Hashing Techniques for Detecting Related Datasets

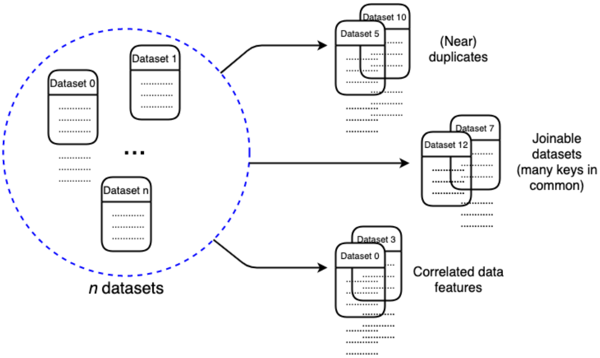
Aline Bessa

NYU Tandon

aline.bessa@nyu.edu

September 25, 2020

Why do related datasets matter?



★ Also:

- Complement datasets with similar information
- Identify primary-key/foreign-key relationships

Challenges

★ *Exactly* computing these relationships is expensive ($O(n^2)$)

★ Data may not fit in main memory $\Rightarrow$ Space and time problems

★ Analysts typically accept faster-to-produce, approximate answers

$\Rightarrow$ Can we use hashing functions for approximate, sketch-based solutions?

Sketches for Jaccard Similarity (JS)

Given columns A and B (possibly from different datasets):

$$JS(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

★ The higher $JS(A,B)$, the higher the similarity between A and B.

MinHash approximation - Given K independent random hash functions $h_1, \ldots, h_K$, we have that

$$P(min(h_i(A)) = min(h_i(B))) = \frac{|A \cap B|}{|A \cup B|}$$

we have that

$$\hat{JS}(A,B) = \frac{1}{K} \sum_{i=1}^{K} \mathbf{1}(min(h_i(A)) = min(h_i(B)))$$

Containment Ratio (CR) or Jaccard Containment (JC)

Given dataset columns A and B (possibly from different datasets):

$$CR_A(A, B) = \frac{|A \cap B|}{|A|}$$

and

$$CR_B(A, B) = \frac{|A \cap B|}{|B|}$$

★ *The higher $CR_A(A, B)$, the more B is "contained" in A.*

Record Matching with CR

★ Restaurant A - {five, guys, burgers, and, fries, downtown, brooklyn, new, york}
★ Restaurant B - {five, kitchen, berkeley}
★ Query Q - {five, guys}

$$JS(A, Q) = \frac{|A \cap Q|}{|A \cup Q|} = \frac{2}{9}$$

$$JS(B, Q) = \frac{|B \cap Q|}{|B \cup Q|} = \frac{1}{4}$$

...If what matters is our query Q, this is not very good.

Record Matching with *CR*

★ Restaurant A - {five, guys, burgers, and, fries, downtown, brooklyn, new, york}
★ Restaurant B - {five, kitchen, berkeley}
★ Query Q - {five, guys}

$$JS(A, Q) = \frac{|A \cap Q|}{|A \cup Q|} = \frac{2}{9}$$

$$JS(B, Q) = \frac{|B \cap Q|}{|B \cup Q|} = \frac{1}{4}$$

...If what matters is our query Q, this is not very good. Better: focus on Q and compute CR_Q

$$CR_Q(A, Q) = \frac{|A \cap Q|}{|Q|} = \frac{2}{2} = 1$$

$$CR_Q(B, Q) = \frac{|B \cap Q|}{|Q|} = \frac{1}{2}$$

Estimating *CR*

Can we use MinHash to estimate CR?

Do we need any extra info?

How about other hashing methods?

★ Yang Yang, Ying Zhang, Wenjie Zhang, and Zengfeng Huang. 2019. GB-KMV: An Augmented KMV Sketch for Approximate Containment Similarity Search. In *Proceedings of ICDE*. 458–469.

Sketches for relationship estimation

The estimation of data relationships in collections is receiving more attention!

- Anshumali Shrivastava and Ping Li. 2015. Asymmetric Minwise Hashing for Indexing Binary Inner Products and Set Containment. In *Proceedings of WWW*. 981–991.
- Erkang Zhu, Fatemeh Nargesian, Ken Q. Pu, and Renée J. Miller. 2016. LSH Ensemble: Internet-Scale Domain Search. *VLDB Journal* 9, 12 (2016), 1185–1196.
- Dawei Huang, Dong Young Yoon, and Seth Pettie, and Barzan Mozafar. 2019. Join on Samples: A Theoretical Guide for Practitioners. *VLDB Journal* 13, 4 (2019), 547–560.
- Raul Castro Fernandez, Jisoo Min, Demitri Nava, and Samuel Madden. 2019. LAZO: A Cardinality-Based Method for Coupled Estimation of Jaccard. In *Proceedings of ICDE*. 1190–1201.

Problem statement

Given a collection of dataset columns, find all column pairs (A, B) such that $R(A, B) > \sigma$, where R is a relationship and σ is a threshold.

- **LAZO**: approximate solution for $R = JS$ and $R = CR$ using hashing techniques! $\Rightarrow$ $O(n)$ instead of $O(n^2)$

How do we get to $O(n)$?

Step 1: Create K MinHash sketches for each dataset column in the collection.

- Hash each column in the collection K times with K independent random hash functions $h_1, \ldots, h_K$.

Sketch Matrix

Hash functions (K = 3)					
	min(h1(A))	min(h1(B))	min(h1(C))	min(h1(D))	min(h1(E))
	min(h2(A))	min(h2(B))	min(h2(C))	min(h2(D))	min(h2(E))
	min(h3(A))	min(h3(B))	min(h3(C))	min(h3(D))	min(h3(E))

Dataset columns

How do we get to $O(n)$?

★ MinHash alone *does not help* in the detection of column pairs with JS or CR above σ

★ **Locality-sensitive hashing (LSH):** Technique to identify such pairs without checking them all $\Rightarrow$ $O(n)$ instead of $O(n^2)$

- LSH indexes MinHash sketches such that those that are similar are likely to be in the same hashtable entry

How is LSH used?

Sketch Matrix

K = 6
b = 3

band 1	min(h1(A))	min(h1(B))	min(h1(C))	min(h1(D))	min(h1(E))
	min(h2(A))	min(h2(B))	min(h2(C))	min(h2(D))	min(h2(E))
band 2	min(h3(A))	min(h3(B))	min(h3(C))	min(h3(D))	min(h3(E))
	min(h4(A))	min(h4(B))	min(h4(C))	min(h4(D))	min(h4(E))
band 3	min(h5(A))	min(h5(B))	min(h5(C))	min(h5(D))	min(h5(E))
	min(h6(A))	min(h6(B))	min(h6(C))	min(h6(D))	min(h6(E))

Dataset columns

- LSH divides each MinHash sketch into a set of b bands with r rows each
- Band values for each column are concatenated and indexed into a hashtable $\Rightarrow$ values within a same band that collide share the same color (and map to the same "LSH bucket"), and are likely to contribute to a higher JS (*candidates*)

How is LSH used?

Sketch Matrix

K = 6
b = 3

band 1	min(h1(A))	min(h1(B))	min(h1(C))	min(h1(D))	min(h1(E))
	min(h2(A))	min(h2(B))	min(h2(C))	min(h2(D))	min(h2(E))
band 2	min(h3(A))	min(h3(B))	min(h3(C))	min(h3(D))	min(h3(E))
	min(h4(A))	min(h4(B))	min(h4(C))	min(h4(D))	min(h4(E))
band 3	min(h5(A))	min(h5(B))	min(h5(C))	min(h5(D))	min(h5(E))
	min(h6(A))	min(h6(B))	min(h6(C))	min(h6(D))	min(h6(E))

Dataset columns

- Parameter b is chosen based on threshold σ and should minimize false positives and negatives $\Rightarrow t = (1/b)^{(1/r)}$
- Only candidates that are hashed to a same bucket by LSH (for at least one band) are considered

How is LSH used?

Sketch Matrix

	band 1	min(h1(A))	min(h1(B))	min(h1(C))	min(h1(D))	min(h1(E))
		min(h2(A))	min(h2(B))	min(h2(C))	min(h2(D))	min(h2(E))
K = 6 **b = 3**	band 2	min(h3(A))	min(h3(B))	min(h3(C))	min(h3(D))	min(h3(E))
		min(h4(A))	min(h4(B))	min(h4(C))	min(h4(D))	min(h4(E))
	band 3	min(h5(A))	min(h5(B))	min(h5(C))	min(h5(D))	min(h5(E))
		min(h6(A))	min(h6(B))	min(h6(C))	min(h6(D))	min(h6(E))

Dataset columns

- Parameter b is chosen based on threshold σ and should minimize false positives and negatives $\Rightarrow t = (1/b)^{(1/r)}$
- Only candidates that are hashed to a same bucket by LSH (for at least one band) are considered

$\Rightarrow$ See Leskovec and Rajaraman's *Mining Massive Datasets* for details

Efficiency Gains in LAZO

	All Pairs Disk	All Pairs RAM	MinHash LSH
DWH (n=1690)	165s	9.5s	7.4s
MassData (n=5514)	261s	201s	82s
canadagov (n=97621)	-	>24h*	17min

TABLE I
RUNTIME OF ALL-PAIRS AND MINHASH/LSH FOR 3 DIFFERENT DATASETS. (* DID NOT FINISH AFTER 24 HOURS)

★ Time to find all column pairs with estimated Jaccard Similarity above $\sigma = 0.7$

★ In practice, MinHash is expensive to compute and LAZO uses a faster hashing method by default ⇒ *Optimal One-Hash Permutation (OOHP)*

How about Containment Ratio (CR)?

★ Containment Ratio in LAZO depends on $JS(A, B)$

★ $JS(A, B)$ is redefined as a function of the columns' cardinality

$$JS(A, B) = \frac{min(|A|, |B|) - \alpha}{max(|A|, |B|) + \alpha}$$

where α must be estimated.

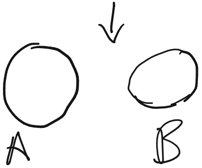
★ Compatible with faster OOHP strategy

★ Needs to store the cardinality of columns

Intuition behind the equation

$$JS(A,B) = \frac{\min(|A|, |B|) - \alpha}{\max(|A|, |B|) + \alpha}$$

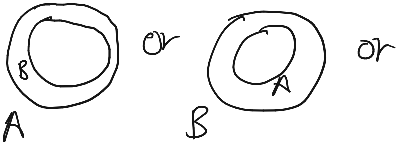
Lowest JS



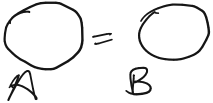
$\min(|A|, |B|) - \alpha = 0$

$\alpha = \min(|A|, |B|)$

Highest JS



$\alpha = 0$



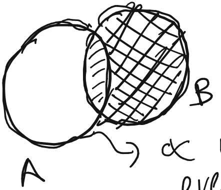
Intuition behind the equation

$\alpha \rightarrow$ # elements that are "removed" from the intersection and added to the union alone.

$C = \arg\min(|A|, |B|)$

$\underline{\underline{\alpha = C - (A \cap B)}}$

* Middle of the way



α is every element that was "inside" B (if B were inside A)

Revisiting restaurant example

★ Restaurant A - {five, guys, burgers, and, fries, downtown, brooklyn, new, york}
★ Restaurant B - {five, kitchen, berkeley}

$$C = min(|A|, |B|) = 3$$

$$\alpha = C - |A \cap B| = 3 - 1 = 2$$

$$JS(A, B) = \frac{min(|A|, |B|) - \alpha}{max(|A|, |B|) + \alpha} = \frac{3 - 2}{9 + 2} = \frac{1}{11}$$

Note that this is equals to:

$$JS(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{1}{11}$$

Estimating α

Given an initial estimate $JS'(A,B)$ calculated with OOHP, we have that

$$\hat{\alpha} = \frac{min(|A|,|B|) - JS'(A,B) * max(|A|,|B|)}{1 + \ JS'(A,B)}$$

And a better estimation for $JS(A,B)$ would then be

$$\hat{JS}(A,B) = \frac{min(|A|,|B|) - \hat{\alpha}}{max(|A|,|B|) + \hat{\alpha}}$$

How about Containment Ratio (CR)?

★ $CR_A(A, B)$ and $CR_B(A, B)$ are also functions of the columns' cardinality

$$CR_A(A, B) = \frac{min(|A|, |B|) - \alpha}{|A|}$$

$$CR_B(A, B) = \frac{min(|A|, |B|) - \alpha}{|B|}$$

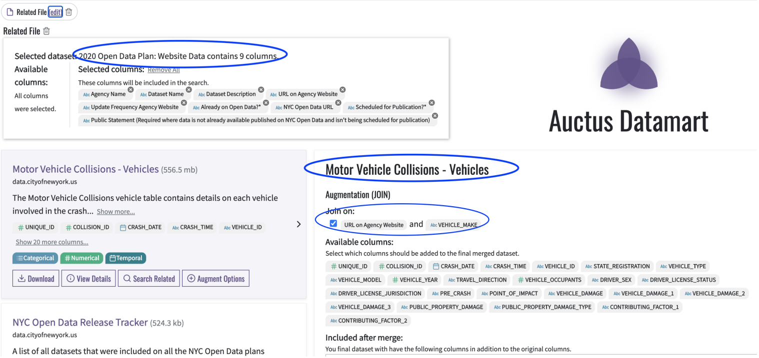
Using estimate $\hat{\alpha}$, we have that

$$\hat{CR}_A = \frac{min(|A|, |B|) - \hat{\alpha}}{|A|}$$

$$\hat{CR}_B = \frac{min(|A|, |B|) - \hat{\alpha}}{|B|}$$

- The quality of CR depends on the estimated value for JS
- This bridges the accuracy gap between MinHash and OOHP $\Rightarrow$ New contribution!

Datamart Auctus – Application of LAZO



https://auctus.vida-nyu.org/

https://gitlab.com/ViDA-NYU/datamart

Sketches for Correlation Estimation

★ What if a user is interested in datasets that *correlate* with the target of a learning task?

- Gather high-quality data for model improvement
- Gather insights about the target

★ Correlations above a certain threshold σ might help find similar data as well!

★ Aecio Santos, Aline Bessa, Chris Musco, Fernando Chirigati, and Juliana Freire. Correlation Sketches for Approximate Join-Correlation Queries. *Submitted to SIGMOD 2021*.

- *CorrelationSketches* summarize information about joinability and correlations $\Rightarrow$ Alignment across keys matters!