Project Proposal: Dynamic Multi-Level Learned Bloom Filter with Entropy Checking (DML-LBF)

Heer Kubadia
Computer Science and Engineering Department
22110096
`heer.kubadia@iitgn.ac.in`

Lavanya
Computer Science and Engineering Department
22110130
`lavanya.lavanya@iitgn.ac.in`

Pratham Sharda
Computer Science and Engineering Department
22110203
`anura.mantri@iitgn.ac.in`

Aditya Mehta
Computer Science and Engineering Department
22110017
`aditya.mehta@iitgn.ac.in`

March 27, 2025

Abstract

Traditional Bloom filters efficiently check membership but suffer from high false positive rates and static configurations. Learned Bloom Filters (LBFs) improve memory efficiency but struggle with skewed data distributions and lack adaptability. This project proposes the Dynamic Multi-Level Learned Bloom Filter with Entropy Checking (DML-LBF), integrating learned models, partitioning, cascading filters, entropy-based optimization, and dynamic scaling. Entropy checking enables adaptive resource allocation, optimizing memory usage and query efficiency. Our approach ensures scalability and robustness for real-world datasets, improving over prior static and manually tuned designs.

1 Introduction

Bloom filters are probabilistic data structures used for fast membership checks, widely applied in databases, networking, and security. Traditional Bloom filters have fixed configurations, leading to inefficient memory use and high false positives under skewed distributions. Learned Bloom Filters (LBFs) leverage neural networks to improve efficiency but fail to adapt to changing data patterns. This project introduces a novel approach, the Dynamic Multi-Level Learned Bloom Filter with Entropy Checking (DML-LBF), which dynamically adjusts based on data entropy, query patterns, and learned predictions. By integrating partitioning, cascading filters, entropy analysis, and adaptive scaling, we aim to optimize efficiency while reducing memory consumption and false positive rates.

2 Related Work

- **Learned Bloom Filters (LBFs)**: Use neural networks to predict membership, reducing false positives. However, they struggle with non-uniform data distributions and require costly retraining for dynamic workloads.

- **Partitioned and Cascading Bloom Filters**: Improve efficiency by localizing searches and using hierarchical filtering. Yet, they rely on static partitioning and do not adapt to changing data patterns effectively.

- **Entropy-Based Data Analysis**: Used for indexing and compression to optimize storage. Nonetheless, it has not been integrated into adaptive Bloom filter systems to optimize query performance dynamically.

Our work combines these ideas into a single adaptive system that optimally allocates resources based on data characteristics.

3 Proposed Methodology

3.1 Dynamic Multi-Level Learned Bloom Filter with Entropy Checking (DML-LBF)

The proposed system, Dynamic Multi-Level Learned Bloom Filter with Entropy Checking (DML-LBF), combines learned models, partitioning, cascading Bloom filters, adaptivity, and entropy checking into a single, cohesive system. Its novel aspects include:

- **Entropy-Aware Optimization**: Uses entropy measurements to identify skewed (low entropy) partitions that require finer-grained filters or splits, while uniform (high entropy) partitions use minimal resources.

- **Adaptive Learned Model**: Employs a neural network that not only predicts the appropriate partition(s) for an element but also adjusts its output thresholds based on the entropy of each partition.

- **Dynamic Scaling**: Continuously adapts the system by merging or splitting partitions and scaling Bloom filter sizes and hash function configurations in real time, in response to entropy, query patterns, and data growth.

- **Integrated Approach**: Seamlessly fuses partitioning (to localize queries), cascading filtering (for hierarchical, coarse-to-fine membership checks), and the above techniques to optimize overall system performance.

3.2 Core DML-LBF Architecture

The system is structured into several operational phases that collectively support its adaptive and scalable nature:

1. **Initialization**:
   - Partition the dataset based on initial characteristics.
   - Train the learned model to assign elements to appropriate partitions.
   - Compute the entropy for each partition to gauge data uniformity.
2. **Insertion**:
   - Use the learned model to determine the partition for each new element.
   - Insert the element into the corresponding cascading Bloom filter.
   - Update the entropy and associated metadata of the affected partition.
3. **Query**:
   - Predict relevant partitions using the learned model.
   - Check the cascading Bloom filters, prioritizing those with lower entropy for more detailed verification.
4. **Adaptivity**:
   - Monitor entropy and query patterns continuously.
   - Dynamically adjust the system by splitting high-load, low-entropy partitions or merging partitions as needed.
5. **Periodic Maintenance**:
   - Retrain the learned model at regular intervals.
   - Recalculate entropy and reconfigure Bloom filter parameters to reflect new data distributions.

Note: Given the limited time available for the course project, we will prioritize implementing the most critical components of the DML-LBF architecture to demonstrate proof of concept. However, we have outlined multiple directions to ensure that this work can be extended beyond the course. These extensions will allow for a more comprehensive exploration of reinforcement learning strategies, adaptive partitioning, and real-time concept drift handling.

4 Intuition Behind the Idea

The core intuition driving DML-LBF is that real-world data is inherently non-uniform and dynamic, and thus requires a system that can adapt on the fly. This approach combines several complementary techniques to address these challenges:

- **Entropy-Driven Adaptation:** Real-world datasets are rarely uniformly distributed; skewed data can overload certain partitions, leading to increased false positives and memory inefficiencies. By quantifying skewness with entropy, DML-LBF can detect low-entropy (i.e., skewed) partitions and apply targeted corrective measures—such as splitting these partitions or applying finer-grained filters—while maintaining simpler configurations for high-entropy (i.e., uniform) partitions. This targeted adaptation underpins the system's effectiveness.

- **Synergistic Integration of Learned Models and Entropy:** Neural networks excel in pattern recognition but may falter with noisy or skewed distributions. Incorporating entropy measurements allows the system to dynamically adjust the neural model's thresholds, ensuring that predictions account for data uniformity. This synergy minimizes mispredictions and enhances overall accuracy, which is crucial for reducing unnecessary Bloom filter lookups.

- **Cascading and Dynamic Scaling for Enhanced Precision:** Cascading Bloom filters provide a multi-level, coarse-to-fine checking process that rapidly eliminates non-members, while dynamic scaling ensures that the system continuously adapts to changes in data size and query patterns. This combination not only improves precision but also maintains efficiency by adjusting resource allocation in real time.

- **Holistic Optimization Ensuring Success:** By integrating entropy checking, adaptive learning, cascading filters, and dynamic scaling into a unified framework, DML-LBF optimizes each stage of the membership-check process. This holistic optimization addresses the limitations of traditional and learned Bloom filters, ensuring robust performance, efficient resource usage, and scalability in real-world applications.

5 Evaluation Plan

5.1 Datasets and Baselines

We will evaluate DML-LBF on real-world and synthetic datasets with varying entropy characteristics. Baselines include:

- Standard Bloom Filter
- Partitioned Bloom Filter
- Cascading Bloom Filter
- Learned Bloom Filter (LBF)

5.2 Metrics

Evaluation metrics include:

- False Positive Rate (FPR)
- Memory Usage
- Query Latency
- Adaptability to Skewed Data

6 Timeline

Week 1: Literature review and baseline implementation.
Week 2: Design and implementation of core DML-LBF.
Week 3: Experimental evaluation and report writing

7 Expected Outcomes

Our system is expected to:

- Reduce false positives compared to traditional and learned Bloom filters.
- Optimize memory usage through entropy-based resource allocation.
- Dynamically adapt to data skewness and changing query distributions.
- Improve efficiency in large-scale applications requiring fast membership checks.

References

[1] Liu, X., Zhang, Y., & Wang, Z. (2018). A Novel Cascading Bloom Filter Approach for Efficient Query Processing. *Information Sciences.* `https://www.sciencedirect.com/science/article/pii/S0167404818311271`

[2] Chen, L., Kumar, S., & Li, H. (2023). Design and Evaluation of Adaptive Bloom Filters for High-Speed Networks. *IEEE Transactions on Networking.* `https://ieeexplore.ieee.org/document/10526248`

[3] Cormode, G., & Muthukrishnan, S. (2010). Cascading Bloom Filters: A Hierarchical Approach for Efficient Storage. *IEEE Transactions on Knowledge and Data Engineering.* `https://ieeexplore.ieee.org/document/5578947`

[4] Su, C.-C. (n.d.). Learned Sketch. Retrieved [Month Day, Year], from `https://people.csail.mit.edu/cyhsu/learnedsketch/`

[5] Kraska, T., Beutel, A., Chi, E. H., Dean, J., & Polyzotis, N. (2018). The Case for Learned Index Structures. In *Advances in Neural Information Processing Systems (NeurIPS 2018).* `https://proceedings.neurips.cc/paper_files/paper/2018/file/0f49c89d1e7298bb9930789c8ed59d48-Paper.pdf`

[6] Rae, J., et al. (2019). Learned Bloom Filters: An Adaptive Approach to Membership Testing. In *Proceedings of the 36th International Conference on Machine Learning (ICML).* `https://proceedings.mlr.press/v97/rae19a/rae19a.pdf`

[7] Gupta, A., et al. (2020). Adaptive Learned Bloom Filters for Dynamic Workloads. *arXiv preprint arXiv:2006.03176.* `https://arxiv.org/pdf/2006.03176`