropriate type of automaton (FSA, CA, transducer, etc.) for different patterns and ensuring the learning process scales to large datasets and numerous patterns are critical hurdles.

- *Performance Benchmarking:* Achieving compression ratios and speeds competitive with highly optimized state-of-the-art compressors (e.g., zstd, LZMA) is a challenging long-term goal, given the potential computational overhead of the AI analysis and automata learning steps.

Addressing these challenges, particularly the core problem of efficiently learning compact generative automata from AI-detected patterns, forms the heart of the proposed research. While ambitious, successfully tackling these issues would yield significant contributions to both the theory and practice of data compression. This makes the topic a suitable and potentially high-impact focus for Ph.D. research.

4 Research Objectives

The research seeks to achieve the following objectives:

1. **Develop AI-Driven Pattern Detection Algorithms**: Create efficient methods to identify recurring, potentially complex patterns in datasets using machine learning techniques, forming the foundation of the compression dictionary.

2. **Design Automata Learning Methods**: Develop algorithms to infer automata (e.g., FSAs, CAs) that can efficiently regenerate detected patterns from minimal initial conditions, addressing the core technical challenge.

3. **Create a Compression Framework**: Build an integrated system that encodes data using references to the learned automata and their initial conditions, optimizing for compression efficiency.

4. **Analyze Theoretically**: Derive formal bounds on compression ratios and computational complexity using information theory and automata theory, quantifying the performance potential and limitations.

5. **Evaluate Practically**: Implement and test the framework on diverse datasets (e.g., genomics, images, time-series), benchmarking against state-of-the-art compression algorithms.

5 Methodology

The research will proceed in five phases, each aligned with an objective:

5.1 Phase 1: Pattern Detection

- **Techniques**: Explore and adapt unsupervised learning methods such as k-means clustering on feature embeddings [13], autoencoders/VAEs [14] for latent space similarity, or sequence analysis methods possibly inspired by transformers [15] to detect recurring, potentially complex patterns. - **Randomized Algorithms**: Incorporate techniques like random projections or sampling to enhance scalability for large datasets. - **Deliverable**: A pattern dictionary containing representations of recurring structures identified within the data.

5.2 Phase 2: Automata Learning

- **Models**: Investigate learning algorithms for finite state automata (FSA) and cellular automata (CA), selecting or adapting models based on pattern complexity and data characteristics. - **Learning Algorithms**: Research and develop novel or adapted algorithms for grammatical inference or automaton learning, focusing on generating target patterns from minimal states. Explore connections to reinforcement learning or evolutionary methods if direct inference proves too difficult. Probabilistic models like hidden Markov models (HMMs) [16] might serve as an intermediate step or alternative. - **Deliverable**: A set of learned automata corresponding to the dictionary patterns, along with the algorithms used for their inference.

5.3 Phase 3: Compression Framework

- **Encoding**: Design an encoding scheme that efficiently replaces pattern occurrences with references to their corresponding learned automata and necessary initialization data. Employ entropy coding (e.g., Huffman or arithmetic coding) on the resulting stream of references and literal data. - **Optimization**: Develop strategies to minimize the total compressed size, considering the trade-off between the size of the automata descriptions, the pattern dictionary, and the encoded reference stream. - **Deliverable**: A functional prototype of the compression and decompression system.

5.4 Phase 4: Theoretical Analysis

- **Information Theory**: Analyze the potential compression efficiency in relation to data entropy and pattern characteristics, aiming to establish theoretical bounds or comparisons. - **Computational Complexity**: Assess the time and space complexity of the pattern detection, automata learning, encoding, and decoding phases. - **Deliverable**: Formal analysis providing theoretical guarantees and insights into the algorithm's performance characteristics.

5.5 Phase 5: Practical Evaluation

- **Implementation**: Develop a robust prototype implementation, potentially optimizing critical components. - **Datasets**: Evaluate performance on diverse benchmark datasets, including genomic sequences, natural images, text corpora, and time-series data. - **Comparison**: Benchmark compression ratio and speed against established algorithms like gzip, bzip2, LZMA, zstd, and potentially relevant grammar-based or neural compression methods. - **Deliverable**: Comprehensive empirical results, comparisons, and analysis published in relevant venues.

6 Expected Contributions

The research is expected to contribute: - **Novel Algorithms**: Development of integrated AI-automata methods for lossless data compression. - **Theoretical Insights**: New understanding and potentially formal results regarding the compressibility of data using learned generative automata models. - **Practical Tools**: Open-source implementation(s) of the developed algorithms, enabling reproducibility and potential real-world application. - **Advancement at KTH**: Contributions aligned with KTH's strengths in theoretical computer science, algorithms (including randomized), and machine learning foundations.

7 Timeline

A projected timeline over four years: - **Year 1**: Literature review; Development and initial experimentation with AI-driven pattern detection methods (Phase 1). - **Year 2**: Focused research on automata learning algorithms (Phase 2); Initial integration with pattern detection. - **Year 3**: Construction and refinement of the compression/decompression framework (Phase 3); Commence theoretical analysis (Phase 4). - **Year 4**: Comprehensive practical evaluation (Phase 5); Finalize theoretical analysis; Dissertation writing and defense.

References

[1] Welch, T. A. (1984). "A Technique for High-Performance Data Compression." *Computer*, 17(6), 8-19.

[2] Ziv, J., & Lempel, A. (1978). "Compression of Individual Sequences via Variable-Rate Coding." *IEEE Transactions on Information Theory*, 24(5), 530-536.

[3] Huffman, D. A. (1952). "A Method for the Construction of Minimum-Redundancy Codes." *Proceedings of the IRE*, 40(9), 1098-1101.

[4] Rissanen, J. (1976). "Generalized Kraft Inequality and Arithmetic Coding." *IBM Journal of Research and Development*, 20(3), 198-203.

[5] Nevill-Manning, C. G., & Witten, I. H. (1997). "Identifying Hierarchical Structure in Sequences: A Linear-Time Algorithm." *Journal of Artificial Intelligence Research*, 7, 67-82.

[6] Larsson, N. J., & Moffat, A. (2000). “Off-Line Dictionary-Based Compression.” *Proceedings of the IEEE*, 88(11), 1722-1732.

[7] Ballé, J., Minnen, D., Singh, S., Hwang, S. J., & Johnston, N. (2018). “Variational Image Compression with a Scale Hyperprior.” *arXiv preprint arXiv:1802.01436*. [Online]. Available: `https://arxiv.org/abs/1802.01436`

[8] Agustsson, E., Mentzer, F., Tschannen, M., Cavigelli, L., Timofte, R., Benini, L., & Gool, L. V. (2019). “Generative Adversarial Networks for Extreme Learned Image Compression.” *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 221-231.

[9] Kingma, D. P., Abbeel, P., & Ho, J. (2019). “Bit-Swap: Recursive Bits-Back Coding for Lossless Compression with Hierarchical Latent Variables.” *arXiv preprint arXiv:1905.06845*. [Online]. Available: `https://arxiv.org/abs/1905.06845`

[10] Goswami, M., Pagh, R., & Silvestri, F. (2015). “Automata and Graph Compression.” *arXiv preprint arXiv:1502.07288*. [Online]. Available: `https://arxiv.org/abs/1502.07288`

[11] Venkateswarlu, N. B., & Boyle, R. D. (1997). “Data Compression and Encryption Using Cellular Automata Transforms.” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 27(6), 926-931.

[12] Maneth, S., & Peternek, F. (2017). “Grammar-Based Graph Compression.” *Information and Computation*, 253, 233-252.

[13] Hartigan, J. A., & Wong, M. A. (1979). “Algorithm AS 136: A K-Means Clustering Algorithm.” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1), 100-108.

[14] Kingma, D. P., & Welling, M. (2013). “Auto-Encoding Variational Bayes.” *arXiv preprint arXiv:1312.6114*. [Online]. Available: `https://arxiv.org/abs/1312.6114`

[15] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). “Attention is All You Need.” *Advances in Neural Information Processing Systems*, 30.

[16] Rabiner, L. R. (1989). “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.” *Proceedings of the IEEE*, 77(2), 257-286.