Using Machine Learning To Improve Count-Min Sketches

COMP 480 Final Report

Taha Hasan (th43)

December 14, 2023

1. Problem Statement

This project focuses on enhancing Count-Min Sketch (CMS), a widely-used frequency estimation algorithm in data stream analysis. The count-min sketch, typically effective for processing large data volumes with limited storage, does not leverage inherent patterns in the input data for improved performance. I explore enhancing the count-min sketch algorithm through the implementation of the learned oracle architecture described in Hsu et al. (2019). This architecture is designed to exploit patterns in the input data, thereby improving the accuracy of frequency estimation by Count-min Sketch.

To improve the resource-accuracy tradeoff of Count-Min Sketch, I augment the count-min sketch with a learned oracle that uses machine learning to predict heavy hitters in the data stream. If the learned oracle predicts an element as a heavy hitter, its count will be stored in a separate bucket. If it does not, the element's count will be stored normally in the count-min sketch. Hsu et al. (2019) used an RNN to implement this learned oracle. However, RNNs take up a lot of memory (which was amortized over the training time in Hsu et al) and need a lot of training data. I explore if more lightweight machine learning models can be used to also improve the space-accuracy tradeoff of Count-Min Sketch without using as much memory or needing as much data as an RNN. To implement the learned oracle, I test four different machine learning models: logistic regression, support vector machines (SVMs), decision trees, and perceptrons. I test each of these variations and explore their accuracy-resource tradeoffs.

2. Literature Survey

Two research papers in particular were most important in providing existing related ideas to solve the problem being addressed by this project.

Firstly, the paper “Learning-Based Frequency Estimation Algorithms“ by Hsu et al. (2019) is directly relevant to this project as it tackles the same core challenge: improving frequency estimation in data streams. The researchers' approach of using machine learning to identify patterns in the data informs this project, especially in the way they use a learned oracle which uses a subset of the data to learn 'heavy hitters'. Their technique of pattern recognition using a learned oracle and its application in optimizing data bucket allocation in Count-min Sketch is the backbone of my project, and I test implementing this technique using various machine learning models. The study's success in reducing collisions and estimation errors through machine learning serves as a proof-of-concept that machine learning can be used to improve the resource-accuracy tradeoff of Count-min Sketch.

Secondly, the study ”Classification of Malicious URLs Using Machine Learning" by Abad et al. (2023) offers insights into the effective application of various machine learning models (such as SVMs and neural networks) in analyzing data streams of strings. The methodologies they employ for selecting and optimizing machine learning models based on the specific characteristics of the data guided my approach in applying different machine learning models to the Count-Min sketch, ensuring both efficiency and accuracy in frequency estimation. Particularly, their technique of how to use decision trees and SVMs for text data classification was very useful to me.

3. Hypothesis

The augmentation of Count-Min Sketch with lightweight machine learning models as learned oracles will enhance the accuracy-memory tradeoff without the extensive training data and memory overhead associated with more complex models like RNNs.

4. Experimental Settings

In this project, I explore the space-accuracy tradeoff of various enhanced versions of the Count-Min Sketch algorithm, applying them to the AOL search query dataset (link). The objective is to estimate the frequencies of different search queries.

Different versions of the Count-Min Sketch algorithm are created by augmenting the Count-Min Sketch algorithm with different types of oracles. The versions of the Count-Min Sketch algorithm under examination include:

1. T**raditional Count-Min Sketch:** This serves as the baseline for comparison.

2. **CMS with Hypothetical Ideal Oracle:** This version assumes an ideal oracle which perfectly identifies heavy hitters in the test data, serving as a theoretical best-case scenario for comparison.
3. **CMS with Logistic Regression Oracle:** This version uses a logistic regression model as the oracle for learning heavy hitters in the data.
4. **CMS with SVM Oracle:** Here, an SVM (Support Vector Machine) model is employed as the oracle.
5. **CMS with Decision Tree Oracle:** This variant uses a decision tree for its oracle.
6. **CMS with Perceptron Oracle:** A perceptron is used as the oracle in this version, utilizing neural networks without using up much memory.

For each of these variants, we use 4 hash functions for the count-min sketch and vary the number of buckets from 2^{17} to 2^{21} (to vary memory usage).

We use the first 5 days of data from the AOL dataset. We insert all queries from the first 3 days into the CMS variant. For each variant, we insert each element of the dataset into the count-min sketch, recording the memory usage of the count-min sketch. In variants with ML models, the memory usage of the model and the vectorizer (after training is completed) is also included in the total memory usage.

We use the fourth day from the AOL dataset for training the ML models (in variants 3-6) and the 5th day for validation and hyper-parameter tuning of the ML models. The ML models and also require a preprocessing step where a TF-IDF vector is trained on the training data and each string is converted into a TF-IDF vector before being input to the model.

In particular, the hyperparameters that needed to be tuned for each model were as follows:

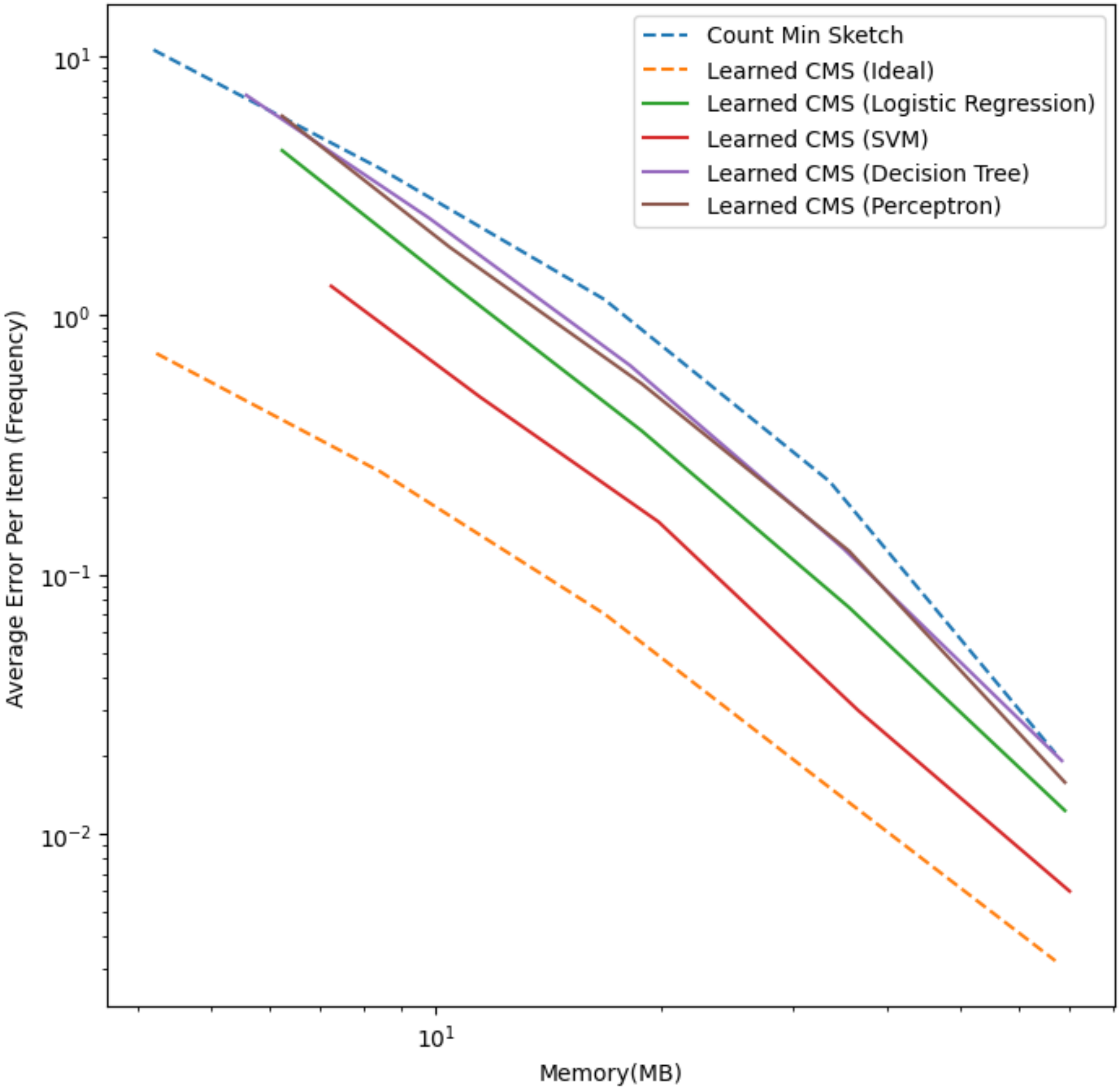
- Logistic Regression: Decision Threshold Probability
- SVM: C
- Decision Tree: Maximum depth

These were tuned by optimizing the F1 score over the 5th day's data. When choosing between hyperparameters with similar F1 scores, the models chosen were ones that were neither too conservative nor too aggressive with predicting heavy hitters, as this adversely affects the space-accuracy tradeoff of the count-min sketch (the ultimate outcome).

As stated above, the primary metric for evaluation is the space-accuracy tradeoff. Once the ML models are trained and the count-min sketches are filled, we take a subset of common elements and random elements from the dataset (first 3 days) and query them in each count-min sketch variant, recording the estimation error for each query and ultimately computing the mean estimation error for the count-min sketch variant. We then assess the space-accuracy tradeoff by plotting and comparing the mean estimation error against the memory usage for each Count-Min Sketch variant.

5. Results

Carrying out the experimental procedure above gives the following result:



The **CMS with Hypothetical Ideal Oracle** serves as a benchmark, showing the theoretical best-case scenario for the tradeoff between memory usage and accuracy. The **CMS with SVM Oracle** demonstrates a substantial improvement over traditional CMS, approaching the performance of the ideal oracle. This suggests that SVMs are capable of effectively identifying heavy hitters in the data stream. The **CMS with Logistic Regression Oracle** also shows an enhanced tradeoff, although not to the same extent as the SVM. This may reflect the logistic regression's capacity for linear decision boundaries in the frequency estimation task. **CMS variants with Decision Tree and Perceptron Oracles** exhibit the least favorable tradeoff. The decision tree may be overfitting, while the perceptron, due to its simplicity, might not capture the data complexity adequately. All machine learning-enhanced CMS variants outperform the baseline CMS, validating the concept of a learned oracle in this context.

6. Conclusion

The experimental results substantiate my hypothesis that lightweight machine learning models can effectively improve the accuracy-memory tradeoff of Count-Min Sketch. The data demonstrates that even with reduced memory requirements and less training data, these models can outperform the traditional CMS, validating the potential of machine learning in optimizing frequency estimation algorithms.