FASTER UPPER BOUNDING OF INTERSECTION SIZES

IBM Research Tokyo - Japan
Daisuke Takuma
Hiroki Yanagisawa
Outline

• Motivation of upper bounding
• Contributions
  – Data structure
    • Proposal
    • Evaluation *(introduce related work here)*
  – Application
    • Evaluation
• Conclusion
Set intersection

Intersection: $A \cap B = \{2, 5, 7\}$
Intersection size: $|A \cap B| = 3$

Applications:
• Multi-word search
• JOIN operation:

Recently the intersection operation is used in a different way.
Some apps compute intersections only for their sizes.

Top-k term query
Input: the hit docs for a search query
Output: the k most frequent terms in the hit docs

Determined by the intersection sizes
Intersection sizes are evaluated in inequalities:
\[ |A \cap B| > \text{threshold}, \text{ but mostly } |A \cap B| \ll \text{threshold}. \]

\[ |A \cap B| < 0.05 \times \text{threshold} \]

In 80% intersections in top-k term queries:
\[ |A \cap B| \leq \text{threshold} \]

\[ \text{Upper bound of } |A \cap B| \]

\( \Rightarrow \) If this holds, we can skip the computation of the exact intersection.
Our contributions:

- We proposed data structures *Cardinality Filters* (CF) to quickly compute upper bounds of intersection sizes.
- We accelerated the top-k term query using CFs approximately by factor of 2.
- CFs are applicable to all the algorithm evaluating $|A \cap B| \geq$ threshold by adding the condition for the upper bound.

$$\text{upperbound}(|A \cap B|) \geq \text{threshold AND}$$

$|A \cap B| \geq \text{threshold}$

It does not change the logic.
Cardinality Filters
Requirements for CFs

Sets $A, B \in 2^X$

- $A \cap B$ (set intersection)
- Set size function: $|\cdot|$
- $|A \cap B| \leq |\Phi(A) \cap \Phi(B)|$
- CF size function: $|\cdot|$
- $\Phi(A) \cap \Phi(B)$ (CF intersection)

Cardinality filters (CFs) $\Phi(A), \Phi(B) \in \Phi(2^X)$
SCF: Single Cardinality Filter

\[ \Phi(A) = (h(A), c(A)) \text{ where} \]
\[ h(A) \text{ hash values, } c(A): \text{ non-smallest values in collisions} \]

\[ |\phi(A) \cap \phi(B)| = |h(A) \cap h(B)| = 3 + |c(A) \cap c(B)| = 2 \]

Bit-wise AND

Small
Formal definition of SCF

Let $X$ be a universe set $(A, B \subseteq X)$, and $H$ be a hash value space $H=\{0, 1, \ldots, \lfloor |X|/N \rfloor - 1\}$, we define SCF $\Phi: 2^X \rightarrow (2^H, 2^X)$ by

$$\Phi(A) = (h(A), c(A)) \ (A \subseteq X)$$

where

$$h(A) = \{ h(x) \mid x \in A \}$$
$$c(A) = \{ y \in A \mid (\exists x \in A) \ (h(x) = h(y), x < y) \}$$

Thm:

$$|A \cap B| \leq |\Phi(A) \cap \Phi(B)| := |h(A) \cap h(B)| + |c(A) \cap c(B)|$$

- Directly used in the proof.
- The proof has no conditional branches.
RCF: Recursive Cardinality Filter

The RCF recursively applies the SCF to c(A).

\[ |\Phi(A) \cap \Phi(B)| = \sum_{1 \sim L} |H_i(A) \cap H_i(B)| + |C_L(A) \cap C_L(B)| \]
We compared CFs with the following algorithms:

- **Exact intersections**
  - Linear merge: comparison between sorted lists.
  - Binary search: binary search \( x \in A \) in \( B \) \((|A| < |B|)\).
  - **DK algorithm (VLDB 2011):**
    - Intersect \( h(A) \) and \( h(B) \) by bit-wise AND (\( h: \text{hash} \))
    - For all \( x \in h(A) \cap h(B) \),
      intersect \( h^{-1}(x) \cap A \) and \( h^{-1}(x) \cap B \).

- **Naïve upper bounding**
  - Bloom filter: Test \( x \in A \) in the Bloom filter of \( B: \) \( h(B) \).
Performance studies:

(A) Large (1M x 1M) - no correlation

(B) Middle (100K x 100K) - no correlation

(C) Small (10K x 10K) - no correlation

(D) Asymmetric (1M x 10K) - no correlation

(E) Middle (100K x 100K) - positive correlation

(F) Middle (100K x 100K) - negative correlation

Size

Asymmetry

Correlation

traditional  DK naïve  ours
Time complexity and performance break down

DK: $O\left( \frac{|A|+|B|}{\sqrt{w}} + |A \cap B| \right)$
RCF: $O\left( \frac{|A|+|B|}{w} \right)$

(With a practical setting of $h$)

$|A \cap B|$ is the barrier for all the exact algorithms.

CFs maximize the utilization of bit-wise AND.

The benefit of considering the upper bounds.
Improvement of the top-k term query
The intersection sizes of CFs are evaluated before evaluation of the exact intersection sizes.

Intersect the CFs using the same hash functions.
Improvement of top-k term queries

The number of hit documents vs. Computation time [millisec]

- LM
- LM+BS
- DK1H
- SCF
- RCF

Ours
More than 80% of the exact intersections were skipped.
Summary

- We proposed *Cardinality Filters* (SCF, RCF) to quickly compute upper bounds of intersection sizes.
  - Property: \(|A \cap B| \leq |\Phi(A) \cap \Phi(B)|\)
  - Performance improvement from the exact algorithms.
    - DK: \(O\left(\frac{|A|+|B|}{\sqrt{w}} + |A \cap B|\right)\)
    - RCF: \(O\left(\frac{|A|+|B|}{w}\right)\)

- We accelerated the top-k term query using the CFs by factor of 2.
Future work

- Using other algebraic properties of SCF and RCF:
  - Monotonicity: \( A \subset B \Rightarrow |\Phi(A)| \leq |\Phi(B)| \)
  - Commutative and associative laws of \( \Phi(A) \cap \Phi(B) \)

- Implementation of CFs having Union-like operation

- Non-parametric CFs (compression ratio of \( h \))
Faster Upper Bounding of Intersection Sizes

Thank you !!