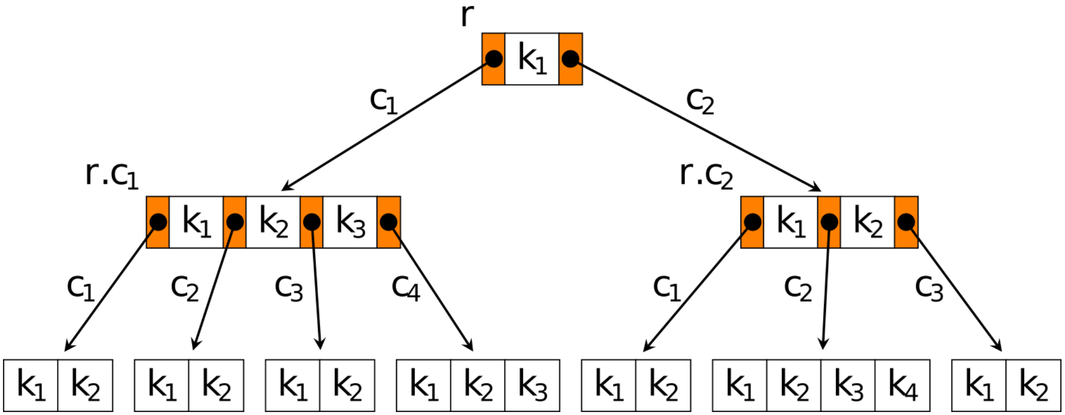
A Tailored Regression for Learned Indexes: Logarithmic Error Regression

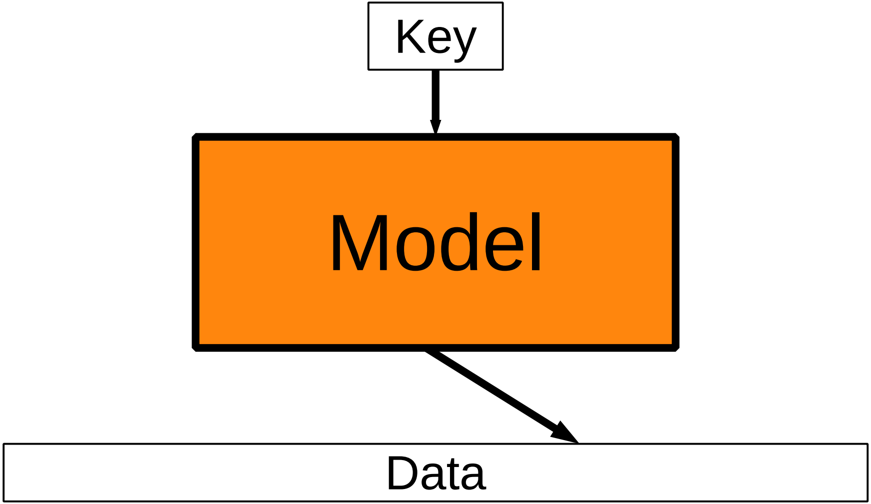
Martin Eppert   Philipp Fent   Thomas Neumann

Technical University of Munich

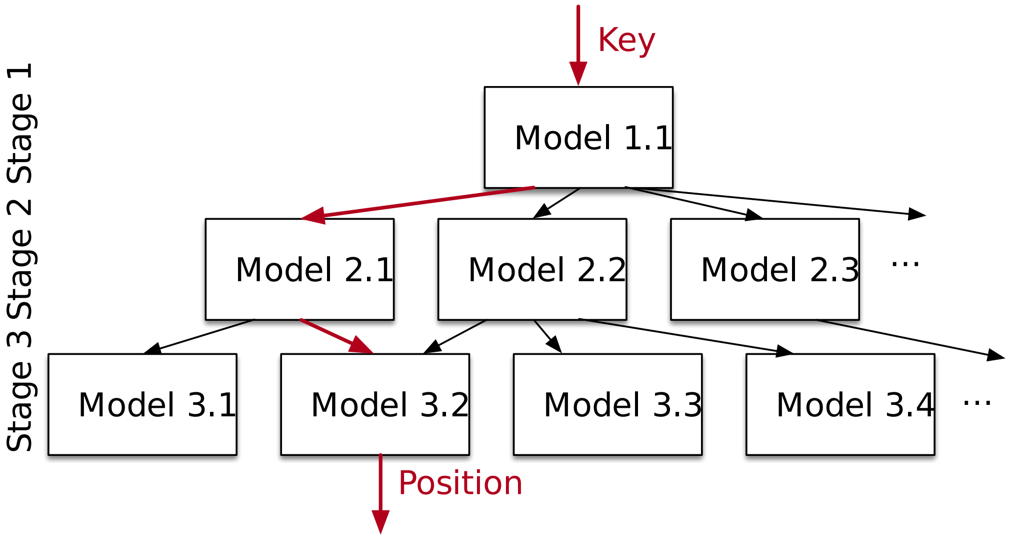
Traditional Index Structures



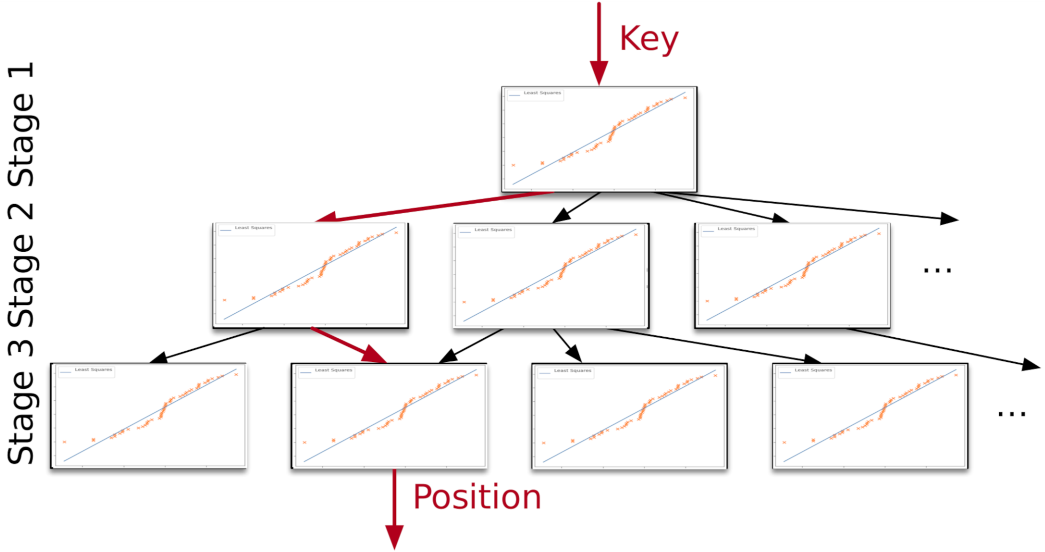
Index Structures



Learned Index Structures



Learned Index Structures



Which Regression Model to Use?

Lasso
Robust
Theil-Sen
Non-Linear
Bayesian Linear
Simple Linear Regression
Least Absolute Deviation
Piecewise Linear
Least Squares
Polynomial
Logistic

Simple Linear Regression

- Optimizes Squared Error
- Linear runtime

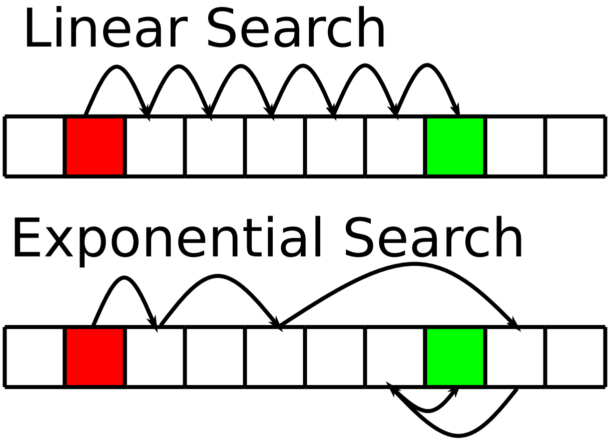
$$\hat{a} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \qquad \hat{b} = \bar{y} - (\hat{a}\bar{x})$$

Evaluation of Simple Linear Regression

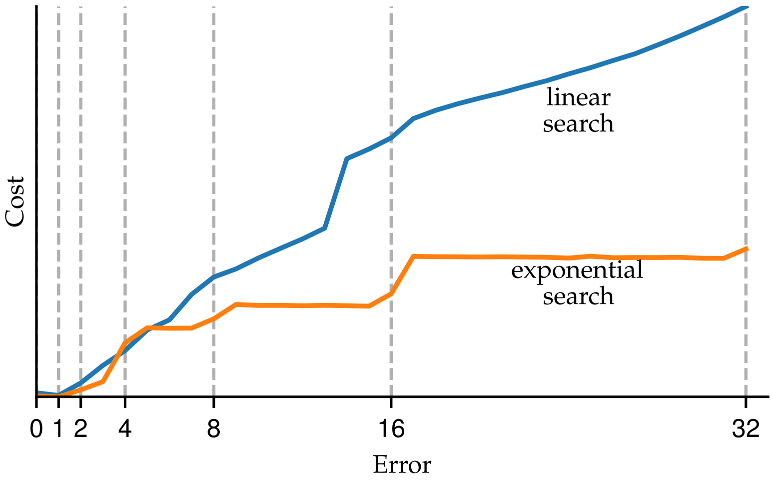
- What metric do we want to optimize?

Lookup Time

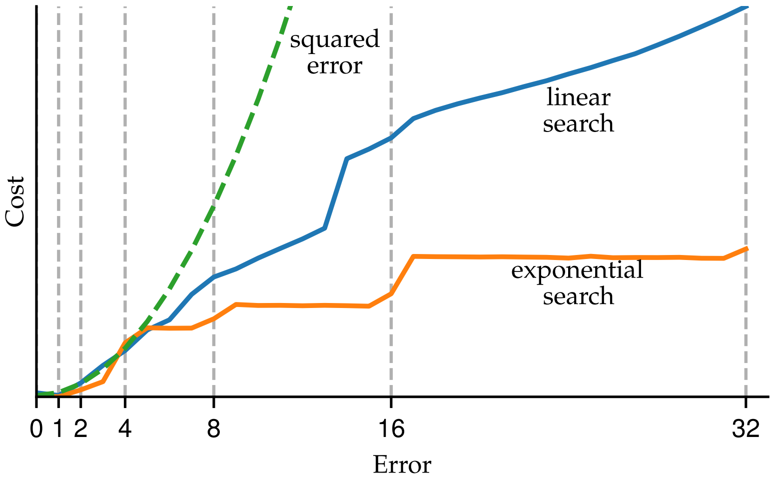
Search Methods



Search Methods



Search Methods



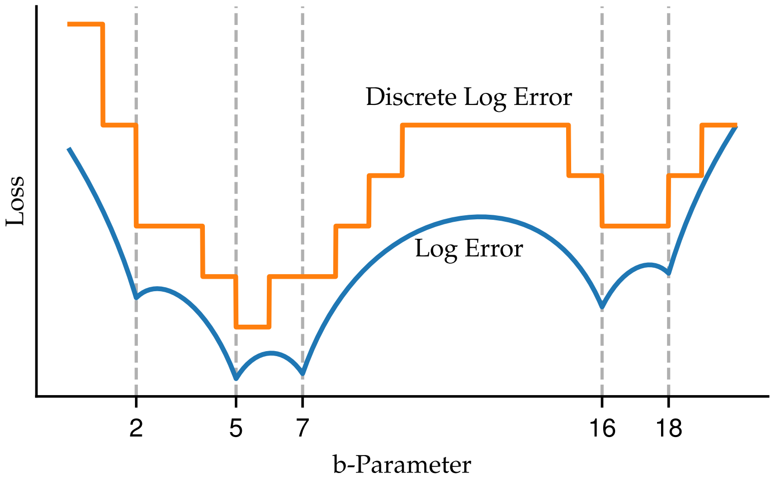
Logarithmic Error

$$L(\epsilon) = \lceil log_2(1 + \epsilon) \rceil$$

Is Optimization feasible?

$$L(\epsilon) = \sum_{i=0}^{N} \log_2(1+\epsilon) = \log_2\left(\prod_{i=0}^{N}(1+\epsilon)\right)$$

Is Optimization feasible?



Brute Force

- The regression must intersect 2 datapoints

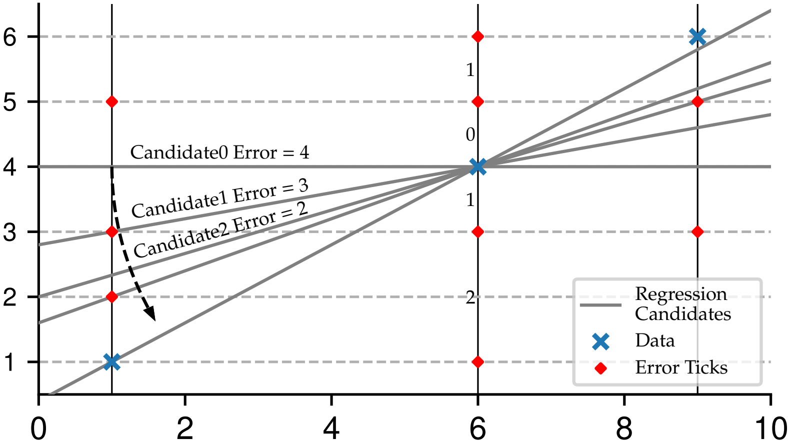
- Test all combinations
  - $O(n^2)$ combinations
  - $O(n)$ time to calculate the error
- $O(n^3)$ time to calculate

Improving The Brute Force Algorithm

- $\lceil log_2(1+\epsilon)\rceil$ is a discrete function
- takes on at most $O(log(\epsilon_{max}))$ discrete error values

- Idea: Find optimal regression intersecting a single datapoint

Improving The Brute Force Algorithm



Do we even need exact solutions?

- NO
- Least Squares Regression is already far off
- Not guaranteed to be optimal due to hardware specifics

Two Point Method

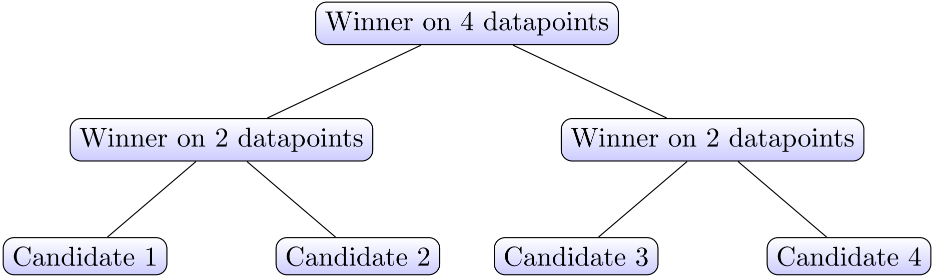
- Idea: create a gradient descent method finding appropriate local minima

- Obtain the optimal regression for a random datapoint
- Use the second crossed datapoint as the new pivot
- Repeat until convergence

Tournament Evaluation

- Idea: Not many samples are needed to determine a bad regression

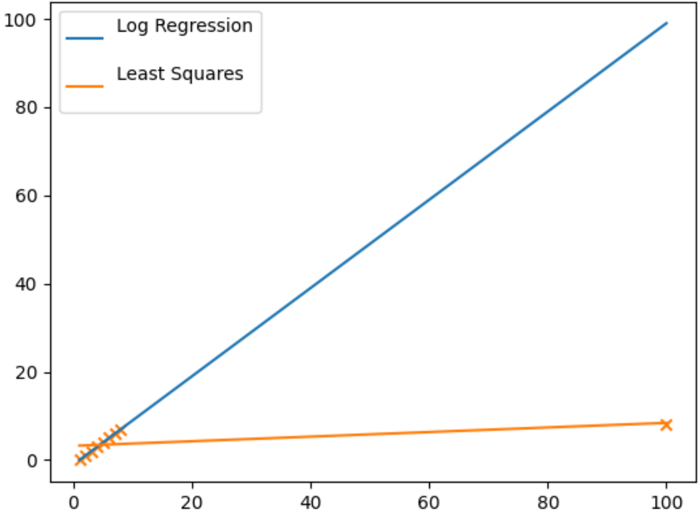
- Pick *n* candidates and let them compete in a tournament
- evaluate on 2^{height} datapoints



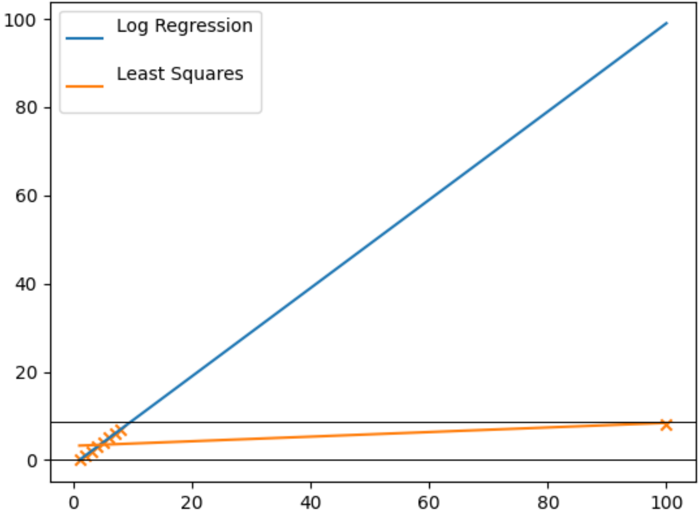
MicroBenchmark

		Speedup	
Method	Data set	Mean	Median
Tournament Evaluation Log	Facebook	6.9 %	5.6 %
Tournament Evaluation DLog		6.7 %	4.9 %
Two Point Method		5.4 %	4.3 %
Tournament Evaluation Log	Wiki	3.9 %	7.2 %
Tournament Evaluation DLog		3.9 %	6.9 %
Two Point Method		4.7 %	7.0 %
Tournament Evaluation Log	Normal	15.2 %	14.7 %
Tournament Evaluation DLog		15.1 %	14.1 %
Two Point Method		14.6 %	13.9 %
Tournament Evaluation Log	Outlier	121.6 %	120.5 %
Tournament Evaluation DLog		120.5 %	121.4 %
Two Point Method		122.8 %	121.9 %
Tournament Evaluation Log	Uniform	1.6 %	3.3 %
Tournament Evaluation DLog		1.4 %	2.8 %
Two Point Method		1.2 %	3.7 %

Robustness



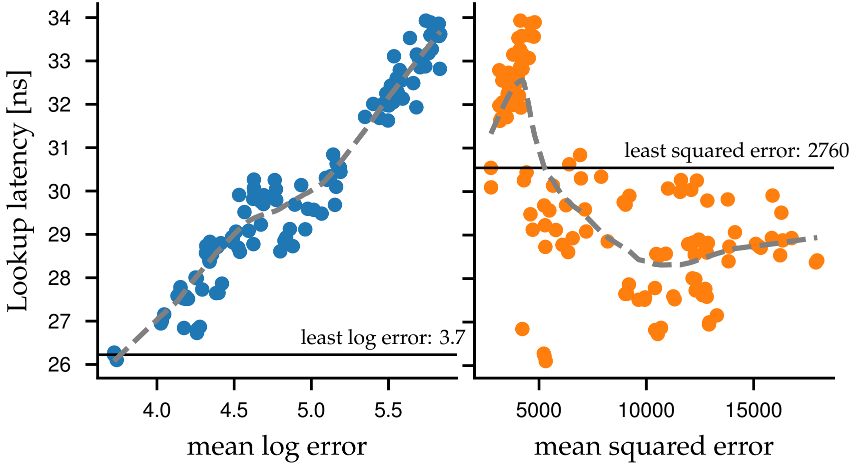
Bounds Checking



Benchmark on the RMI

Data set	1^{st} Layer	Lookup [ns] SLR	Lookup [ns] Log	Lookup [ns] Speedup	Build [s] SLR	Build [s] Log	Build [s] Slowdown
Facebook	linear	245.7	244.1	0.7 %	23	65	2.8×
Wiki	linear	172.1	160.6	7.2 %	24	67	2.8×
Osm	cubic	399.4	383.4	4.2 %	27	122	4.5×
Gaussian Mixture	linear	296.7	266.0	11.5 %	21	47	2.2×

Lessons Learned



Conclusion

- Standard regression is suboptimal
- Proposed methods fit underlying search method
- Logarithmic error is an imporant measure
- Show improvement on the Recursive Model Index

Future Work

- Build time optimized regression
- Detect if the logarithmic error regression is needed
- Integrate cache lines into the error function