CS6931 Database Seminar

Lecture 6: Set Operations on Massive Data

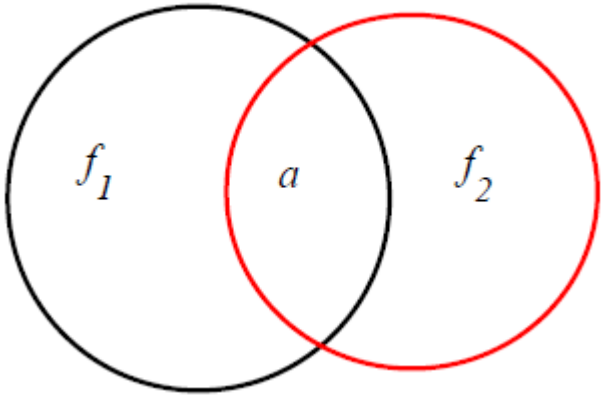
Set Resemblance

and

MinWise Hashing/Independent Permutation

Basics

- Consider two sets S1, S2 $\subseteq$ U = {0, 1, 2, ...,D − 1} (e.g., D = 2^{64})



- f1 = |S1|, f2 = |S2|, a = |S1 ∩ S2|.
- The **resemblance** R is a popular measure of set similarity

$$R = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = \frac{a}{f_1 + f_2 - a}$$

Basics and Applications

- A set $S \subseteq U = \{0, 1, ..., D . 1\}$ can be viewed as 0/1 vector in D dimensions.
- Application—Shingling: Each document (Web page) can be viewed as a set of w-shingles. For example, after parsing, a sentence "welcome to school of computing" becomes
  - w = 1: {welcome, to, school, of, computing}
  - w = 2: {welcome to, to school, school of, of computing}
  - W = 3: {welcome to school, to school of, school of computing}

  Shingling generates extremely high dimensional vectors, e.g., $D = (10^5)^w$.

Near Duplicate Detection

- Each document (Web page) is parsed into a set of w-shingles. Suppose there are roughly 10^{10} English Web pages, i.e., $N = 10^{10}$ sets.

- **Q:** Can we efficiently find the duplicate documents (e.g., paper plagiarism), fast and using only affordable memory space?

- **A:** If Resemblance is the similarity measure, then minwise hashing may provide a highly practical algorithm for duplicate detection.

MinWise Hashing/Independent Permutation

- Suppose a random permutation π is performed on U, i.e.,
  - π: U → U, where U={0,1,…., D-1}.
  - A simple argument can show that, where S1 ⊆ U and S2 ⊆ U

    Pr (min(π(S1)) = min(π(S2)))=$\frac{|S1 \cap S2|}{|S1 \cup S2|} = R$

    Why?

To Reduce Variance

- After k independent permutations, π_1, π_2, ..., π_k, one can estimate R without bias, as a binomial:
  - $\hat{R} = \frac{1}{k}\sum_{j=1}^{k} 1\{\min(\pi_j(S1)) = \min(\pi_j(S2))\}$

    why?
  - $\mathrm{Var}(\hat{R}) = \frac{1}{k} R(1-R)$

    why?
- Because $a = \frac{R}{1+R}(f_1 + f_2)$, we can estimate a from the estimated R!

How to implement minwise independent permuation and other set operations

- There is a stoc 98 paper
- Or in practice, you can simulate minwise independent permuation using k-wise indepdent hash function: usually works pretty well.
- MinWise synopsis:
  - $S(A)=\{\min(\pi_1(A), \pi_2(A), ..., \pi_k(A)\}$
  - How to compute $S(A \cup B)$?
  - Given $S(A)$, $S(B)$, and $|A|$, $|B|$, how to estimate $|A \cup B|$ and $|A \cap B|$?

Storage Problem of Minwise Hashing

- Each hashed value, e.g., min(π(S1)), is usually stored using 64 bits
- For typical applications, k=50 or 100 is required (hashed values)
- The total storage can be prohibitive:
  - Nx64x100=8TB, if $N=10^{10}$!
- Storage problems also cause computational problems, of course.

b-Bit MinWise Hashing

- Basic idea: Only store the lowest b-bits of each hashed value, for small b.
- Intuition why this works:
  - When two sets are identical, then their lowest b-bits of the hashed values are of course also equal.
  - When two sets are similar, then their lowest b-bits of the hashed values "should be" also similar (True?).
  - Therefore, hopefully, we do not need many bits to obtain useful information, especially considering real applications often care about pairs with reasonably large resemblance values (e.g., 0.5).
- For more details on why this works, refer to the resource on the web.

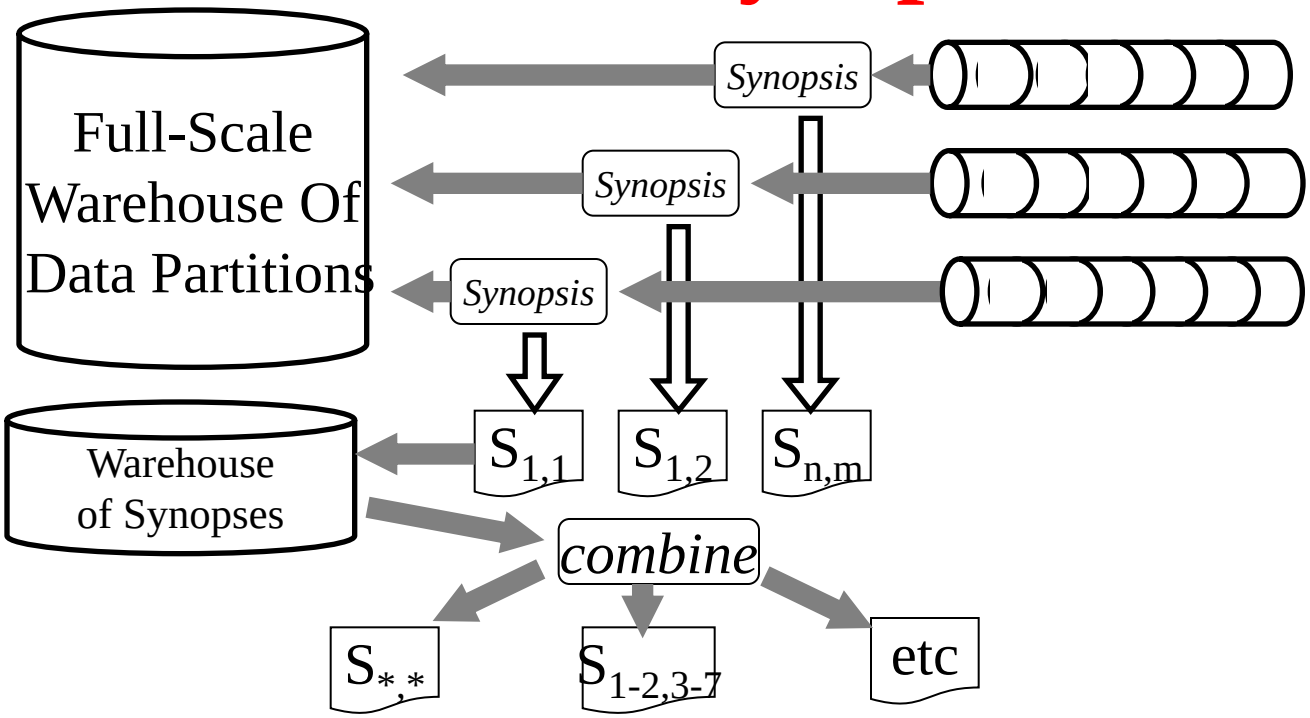
General set operations and Distributed sets

Kevin Beyer
Peter J. Haas
Berthold Reinwald
Yannis Sismanis
Rainer Gemulla

Introduction

- Estimating # Distinct Values (DV) crucial for:
  - Data integration & cleaning
    - E.g. schema discovery, duplicate detection
  - Query optimization
  - Network monitoring
  - Materialized view selection for datacubes
- Exact DV computation is impractical
  - Sort/scan/count or hash table
  - Problem: bad scalability
- Approximate DV “synopses”
  - 25 year old literature
  - Hashing-based techniques

Motivation: A Synopsis Warehouse

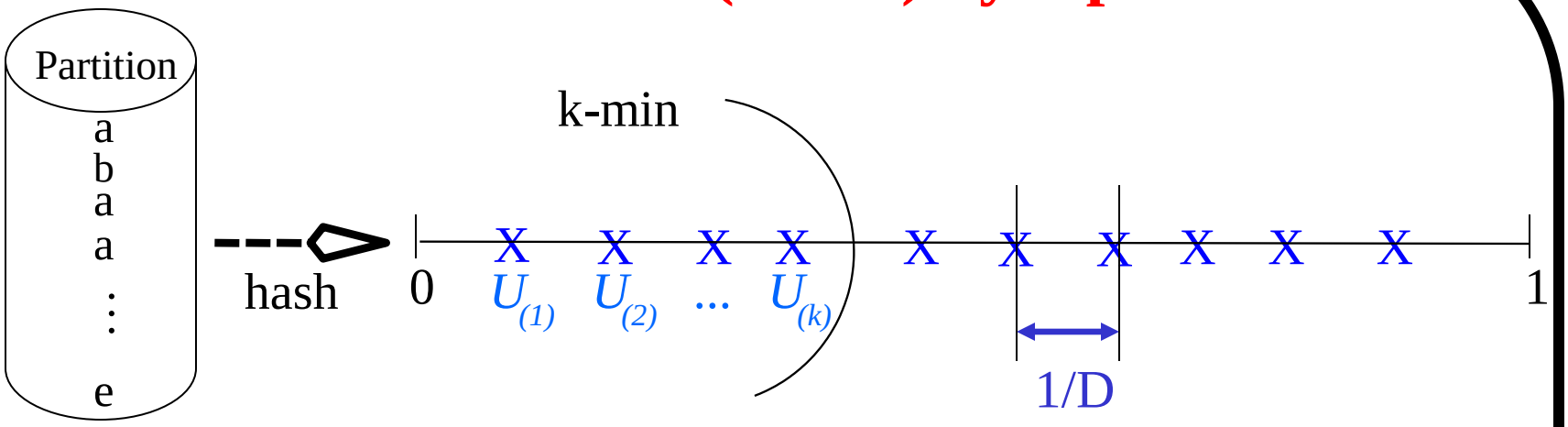


- **Goal**: discover partition characteristics & relationships to other partitions
  - Keys, functional dependencies, similarity metrics (Jaccard)
  - Similar to Bellman [DJMS02]
- **Accuracy challenge**: small synopses sizes, many distinct values

Outline

- Background on KMV synopsis
- An unbiased low-variance DV estimator
  - Optimality
  - Asymptotic error analysis for synopsis sizing
- Compound Partitions
  - Union, intersection, set difference
  - Multiset Difference: AKMV synopses
  - Deletions
- Empirical Evaluation

K-Min Value (KMV) Synopsis



D distinct values

- Hashing = dropping DVs uniformly on [0,1]
- KMV synopsis: $L = \{U_{(1)}, U_{(2)}, ..., U_{(k)}\}$
- Leads naturally to basic estimator [BJK+02]
  - *Basic estimator:* $E[U_{(k)}] = k/D \Rightarrow \hat{D}_k^{BE} = k/U_{(k)}$
  - All classic estimators *approximate* the basic estimator
- Expected construction cost: $O(N + k \log\log D)$
- Space: $O(k \log D)$

Intuition

- Look at spacings
  - Example with $k = 4$ and $D = 7$:

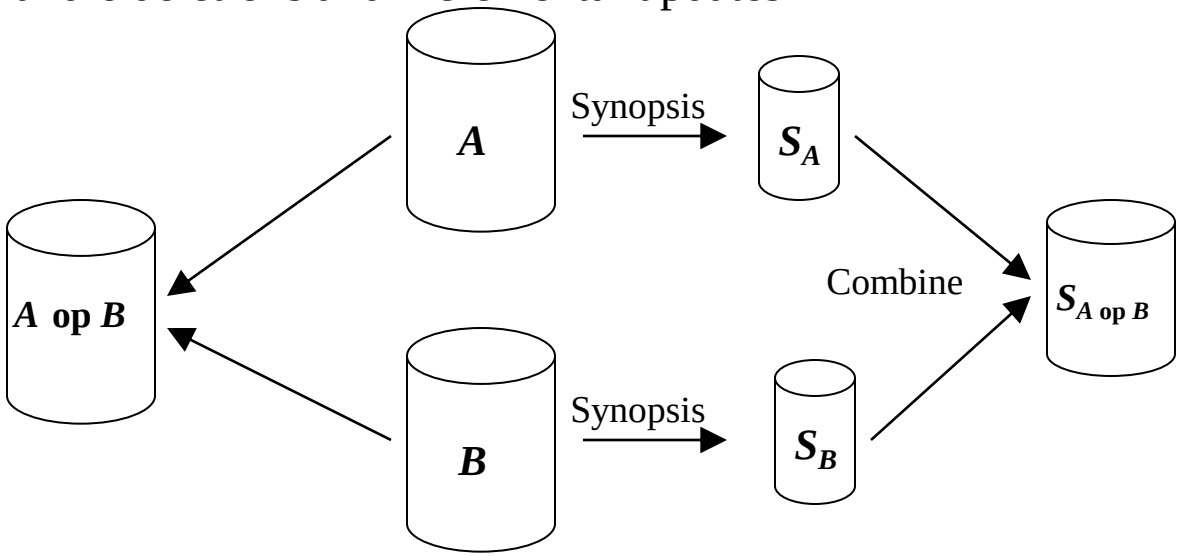
  - $E[V] \approx 1 / D$ so that $D \approx 1 / E[V]$
  - Estimate D as $1 / \text{Avg}(V_1,\ldots,V_k)$
  - I.e., as $k / \text{Sum}(V_1,\ldots,V_k)$
  - I.e., as $k / u_{(k)}$
  - Upward bias (Jensen's inequality) so change k to k-1

    if X is a random variable and φ is a convex function, then

    $\varphi(E(X)) \leq E(\varphi(X))$ (the secant line of a convex function lies *above* the graph of the function)
  -

New Synopses & Estimators

- Better estimators for classic KMV synopses
  - Better accuracy: unbiased, low mean-square error
  - Exact error bounds (in paper)
  - Asymptotic error bounds for sizing the synopses
- Augmented KMV synopsis (AKMV)
  - Permits DV estimates for compound partitions
  - Can handle deletions and incremental updates



Unbiased DV Estimator from KMV Synopsis

- Unbiased Estimator [Cohen97]: $\hat{D}_k^{UB} = (k-1)/U_{(k)}$
  - Exact error analysis based on theory of order statistics
  - Asymptotically optimal as *k* becomes large (MLE theory)

- Analysis with many DVs
  - Theorem:

$$E\left[\frac{\left|\hat{D}_k^{UB} - D\right|}{D}\right] \approx \sqrt{\frac{2}{\pi(k-2)}}$$

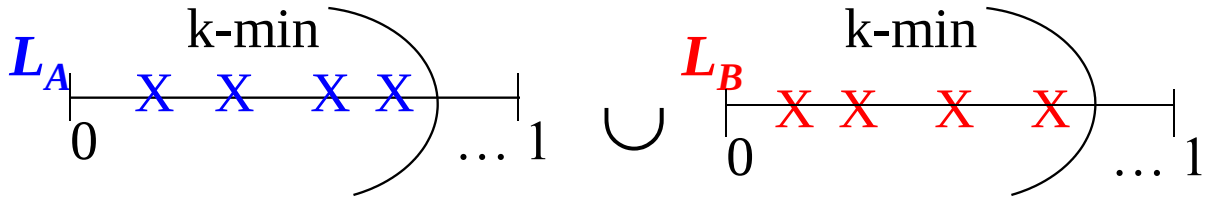
  - Proof:
    * Show that $U_{(i)}$-$U_{(i-1)}$ approx exponential for large D
    * Then use [Cohen97]

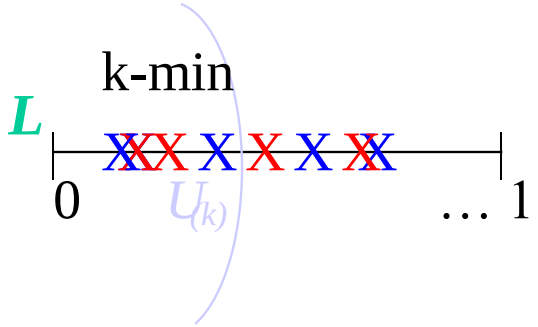
- Use above formula to size synopses a priori

Outline

- Background on KMV synopsis
- An unbiased low-variance DV estimator
  - Optimality
  - Asymptotic error analysis for synopsis sizing
- Compound Partitions
  - Union, intersection, set difference
  - Multiset Difference: AKMV synopses
  - Deletions
- Empirical Evaluation

(Multiset) Union of Partitions





- ***Combine*** KMV synopses: $L = L_A \oplus L_B$
- Theorem: L is a KMV synopsis of $A \cup B$
- Can use previous unbiased estimator: $\hat{D}_k^{UB} = (k-1)/U_{(k)}$

(Multiset) Intersection of Partitions

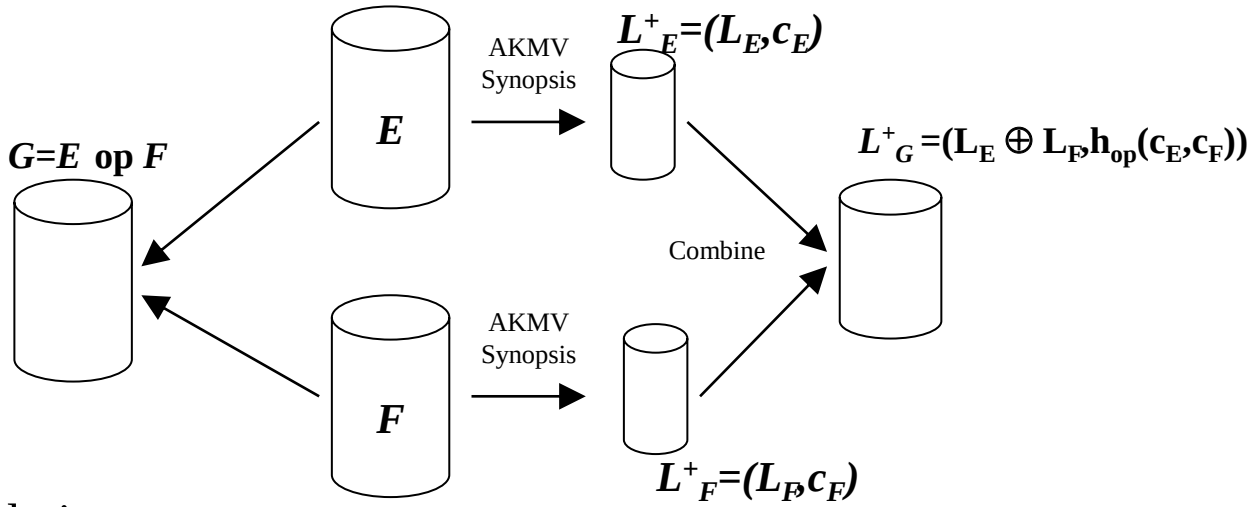
- $L = L_A \oplus L_B$ as with union (contains k elements)
  - Note: L corresponds to a uniform random sample of DVs in $A \cup B$
- $K_\cap$ = # values in L that are also in $D(A \cap B)$
  - Theorem: Can compute from L_A and L_B alone
- $K_\cap / k$ estimates Jaccard distance $\frac{D_\cap}{D_\cup} = \frac{|D(A \cap B)|}{|D(A \cup B)|}$
- $\hat{D}_\cup = (k-1)/U_{(k)}$ estimates $D_\cup = |D(A \cup B)|$
- Unbiased estimator of #DVs in the intersection: $\hat{D}_\cap = \frac{K_\cap}{k}\left(\frac{k-1}{U_{(k)}}\right)$
  - See paper for variance of estimator
- Can extend to general compound partitions from ordinary set operations

Multiset Differences: AKMV Synopsis

- Augment KMV synopsis with multiplicity counters $L^+=(L,c)$
  - Space: $O(k \log D + k \log M)$ M=max multiplicity
  - Proceed almost exactly as before i.e. $L^+_{(E/F)}=(L_E \oplus L_F,(c_E - c_F)^+)$
  - Unbiased DV estimator: $\frac{K_g}{k}\left(\frac{k-1}{U_{(k)}}\right)$

    K_g is the #positive counters

- Closure property:



- Can also handle deletions