A “learned” approach to quicken and compress rank/select dictionary

[ALENEX21]

~

PRIN Meeting 12/03/2021

Antonio Boffa, Paolo Ferragina, Giorgio Vinciguerra

A³Lab
acube.di.unipi.it
Rank/Select dictionary

- Given a set $S$ of $n$ integers drawn from a universe of size $u$
  - Store them in compressed form
  - Implement $\text{rank}(x)$: number of elements in $S$ which are $\leq x$
  - Implement $\text{select}(i)$: the $i$th smallest element in $S$

- Building block of succinct data structures for texts, genomes, graphs, hash tables, etc.

```
Select(2) 3 6 10 15 18 22 40 43 47 53

Rank(21) 5
```
Patterns

- New applications produce data with inherent patterns and trends (IoT, I4.0, etc.)
- It is inefficient to design a system for every specific pattern/data distribution
- Machine Learning techniques automatically discover and exploit patterns
Learned Data Structures

- Unexpected combination of Machine Learning and Data Structures
- Learned Indexes are achieving significant results in practice
- Some preliminary results are appearing in theory too [Ferragina et al., ICML 2020]

Figure 1: Why B-Trees are models
Learned Data Structures

- Which ML model?
  - Trade offs between model complexity and its performance
  - Deep Neural Networks?
  - Linear Regression?

- Piecewise Linear Approximation (PLA)
  - Effective compromise [Ferragina et al., VLDB 2020]
  - Pairs \((i, S[i])\)
  - Map the pairs in a Cartesian plane
  - Choose maximum error \(\varepsilon\)
Our Proposal: the LA-vector

• Combination of
  • Segments \((s_1, s_2)\)
  • Vector of corrections \((C)\)

• Compression scheme
  • \(S[p] = \text{slope} \cdot p + \text{intercept} + C[p]\)
Complexity analysis

- Segments as efficient representations of sequences of integers with an information loss of $\varepsilon$

- Space occupancy = $b\ell + cn$ bits, where
  - $b =$ space for a segment = $\log n + \log u + w$
  - $\ell =$ #segments, that is the model complexity
  - $c = \log(2\varepsilon + 1)$

- Select time = $O(1)$
- Rank time = $O(\log \ell + c)$

... so everything depends on the number of segments?
Complexity analysis

In turn the number of segments depends on:
- The size of the input dataset
- How the points \((pos, key)\) map to the plane
- The value \(\varepsilon\), i.e. how much the approximation is precise
Suppose that the gaps between the sorted integers are a realisation of a random process with finite mean and variance.

Then the expected number of keys covered by a segment with maximum error $\varepsilon$ is

$$\Theta(\varepsilon^2)$$

and the segments on $n$ keys are, whp,

$$\Theta\left(\frac{n}{\varepsilon^2}\right)$$

Practically the #segments is order of magnitudes smaller than $n$ [Ferragina et al., VLDB 2020]
Theoretical comparison against Elias-Fano

- LA-vector uses less space than EF if
  \[ \ell = O \left( \frac{n}{\log n} \right) \]

- From the previous theoretical results this holds for
  \[ \varepsilon = \Omega \left( \sqrt{\log n} \right) \]
Space optimization

- One epsilon for all the dataset could waste space

- Our idea to optimize space:
  - Partition the dataset according to its regularities
  - Use a different $\epsilon$ for each partition

- Reduction to the shortest path problem on *ad hoc* graphs

- We propose a greedy approximation algorithm
  - Taking $O(n \log u)$ time and $O(n)$ space
  - Losing only a *constant factor* of bits wrt the minimum sized LA-vector
Experiments

- Our solution (varying $\epsilon$)
- Our space-optimized solution
- sdsl::rrr_vector (varying block size)
- sdsl::sd_vector (Elias-Fano)
- sdsl::enc_vector (Gap-encoding + Elias $\gamma/\delta$-code)
- ds2i::partitioned_EF_uniform
- ds2i::partitioned_EF_optimal
Conclusions

- First **learned** and **compressed** data structure for rank/select
- Proved theoretical results which compare favourably to Elias-Fano
- Experimentally
  - New interesting space-time trade-offs
  - Our Select is the fastest
  - Our Rank is on the Pareto curve
- **Take home message:**
  - LA-vector is a novel tool for building efficient rank/select data structures
  - Two ingredients: linear $\varepsilon$-approximation and fixed-len integer compression (vector $C$)
- Preliminary research, it opens several interesting new lines of research