A Critical Analysis of Recursive Model Indexes

Marcel Maltry Jens Dittrich

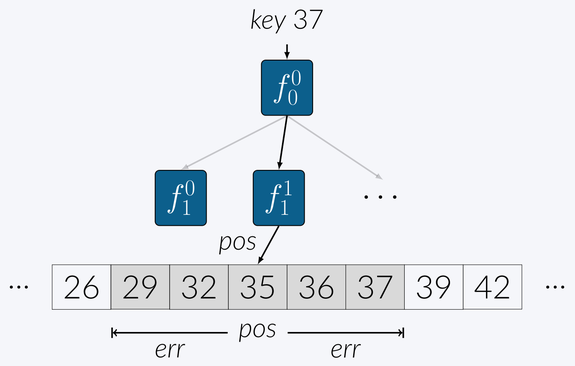
Big Data Analytics Group, Saarland University, Saarland Informatics Campus

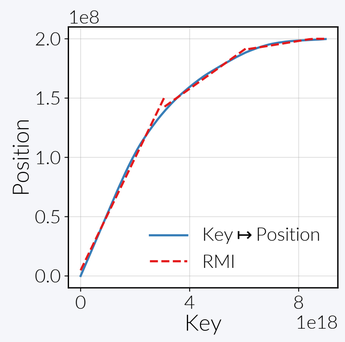
Introduction

The *recursive model index (RMI)* [1] was introduced by Kraska et al. as a machine-learned index over sorted data that accomplishes remarkably fast lookups compared to other state-of-the-art indexes.

Core Idea of RMIs

- View index as a function that maps keys to their position in the sorted data.
- Approximate this function using a hierarchical machine-learning model.
- Correct predictive errors by performing a search around the predicted position.





We tried to understand the superior performance and reproduce the results.

Reproducibility Issues

- At the time of publication, source code was not made available.
- Configurations for large hyperparameter space were not reported.
- Hyperparameter configurations determined by time-consuming enumeration [2].

Contributions

- Provide extensible open-source implementation of RMIs.
- Conduct hyperparameter analysis with the goal of reproducing results.
- Develop guideline for configuring hyperparameters in practice.

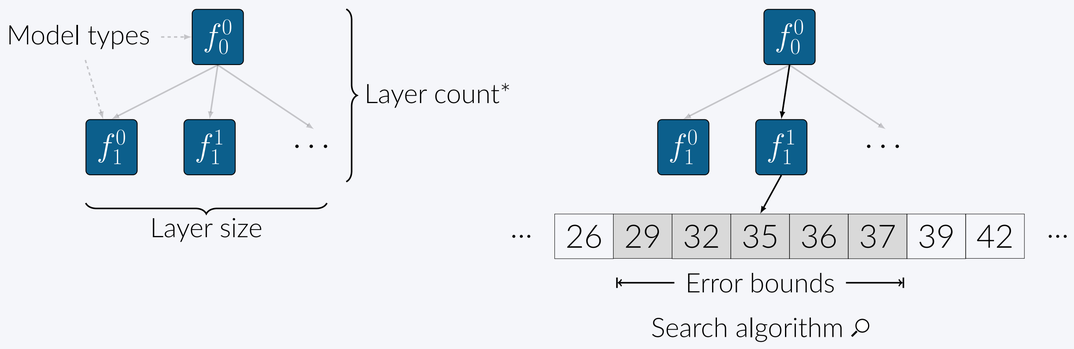
Experimental Setup

We conduct a series of experiments to evaluate RMIs in terms of segmentation capabilities, predictive accuracy, evaluation time, lookup time, and build time.

Hyperparameters

We evaluate 1280 configurations obtained by varying the hyperparameters below.

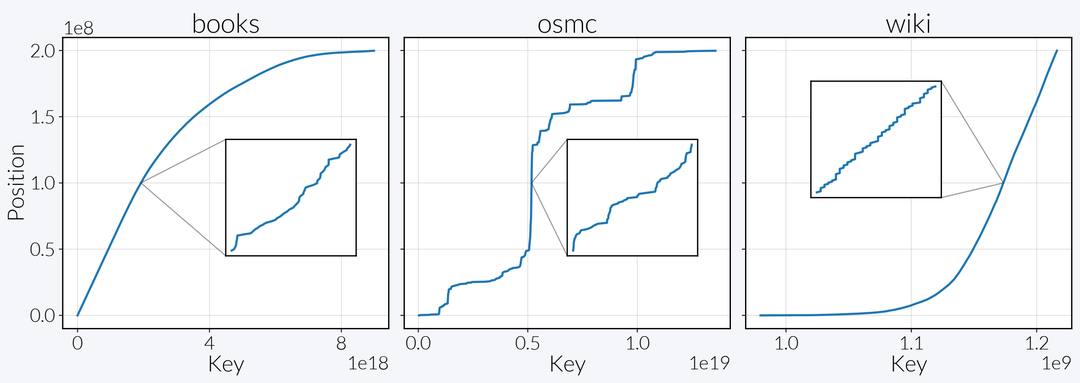
Models types	Linear Regression, Linear Spline, Cubic Spline, Radix
Layer sizes	between 2^6 and 2^{25}
Error bounds	Local Bounds, Global Bounds (2 variants each), No Bounds
Search algorithms	Linear Search, Binary Search (2 variants), Exponential Search



We refer to an RMI that uses Linear Spline and Linear Regression as first and second-layer model type, respectively, as Linear Spline↦Linear Regression.

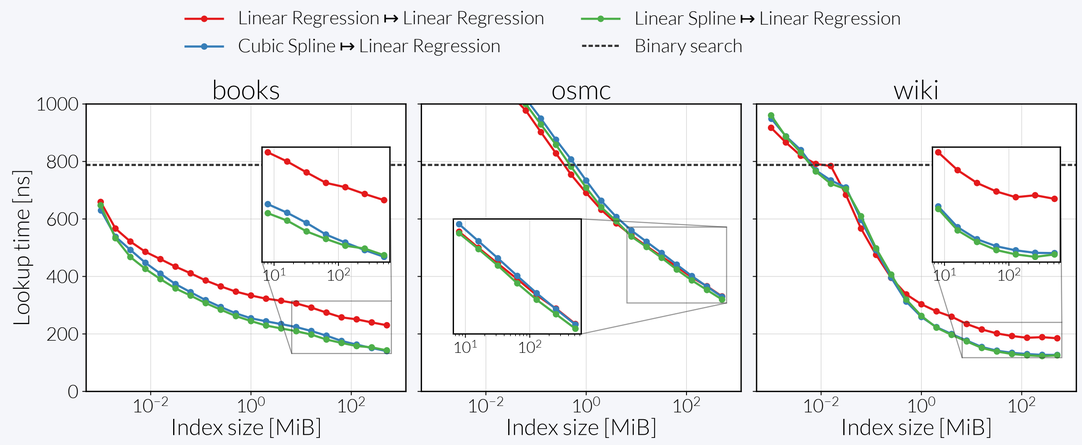
Datasets

Each configuration is evaluated on four real-world datasets from *Search On Sorted Data (SOSD)* [3], three of which are shown below.



Model Types

We compare lookup times of different combinations of model types, here using No Bounds + Exponential Search for error correction.

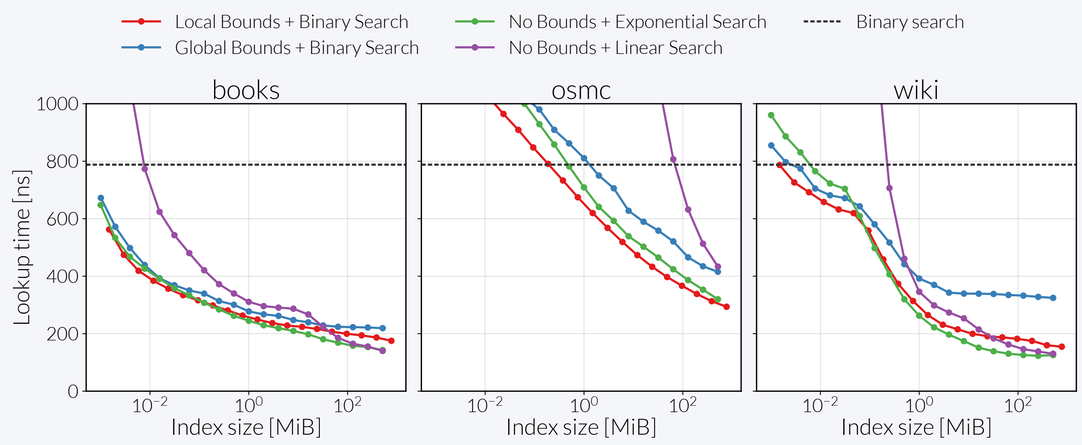


Result

- Linear/Cubic Spline↦Linear Regression usually perform best.

Error Correction

We compare lookup times of different combinations of error bounds and search algorithms, here using Linear Spline↦Linear Regression as model types.



Result

- Local Bounds + Binary Search generally achieves fast lookup times.
- No Bounds + Exponential/Binary Search is faster in case of low predictive error.

Guideline

We develop a compact guideline for configuring RMIs in practice.

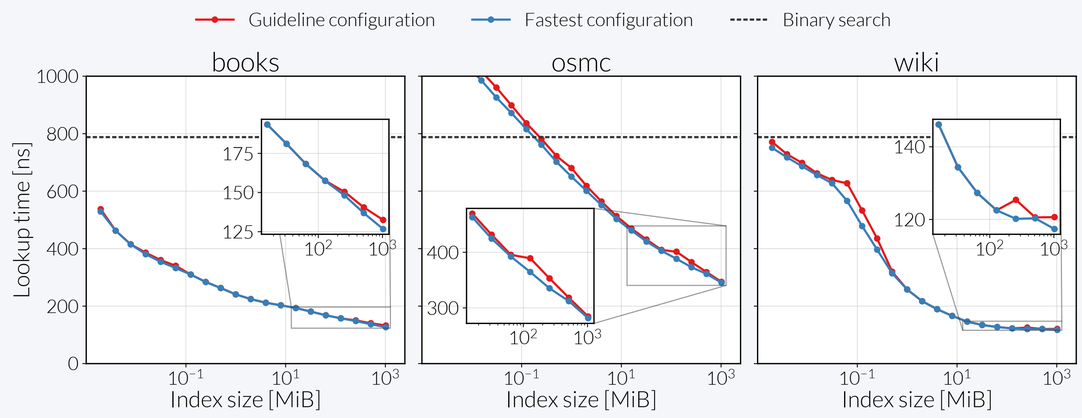
Model Types and Layer Size

- Linear Spline↦Linear Regression
- Pick largest layer size that achieves desired index size.

Error Bounds and Search Algorithm

- No Bounds + Exponential Search in case of low predictive error,
- Local Bounds + Binary Search otherwise.

We compare lookup times of our guideline with the fastest configurations.



Result

- Time-consuming enumeration not needed to determine hyperparameters.
- Average performance decline of 2%, maximum performance decline of 11%.

[1] Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. The case for learned index structures. In *Proceedings of SIGMOD 2018*, pages 489–504. ACM, 2018.

[2] Ryan Marcus, Emily Zhang, and Tim Kraska. CDFShop: exploring and optimizing learned index structures. In *Proceedings of SIGMOD 2020*, pages 2789–2792. ACM, 2020.

[3] Andreas Kipf, Ryan Marcus, Alexander van Renen, Mihail Stoian, Alfons Kemper, Tim Kraska, and Thomas Neumann. SOSD: A benchmark for learned indexes. *NeurIPS Workshop on Machine Learning for Systems*, 2019.

published in
Proc. VLDB Endow. 15(5)

Workshop on Reproducible Science 2022
TU Dresden

Paper Code