Fingerprints for Highly Similar Streams

Yoram Bachrach†, Ely Porat‡
†Microsoft Research, Cambridge, UK, yobach@micorosft.com
‡Bar-Ilan University, Ramat Gan, Israel, porately@cs.biu.ac.il

Abstract

We propose an approach for approximating the Jaccard similarity of two streams, $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$, for domains where this similarity is known to be high. Our method is based on a reduction from Jaccard similarity to F_2 norm estimation, for which there exists a sketch that is efficient in terms of both size and compute time, which we augment by a sampling technique. Our approach offers an improvement in the fingerprint size that is quadratic in the degree of similarity between the streams. More precisely, to approximate the Jaccard similarity up to a multiplicative factor of ϵ with confidence δ, it suffices to take a fingerprint of size $O(\ln(\frac{1}{\delta})\frac{(1-t)^2}{\epsilon^2}\log\frac{1}{1-t})$ where t is the known minimal Jaccard similarity between the streams. Further, computing our fingerprint can be done in time $O(1)$ per element in the stream.

1. Introduction

Fingerprinting methods are a key building block for massive dataset processing. They store very concise descriptions of big data streams, called "fingerprints" or "sketches", and allow approximating properties of the streams or relations between them. As these fingerprints do not allow completely reconstructing the streams themselves, but are rather designed for very specific purposes, they can be much shorter than standard compression techniques would allow.

One key problem in massive dataset processing is computing similarity between streams or sets. Specifically, the Jaccard similarity has important applications, ranging from duplicate detection [35, 38] to recommender systems [11]. An interesting property of such applications is that they only require a highly accurate approximation of the Jaccard similarity when this similarity is high. For example, collaborative filtering based recommender systems attempt to locate users who examined a set of items that is similar to a target user. The system "weeds out" users who examined very *dissimilar* item sets, then takes the highly similar users and recommends an item based on a highly accurate estimate of the similarity between their sets and the target user's set [11].

Similarly, in web page duplicate detection, when testing for near duplicates of a target web page, if an examined page has a word set that is very dissimilar from the target page, it can be ruled out as a duplicate. On the other hand, if the similarity is high, we must approximate it accurately to determine if the similarity exceeds the threshold required to classify this page as a near duplicate.

In this paper, we design fingerprints for accurately computing similarity between two streams, which become more accurate when the streams are highly similar. Our technique offers a superior performance in terms of the relation between the fingerprint size and the accuracy and confidence of the similarity estimate. The more similar we assume the target streams are, the smaller the space that our fingerprint requires: if we are only interested in the exact similarity when the Jaccard similarity between the tested streams is at least t (for some $\frac{1}{2} < t < 1$), we achieve an accuracy of ϵ using a fingerprint of size $O(\frac{(1-t)^2}{\epsilon^2} \log \frac{1}{1-t})$. Further, we only require a running time of $O(1)$ per element in the stream, as it only applies one hash from a pairwise independent hash family and one hash from a 4-wise independent hash family to each stream element. We compare our method with state of the art sketches in Section 2.

Preliminaries and Paper Outline: Consider a recommender system providing users with recommendations of items, such as movies or books, from a universe U of items. Many recommender systems are based on collaborative filtering: when a target user asks for a recommendation, the system first searches for other users who have similar consumption patterns to the target user; it then examines the items consumed by these users, seeking for items that many of these similar users have consumed, but that the user has not yet consumed. Such items are good candidate items to recommend, as many users who are similar to our target user liked these items. However, to use this approach we need to define a notion of similarity between users, and construct a good algorithm to compute the similarity between any two users. Each user is characterized by a set of items they consumed in the past, and the similarity between two users can be measured by the Jaccard similarity between these two item sets. Given item sets, $A \subseteq U$ and $B \subseteq U$, the Jaccard similarity is: $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$.

A naive approach would be storing the full list of items each user has consumed. However, in many large scale systems it is difficult to fit this entire dataset in memory. An alternative approach in such situations is fingerprinting: rather than storing the entire set of consumed items for each user, we store a short representation for each user, called a fingerprint. As opposed to lossless compression, we cannot use a user's fingerprint to reconstruct the entire set of items the user has consumed. Instead, given two fingerprints of two users, we can get a good approximation for the similarity between these two users (for example, the Jaccard similarity between the item sets they have consumed). Good fingerprints are designed to allow this while using very little space, so that the fingerprints of all users could be stored on main memory. [1]

[1]We assume that the user items sets are given in advance. We then apply some processing on this data to construct the user fingerprints. Clearly, if the set of items representing a user is changed (for example, when a user consumes a new item), the fingerprint must be recomputed. However, we do not need to assume a specific order of arrival for the user items,

When providing a recommendation to a target user, most collaborative filtering recommender systems only examine users who are "similar enough" to the target user. In other words, these systems ignore the items consumed by users who are dissimilar to the target user (or at least, they place such a low weight on these users that it suffices to only examine the items of users who are highly similar to the target user to decide on the recommendation to be made). We thus assume that we only require an accurate similarity estimate when the two sets are highly similar, i.e. $J(A, B) > t$ where $0 \leq t \leq 1$ is some threshold value. When for two given sets A, B we have a similarity level above t, so $J(A, B) > t$, we call them *t-similar sets.* [2].

Given set A of items of a universe U of size $|U|$, the *characteristic vector* of A is a vector v_A of length $|U|$ where $v_A[i] = 1$ if $i \in A$ and $v_A[i] = 0$ when $i \notin A$. Given a vector $v = (v[1], \ldots, v[n])$, its F_p norm is defined as: $F_p(v) = (\sum_{i=1}^{n}(v[i])^p)^{\frac{1}{p}}$. We note that for a set $A \subseteq U$ and its characteristic vector v_A we have $F_2(v_A) = |A|$. Similarly, for two sets $A, B \subseteq U$ we have: $F_2(v_A - v_B) = |A \cup B| - |A \cap B| = |A| + |B| - 2 \cdot |A \cap B|$. We first show how to reduce the Jaccard similarity computation between two sets to computing the F_2 norm of a vector created by subtracting the characteristic vectors of the original sets. As there are excellent known sketches for F_2 norm estimation, we show that this allows a sketch of size $O(\frac{(1-t)^2}{\epsilon^2} \log n)$, for example by using the method of Thorup and Zhang [37]. We then show that it is possible to improve the space requirement further, by employing a sampling technique.

Section 3 shows how to reduce the Jaccard approximation problem to F_2 norm estimation. We assume that all item streams are t-similar, and show that given a fingerprint that approximates the F_2 norm up to a multiplicative factor of $1 \pm \epsilon$, we can get an approximation factor of $1 \pm \epsilon_J$ for the Jaccard similarity, where $\epsilon_J < 2(1-t)\epsilon$. When $t > \frac{1}{2}$ we have $2(1-t) < 1$, so our Jaccard approximation is more accurate than our F_2 approximation. A key property of our reduction to F_2 approximation is that we get an instance with no "heavy" elements: we reduce the Jaccard similarity problem to an instance of F_2 norm computation where all the elements in the input vector are in $\{-1, 0, +1\}$. By exploiting this fact together with our assumption that the item sets are at least t-similar, Section 4 shows how to augment the F_2 norm fingerprinting techniques of [1, 13] with a sampling technique to achieve an improved fingerprint size of $O(\frac{(1-t)^2}{\epsilon^2} \log \frac{1}{1-t})$. Section 4.1 discusses the F_2 with the improved space require-

or that the items for two users arrive "in pairs".

[2]Choosing the appropriate threshold t is a design choice. In the collaborative filtering scenario, for example, one may examine a running recommender system and modify it to only take into account users who are t similar to the target user. For low values of t (i.e. when the similarity threshold is so low that almost all users are considered when making the recommendation), the resulting recommendation is unlikely to change. As the chosen value for t becomes higher, this modified recommender is likely to start outputting different recommendations from the original system. We suggest using a value of t where most recommendations stay the same. This allows applying fingerprinting techniques to improve space requirements, without changing the resulting recommendations in most cases.

ment, and the required fingerprint size to achieve accuracy ϵ_J for the Jaccard similarity, showing that it suffices to use $O(\frac{(1-t)^2}{\epsilon_J^2} \log \frac{1}{1-t})$ bits.

2. Related Work

Fingerprinting techniques have many applications, including estimating frequency moments and L_p norms [1, 25, 19], data summarization and subpopulation-size queries [16, 15], approximate membership [32], greedy list intersection [29], hamming distance sketching [31], approximate edit distance and pattern matching [34, 14, 2], compressed sensing [23], approximating rarity and similarity for data streams [17, 9] and sketching for recommender systems [11, 7, 6, 4]. Many fingerprints rely on random hashing, typically hashes chosen from a family that must exhibit certain independence properties. One example of a hash family with strong properties is the family of min-wise independent hash functions [33, 12], which are slower to compute than simpler families. For some approaches it is sufficient to use k-wise independent families of hash functions with a range R (see for example [18]), where for any k distinct elements and any k target values, the probability that each element is mapped to its respective target value under k hashes chosen randomly from the family is $\frac{1}{|R|^k}$. As these hashes are faster to apply, such approaches can achieve a fast running time. Indyk [24] shows how to build an approximately min-wise independent family using hashes which are $O(\log \frac{1}{\epsilon})$-independent, and recently it was shown [20] that the k smallest elements of a hash randomly chosen from a family of $O(1)$-independent hashes behave almost uniformly where $k > O(\frac{1}{\epsilon^2})$. Our method is similar to [1, 37], and thus also utilizes random hashes from 2-wise or 4-wise independent families. However, we utilize the restricted inputs generated when reducing Jaccard similarity to F_2 norm estimation.

Current state of the art similarity sketches either require a higher compute time [9, 10, 21, 22] or more space [37, 21, 26]. Table 2 compares state of the art approaches, where ϵ is the required accuracy, and t is the minimal Jaccard similarity between the streams. Note that for a high enough similarity threshold t we have $(1-t)^2 \log \frac{1}{1-t} < 1$ so our approach offers a significant space improvement, while maintaining a constant computation time per element in the set. Also note that in our statement for the fingerprint size we disregard the space required to store the hash functions themselves.

We note that our approach is tailored to domains where the streams are guaranteed to be "similar enough" (t-similar for some value of t). To achieve this one may use a different sketch and reject streams where this criterion is not met. [3]

[3] In some domains, such as recommender systems, one may have a first layer that only targets users who are likely to be similar to the target user through alternative mechanisms. For example, when giving recommendations to a target user, one may only examine candidate users who have a similar personality profile or interest [3]. Such information about users can be obtained, for example, through social network analytics services. [8, 28, 5, 27], or by directly profiling users.

	Computation time	Fingerprint size
This paper - reduction to F_p norm	$O(1)$	$O(\frac{(1-t)^2}{\epsilon^2}\log n)$
This paper - with sampling	$O(1)$	$O(\frac{(1-t)^2}{\epsilon^2}\log\frac{1}{1-t})$
Bachrach and Porat [9]	$O(\log\frac{1}{\epsilon})$	$O(\frac{1}{\epsilon^2})$
Feigenblat, Porat and Shiftan [21]	$O(1)$	$O(\frac{1}{\epsilon^2}\log n)$
Thorup [36]	$O(1)$	$O(\frac{1}{\epsilon^2}\log n)$
Li and Koenig [30]	$O(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon})$	$O(\frac{1}{\epsilon^2})$
Kopelowitz and Porat [26] (Given oracle for fully random hashes)	$O(1)$	$O(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon}+\log\log n)$

Figure 1: Jaccard similarity fingerprint algorithms

3. Jaccard Similarity Using F_p Norm Estimation

Given two t-similar sets A, B we have $J(A,B) > t$ so $\frac{|A\cap B|}{|A\cup B|} > t$, so $|A\cup B| < \frac{|A\cap B|}{t}$, yielding: $F_2(v_A - v_B) = |A\cup B| - |A\cap B| < (1-t)\cdot|A\cup B| < \frac{(1-t)\cdot|A\cap B|}{t}$.

Consider using a fingerprint such as that in Section 4 (or other alternatives [1]), to get an estimator $F_2(\widehat{v_A - v_B})$ for $F_2(v_A - v_B)$ with an error guaranteed to be small: $(1-\epsilon)F_2(v_A - v_B) < F_2(\widehat{v_A - v_B}) < (1+\epsilon)F_2(v_A - v_B)$. Denote the error as: $\delta = F_2(\widehat{v_A - v_B}) - F_2(v_A - v_B)$. We define several estimators. [4]

Definition 1 (Estimators based on an F_2 approximation.)**.** *The intersection estimator is:* $|\widehat{A\cap B}| = \frac{|A|+|B|-F_2(\widehat{v_A-v_B})}{2}$. *The union estimator is:* $|\widehat{A\cup B}| = |A|+|B|-|\widehat{A\cap B}|$. *The Jaccard estimator is:* $\widehat{J(A,B)} = \frac{|\widehat{A\cap B}|}{|\widehat{A\cup B}|}$.

We show that the accuracy of our Jaccard estimator is better than that of the F_2 estimator, and that the bound depends on the streams' similarity.

Theorem 1 (Jaccard approximation using an F_2 approximation)**.** *The accuracy of our Jaccard estimator improves with the similarity between the streams:*

$$\widehat{J(A,B)} = (1 \pm 2(1-t)\epsilon)J(A,B)$$

Proof. We denote the errors in the union, intersection and F_2 norm estimators as $\delta_U = |\widehat{A\cup B}| - |A\cup B|$, $\delta_I = |\widehat{A\cap B}| - |A\cap B|$ and $\delta = F_2(\widehat{v_A - v_B}) - F_2(v_A - v_B)$.

[4] As $|A\cap B| = \frac{|A|+|B|-F_2(v_A-v_B)}{2}$, a natural estimator for the intersection is: $|\widehat{A\cap B}| = \frac{|A|+|B|-F_2(\widehat{v_A-v_B})}{2}$. Since $|A\cup B| = |A|+|B|-|A\cap B|$, a natural estimator for the union is: $|\widehat{A\cup B}| = |A|+|B|-|\widehat{A\cap B}|$. Finally, since $J(A,B) = \frac{|A\cap B|}{|A\cup B|}$, a natural estimator for the Jaccard similarity is $\widehat{J(A,B)} = \frac{|\widehat{A\cap B}|}{|\widehat{A\cup B}|}$.

Since $F_2(v_A - v_B) - \epsilon F_2(v_A - v_B) < F_2(\widehat{v_A - v_B}) < F_2(v_A - v_B) + \epsilon F_2(v_A - v_B)$, and $F_2(v_A - v_B) < (1-t) \cdot |A \cup B| < \frac{1-t}{t}|A \cap B|$ we have:

(1) $F_2(v_A - v_B) - \epsilon(1-t) \cdot |A \cup B| < F_2(\widehat{v_A - v_B}) < F_2(v_A - v_B) + \epsilon(1-t) \cdot |A \cup B|$

(2) $F_2(v_A - v_B) - \epsilon \cdot \dfrac{1-t}{t}|A \cap B| < F_2(\widehat{v_A - v_B}) < F_2(v_A - v_B) + \epsilon \cdot \dfrac{1-t}{t}|A \cap B|$

Thus we have:

$$(1) \quad -\epsilon(1-t) \cdot |A \cup B| < \delta < \epsilon(1-t) \cdot |A \cup B|$$

$$(2) \quad -\epsilon \cdot \frac{1-t}{t}|A \cap B| < \delta < \epsilon \cdot \frac{1-t}{t}|A \cap B|$$

Since δ, the approximation error for $F_2(v_A - v_B)$, is bounded by $\epsilon \cdot \frac{1-t}{t}|A \cap B|$ we can also bound δ_I: $\frac{|A|+|B|}{2} - \frac{1}{2}\epsilon \cdot \frac{1-t}{t}|A \cap B| < |\widehat{A \cap B}| < \frac{|A|+|B|}{2} + \frac{1}{2}\epsilon \cdot \frac{1-t}{t}|A \cap B|$, so $\delta_I = \frac{1}{2}\delta$. Thus the estimator $|\widehat{A \cup B}|$ has an error which is also bounded by $\frac{1}{2}\epsilon(1-t) \cdot |A \cup B|$. We denote $\delta_U = |\widehat{A \cup B}| - |A \cup B|$. We note that $\delta_U = -\delta_I = -\frac{1}{2}\delta$.

The approximation error for $J(A,B)$ depends δ_I and δ_U, which in turn depend on δ. Since δ is bounded $-\epsilon \cdot \frac{1-t}{t}|A \cap B| < \delta < \epsilon \cdot \frac{1-t}{t}|A \cap B|$, and $-\epsilon(1-t) \cdot |A \cup B| < \delta < \epsilon(1-t) \cdot |A \cup B|$, we obtain a bound for $\widehat{J(A,B)}$:

$$\frac{|A \cap B| - \frac{1}{2}\epsilon \cdot \frac{1-t}{t}|A \cap B|}{|A \cup B| + \frac{1}{2}\epsilon(1-t) \cdot |A \cup B|} < \widehat{J(A,B)} < \frac{|A \cap B| + \frac{1}{2}\epsilon \cdot \frac{1-t}{t}|A \cap B|}{|A \cup B| - \frac{1}{2}\epsilon(1-t) \cdot |A \cup B|}$$

$$\frac{(1 - \frac{1}{2}\epsilon\frac{1-t}{t})|A \cap B|}{(1 + \frac{1}{2}\epsilon(1-t))|A \cup B|} < \widehat{J(A,B)} < \frac{(1 + \frac{1}{2}\epsilon\frac{1-t}{t})|A \cap B|}{(1 - \frac{1}{2}\epsilon(1-t))|A \cup B|}$$

$$\frac{(1 - \frac{1}{2}\epsilon(1-t))(1 - \frac{1}{2}\epsilon\frac{1-t}{t})}{(1 - (\frac{1}{2}\epsilon(1-t))^2)}J(A,B) < \widehat{J(A,B)} < \frac{(1 + \frac{1}{2}\epsilon(1-t))(1 + \frac{1}{2}\epsilon\frac{1-t}{t})}{(1 - (\frac{1}{2}\epsilon(1-t))^2)}J(A,B)$$

$$(1 - \frac{1}{2}\epsilon(1-t))(1 - \frac{1}{2}\epsilon\frac{1-t}{t})J(A,B) < \widehat{J(A,B)} < \frac{(1 + \frac{1}{2}\epsilon(1-t))(1 + \frac{1}{2}\epsilon\frac{1-t}{t})}{(1 - (\frac{1}{2}\epsilon(1-t))^2)}J(A,B)$$

$$(1 - \frac{1}{2}\epsilon(\frac{1}{t} - t))J(A,B) < \widehat{J(A,B)} < \frac{(1 + \frac{1}{2}\epsilon(\frac{1}{t} - 1))}{(1 - (\frac{1}{2}\epsilon(1-t))^2)}J(A,B)$$

For $\epsilon < 0.1$ and $t > 0.5$ we can bound $\widehat{J(A,B)}$ in a way that is not tight, but sufficient for our purposes:

$$\widehat{J(A,B)} = (1 \pm 2(1-t)\epsilon)J(A,B)$$

□

Theorem 2. *Given t-similar streams, it suffices to use a fingerprint of size $O(\frac{(1-t)^2}{\epsilon^2}\log n)$ to approximate the Jaccard similarity with accuracy ϵ and processing time $O(1)$ per stream element.*

Proof. The F_2 linear fingerprint of Thorup-Zhang [37] has a space requirement of $O(1/\epsilon^2 \log n)$, with an $O(1)$ computation time per element. By Theorem 1, if the item sets are t-similar, the required space for any F_2 fingerprint can be reduced by a factor of $(1-t)^2$, so we get a required fingerprint size of $O(\frac{(1-t)^2}{\epsilon^2}\log n)$. $\square$

4. Fingerprints for restricted F_2 norm estimation

Section 3 examines a Jaccard fingerprint for t-similar sets, using a building block that estimates the F_2 norm of two characteristic vectors. Though F_2 norm fingerprints already exist, we propose a more space efficient approach for a restricted class of inputs. We assume an F_2 norm computation instance for a vector where all input vector elements are in $\{-1, 0, +1\}$, such as the one generated by our reduction from Jaccard similarity in Section 3. Given this restriction on the input, in Section 4.1 we design a block fingerprint that approximates the F_2 norm up to a multiplicative factor of ϵ, and computing it requires $O(1)$ time per element.

Each such F_2 block fingerprint achieves the required accuracy ϵ with probability $p_s = \frac{21}{40}$. To get the desired confidence δ we use a fingerprint comprised of multiple such blocks. Each block i results in an estimate F_i that fails to achieve accuracy ϵ with probability $p_f = 1 - p_s = \frac{19}{40}$ (i.e. with probability p_f the difference between the actual F_2 norm F and the estimate using block i, F_i, is too high: $|F - F_i| > \epsilon$), and we use the "median trick" to achieve both the desired accuracy and confidence. We take the median value from w such blocks, i.e. the median of $\{F_i\}_{i=1}^{w}$. Denote this median value as M. The median is inaccurate if it deviates the target value F by more than ϵ (i.e. $|F - M| > \epsilon$). This only happens if at least half the F_i's deviate by at least ϵ (i.e. we have $\frac{w}{2}$ blocks with an estimation error of at least ϵ). Applying the union bound, $Pr(\exists F_{j1}, \dots, F_{j\frac{w}{2}} \text{ s.t. } |F_{ji} - F| > \epsilon) \le p_f^{\frac{w}{2}}$. To guarantee this error probability does not exceed δ, we make sure that $Pr(|F - M| > \epsilon) \le p_f^{\frac{w}{2}} < \delta$, or equivalently $w > \frac{-2}{ln(p_f)} \cdot \log(\frac{1}{\delta})$. Noting that $\frac{-2}{ln(p_f)}$ is constant (about 2.7), it suffices to take $O(\log\frac{1}{\delta})$ such fingerprint blocks to get the desired confidence level δ.

Norm estimation with sampling: Our technique is based on the F_2 norm fingerprint of [1, 13]. We map elements to "buckets" in either a positive or negative sign, treating each "bucket" as a counter. As opposed to [1, 13], we sample elements, so not every element is mapped to a "bucket", keeping counters small. This sampling only works due to the fact that the inputs are restricted to be vectors v which only contain small elements, such as those generated by the reduction of Section 3, where $v_i \in \{-1, 0, +1\}^{|U|}$.

Synchronizing a parameter for all item sets: Section 3 assumes each item set $A \subseteq U$ is represented as its characteristic vector, and a fingerprint f_A for this vector is computed such that given the fingerprints f_A for $A \subseteq U$

and f_B for $B \subseteq U$ we can accurately estimate $F_2(v_A - v_B)$. When generating the fingerprint f_A for A we cannot use any information regarding any other set $B \neq A$. To use our approach, we must "synchronize" a certain number N between the fingerprints of all such sets. To do this, we use the number of elements in the fingerprinted item set. The number of items in each fingerprinted set may be different, so at first it may seem hard to see how we could "agree" on a common number for all item sets. Consider, however, two such sets $A, B \subseteq U$, with sizes $|A| = n$ and $|B| = m$. Assume w.l.o.g that $n \leq m$. When n and m are very different, the Jaccard similarity is small: if $m < \frac{n}{2}$, the Jaccard similarity is bounded: $J(A, B) < \frac{1}{2}$. We focus on the case of high Jaccard similarity $J(A, B) > t > \frac{1}{2}$, so from now on we assume $\frac{1}{2}n \leq m \leq n$. To "agree" on a parameter for all the item sets, we can thus use the closest multiple of two to the number of elements in each item set. We denote the multiple of two closest to the number of elements of a set A as N_A, and the previous multiple of two for that set as $N'_A = \frac{1}{2}N_A$. We repeat the fingerprinting process for any set A based on both N_A and N'_A. Since we are only required to estimate the Jaccard similarity between two sets when this similarity is high, then either $N_A = N_B$ or $N'_A = N_B$.

Formally, to generate the fingerprint for the set A, we choose $N = 2^i$ such that $2|A| > N \geq |A|$. In section 4.1 we describe a fingerprinting process for the chosen value of N. This process is repeated for $N' = \frac{N}{2}$ to handle the case where N is close to $|A|$. Due to the high Jaccard similarity, for any two item sets, we can find a fingerprint that was generated using a common parameter N for both sets. Note that the parameter N is a part of the fingerprint. Thus, if for two sets we have parameters that are too far apart, so one parameter is more than twice the other, we immediately return that the Jaccard similarity is small.

4.1. F_2 Norm Block Fingerprint

Our fingerprint combines the approaches of [1, 37] with a sampling technique. The algorithm of [1] maps elements to the set $\{-1, +1\}$, and sums these up to get an estimator whose expected value is the F_2 norm and whose variance is not too large. It uses hashes from a 4-wise independent family, and takes the median of many such average estimators. Our fingerprints use a similar technique, but we sample only some of the items, and map them into "buckets" of elements, each of which is used as a counter. Our fingerprint has two parameters, an integer k and a number $p \in [0, 1]$; p can be thought of as the number of samples we take (as a proportion of the total number of elements), and k controls the number of such "buckets", and is chosen as a function of the target accuracy level ϵ, so $k = O(\frac{1}{\epsilon^2})$. The technique is reminiscent of [37], but while [37] maps each element to a bucket, we may not sample it at all.

Our fingerprints use two hashes, $h_1 : U \to [0, 1]$ and $h_2 : U \to \{+1, -1\}$, where h_1 is chosen at random from a family of pairwise independent hashes, and h_2 is chosen at random from a 4-wise independent family of hashes. The random choice of the hash functions h_1, h_2 occurs prior to fingerprinting any of the item sets. These functions are used for all the fingerprints, so when we fingerprint item sets A and B we always use the same hashes. As explained earlier, we use a parameter N synchronized across all item sets. Our process takes the item set and the parameter N and generates a list of k integers, $a_1, \ldots, a_k$ (a similar process is used with the parameter $N' = \frac{N}{2}$ to generate another k integers $a'_1, \ldots, a'_k$). Essentially, our process computes "fingerprint bucket summaries" as follows. We weed out roughly a proportion $1 - kp$ of the elements, and map the remaining elements into k buckets according to their value under h_1. The value a_i is a counter, which is increased if the element is mapped to $+1$ under h_2 and decreased if the element is mapped to -1 under h_2.

Definition 2 (Fingerprint Bucket Summaries)**.** *Given a "synchronized" number N and $N' = \frac{N}{2}$, the fingerprint block summaries are two tuples of numbers, $a_1, \ldots, a_k$ and $a'_1, \ldots, a'_k$ computed using the same process, as follows. To generate $a_1, \ldots, a_k$ we set $p = min(\frac{2}{(1-t)N}, \frac{1}{k})$; To generate $a'_1, \ldots, a'_k$ we use the same process, but set $p = \frac{1}{(1-t)N'}$. We treat a_i as a "counter" for elements in a bucket i. Each a_i is initialized to zero, and updated as follows:*

```
1. For each $x \in A$
   (a) $i := \lceil \frac{h_1(x)}{p} \rceil$
   (b) if $i \leq k$ then $a_i := a_i + h_2(x)$
```

Definition 3 (F_2 Fingerprint)**.** *Given a "synchronized" number N and $N' = \frac{N}{2}$, the fingerprint of the item set A is $(|A|, N, a_1, a_2, \ldots, a_k, a'_1, \ldots, a'_k)$, where $a_1, a_2, \ldots, a_k, a'_1, \ldots, a'_k$ are the fingerprint bucket summaries of definition 2.*

We now show how to estimate $F_2(v_A - v_B)$ given the fingerprints f_A, f_B of two item sets $A, B \subseteq U$.

Definition 4 ($F_2(v_A - v_B)$ Estimator)**.** *Given fingerprints $f_A = (|A|, N, a_1, a_2, ..., a_k)$ and $f_B = (|B|, N, b_1, b_2, ..., b_k)$ we estimate $F_2(v_A - v_B)$ as:*

$$F_2(\widehat{v_A - v_B}) = \frac{1}{pk} \sum_i (a_i - b_i)^2$$

Given the estimator $F_2(\widehat{v_A - v_B})$ we compute the Jaccard similarity $J(A, B)$ as discussed in Section 3, so we desire an accuracy ϵ for this estimator. We now prove that $F_2(\widehat{v_A - v_B})$ is an accurate estimation for $F_2(v_A - v_B)$ and discuss how p and k are chosen depending on this desired accuracy.

Definition 5 (Bucket elements)**.** *Given a bucket i, denote $X_i = (a_i - b_i)^2$ and $X = 1/(pk) \sum X_i$. Denote the unique elements as $D = (A \cup B) \setminus (A \cap B)$ and the unique elements falling in bucket i as $D_i = \{x \in D | (i-1)p < h_1(x) < ip\}$.*

Lemma 1 (Estimator expectation)**.** *For any i the expected value of X_i (over the choice of hashes h_1, h_2) is $p \cdot F_2(v_A - v_B)$, and the estimator $F_2(\widehat{v_A - v_B})$ is unbiased: $E_{h_1,h_2}(F_2(\widehat{v_A - v_B})) = F_2(v_A - v_B)$.*

Proof. For any x, y we have $(h_2(x))^2 = 1$ and $E(h_2(x)h_2(y)) = 0$ and $Pr_{h_1}[x \in D_i] = p$ so we obtain: $E_{h_1,h_2}[X_i] = E_{h_1,h_2}[(\sum_{x \in D_i} h_2(x))^2] = E_{h_1,h_2}[\sum_{x \in D_i} h_2(x)^2] + 2E_{h_1,h_2}[\sum_{x<y \in D_i}(h_2(x) \cdot h_2(y))] = E_{h_1,h_2}[\sum_{x \in D_i} h_2(x)^2] = E_{h_1,h_2}[\sum_{x \in D_i} 1] = \sum_{x \in (A \cup B) \setminus (A \cap B)} Pr[x \in D_i] \cdot 1 = p \cdot |(A \cup B) \setminus (A \cap B)| = p \cdot F_2(v_A - v_B)$. Thus, $E_{h_1,h_2}(F_2(\widehat{v_A - v_B})) = E(\frac{1}{pk}\sum_i X_i) = \frac{1}{pk} \cdot k \cdot p \cdot F_2(v_A - v_B) = F_2(v_A - v_B)$. □

Our goal is to show that with probability of at least $\frac{21}{40}$, our estimate $F_2(\widehat{v_A - v_B})$ is within a factor ϵ of the correct value $F_2(v_A - v_B)$. Consider and fix any hash h_1. The elements that are mapped to "bucket" i are those elements $x \in (A \cup B) \setminus (A \cap B)$ for which $(i-1)p < h_1(x) < ip$. Denote by $E_{h_2}[X_i|D_i]$ the expected value of X_i (under h_2) given that the elements mapped to bucket i are exactly those in D_i. Denote by $E_{h_2}[X_i|D_1, D_2, \ldots, D_k]$ the expected value of X_i given that for any $j \in \{1, 2, \ldots, k\}$ the elements mapped to bucket j are D_i. Denote by $E_{h_2}[X|D_i]$ and $E_{h_2}[X|D_1, D_2, \ldots, D_k]$ the expected value of X under the above conditions, respectively.

Lemma 2 (Counters given unique bucket elements)**.** *The expected counter values are:* $E_{h_2}[X_i|D_i] = |D_i|$ *and* $E_{h_2}[X|D_1, D_2, \ldots, D_k] = \frac{1}{pk}\sum_{i=1}^{k}|D_i|$.

Proof. $E_{h_2}[X_i|D_i] = E_{h_2}[(\sum_{x \in D_i} h_2(x))^2] = E_{h_2}[\sum_{x \in D_i} h_2^2(x) + 2\sum_{x<y \in D_i} h_2(x)h_2(y)] = \sum_{x \in D_i} E_{h_2}[h_2^2(x)] + 2\sum_{x<y \in D_i} E_{h_2}[h_2(x)h_2(y)] = \sum_{x \in D_i} E_{h_2}[1] = |D_i|$.
Thus we also obtain: $E_{h_2}[X|D_1, D_2, \ldots, D_k] = E_{h_2}[\frac{1}{pk}\sum_{i=1}^{k} X_i|D_1, D_2, \ldots, D_K] = \frac{1}{pk}\sum_{i=1}^{k} E_{h_2}[X_i|D_i] = \frac{1}{pk}\sum_{i=1}^{k}|D_i|$. □

Denote by $V_{h_2}(X_i|D_i)$ the variance, over our choice of h_2, of X_i given that the elements that were mapped to bucket i are exactly those in D_i. Similarly, denote by $V_{h_2}(X|D_1, D_2 \ldots, D_k)$ the variance of X given that for any $i \in \{1, 2, \ldots, k\}$ the elements that were mapped to bucket i are exactly those in D_i.

Lemma 3 (Estimator variance given unique bucket elements)**.** *The variance of counter values is* $V_{h_2}(X_i|D_i) = 4\binom{|D_i|}{2}$*. Our estimator's variance is:* $V_{h_2}(X|D_1, D_2 \ldots, D_k) = \frac{4}{p^2k^2}\sum_{i=1}^{k}\binom{|D_i|}{2}$.

Proof. $V_{h_2}(X_i|D_i) = E_{h_2}[X_i^2|D_i] - E_{h_2}^2[X_i|D_i] = E_{h_2}[(\sum_{x \in D_i} h_2(x))^4|D_i] - (\sum_{x \in D_i} 1)^2 = \sum_{x \in D_i} E_{h_2}[h_2^4(x)] + 6\sum_{x<y \in D_i} E_{h_2}[h_2^2(x)h_2^2(y)] - (\sum_{x \in D_i} 1^2 + 2\sum_{x<y \in D_i} 1 \cdot 1) = \sum_{x \in D_i} 1 + 6\sum_{x<y \in D_i} 1 - \sum_{x \in D_i} 1 - 2\sum_{x<y \in D_i} 1 = 4\sum_{x<y \in D_i} 1 = 4\binom{|D_i|}{2}$. Denote by $E_{h_2}[X_iX_j|D_i, D_j]$ the expectation of $X_i \cdot X_j$ given that the elements mapped to bucket i are exactly D_i and those mapped to j are exactly D_j. As $D_i \cap D_j = \emptyset$ then $x_i \neq y_j$ and h_2 is chosen from a 4-wise independent family: $E_{h_2}[h_2^2(x)h_2(y_1)h_2(y_2)] = E_{h_2}[h_2^2(y)h_2(x_1)h_2(x_2)] =$

$E_{h_2}[h_2(x_1)h_2(x_2)h_2(y_1)h_2(y_2)] = 0$, so we obtain:

$$\begin{aligned}
&E_{h_2}[X_iX_j|D_i, D_j] \\
&\quad =E_{h_2}[(\sum_{x\in D_i} h_2(x))^2(\sum_{y\in D_j} h_2(y))^2|D_i, D_j] = \\
&\quad =E_{h_2}[(\sum_{x\in D_i} h_2^2(x) + 2\sum_{x_1<x_2\in D_i} h_2(x_1)h_2(x_2))\cdot \\
&\quad (\sum_{y\in D_j} h_2^2(y) + 2\sum_{y_1<y_2\in D_j} h_2(y_1)h_2(y_2))|D_i, D_j] \\
&\quad =E_{h_2}[\sum_{x\in D_i, y\in D_j} h_2^2(x)h_2^2(y)|D_i, D_j] = |D_i||D_j|
\end{aligned}$$

$$\begin{aligned}
&V_{h_2}(X|D_1, D_2 \dots, D_k) = \\
&=E_{h_2}[X^2|D_1, D_2 \dots, D_k] - E^2_{h_1,h_2}[X|D_1, D_2 \dots, D_k] \\
&=E_{h_2}[(\frac{1}{pk}\sum_{i=1}^k X_i)^2|D_1, D_2 \dots, D_k] - E^2_{h_2}[\frac{1}{pk}\sum_{i=1}^k X_i|D_1, D_2 \dots, D_k] \\
&=\frac{1}{p^2k^2}(E_{h_2}[\sum_{i=1}^k X_i^2 + 2\sum_{i<j}^k X_iX_j|D_1, D_2 \dots, D_k] - (\sum_{i=1}^k E_{h_2}[X_i|D_i])^2) \\
&=\frac{1}{p^2k^2}(\sum_{i=1}^k E_{h_2}[X_i^2|D_i] + 2\sum_{i<j}^k E_{h_2}[X_iX_j|D_i, D_j] - \sum_{i=1}^k E_{h_2}[X_i|D_i]^2 - 2\sum_{i<j}^k E_{h_2}[X_i|D_i]E_{h_2}[X_j|D_j]) \\
&=\frac{1}{p^2k^2}(\sum_{i=1}^k (E_{h_2}[X_i^2|D_i] - E_{h_2}[X_i|D_i]^2) + 2\sum_{i<j}^k (E_{h_2}[X_iX_j|D_i, D_j] - E_{h_2}[X_i|D_i]E_{h_2}[X_j|D_j])) \\
&=\frac{1}{p^2k^2}(\sum_{i=1}^k (E_{h_1,h_2}[X_i^2|D_i] - E^2_{h_1,h_2}[X_i|D_i])) = \frac{1}{p^2k^2}\sum_{i=1}^k V_{h_2}(X_i|D_i) = \frac{4}{p^2k^2}\sum_{i=1}^k \binom{|D_i|}{2}
\end{aligned}$$

□

We show that the variance $V_{h_2}(X|D_1, D_2 \dots, D_k)$ is probably small.

Lemma 4 (Estimator variance bound). *The probability of a having a high variance of the estimator is low:* $Pr_{h_1}[V_{h_2}(X) > \frac{8F_2^2(v_A - v_B)}{k}] < \frac{1}{4}$.

Proof. Consider the variables $Z_{x,y}$ where $Z_{x,y} = 1$ if there exists $1 \le i \le k$ such that $x, y \in D_i$ and 0 otherwise. $Z_{x,y}$ is a characteristic variable which is one if x and y are mapped to the same bucket under h_1 (i.e. they are a collision in some bucket). Let $Z = \sum_{x<y\in(A\cup B)\setminus(A\cap B)} Z_{x,y}$. Z is a random variable representing the number of collisions (i.e. pairs of elements in $D = (A \cup B) \setminus (A \cap B)$ which map to the same bucket under h_1). If $p \le \frac{1}{k}$ then $E_{h_1}[Z_{x,y}] = \sum_{i=1}^k p^2 = p^2k$.

$$E_{h_1}[Z] = E_{h_1}[\sum_{x<y\in D} Z_{x,y}] = \sum_{x<y\in D} E_{h_1}[Z_{x,y}] = \binom{|D|}{2} p^2k = p^2k\binom{F_2(v_A - v_B)}{2}$$

As Z is a positive random variable, we can apply Markov's inequality [5] : $Pr_{h_1}[Z > 2p^2kF_2^2(v_A - v_B)] < Pr_{h_1}[Z > 4p^2k\binom{F_2(v_A - v_B)}{2}] < 1/4$. If d elements are mapped to bucket i, then there are $\binom{d}{2}$ pairs mapped to bucket i. Thus $Z = \sum_{i=1}^{k}\binom{|D_i|}{2}$, so we obtain: $Pr_{h_1}[V_{h_2}(X) > \frac{8F_2^2(v_A - v_B)}{k}] = Pr_{h_1}[\frac{4}{p^2k^2}\sum_{i=1}^{k}\binom{|D_i|}{2} > \frac{8F_2^2(v_A - v_B)}{k}] = Pr_{h_1}[\frac{4}{p^2k^2}Z > \frac{8F_2^2(v_A - v_B)}{k}] = Pr_{h_1}[Z > 2p^2kF_2^2(v_A - v_B)] < \frac{1}{4}$. □

Lemma 5. *For $k = \frac{160}{\epsilon^2}$, if the estimator variance is small, $V_{h_2}(X) < \frac{8F_2^2(v_A - v_B)}{k}$, then it probably does not deviate significantly from its expected value: $Pr_{h_2}[|X - \frac{1}{pk}\sum_{i=1}^{k}|D_i|| > \frac{\epsilon}{2}F_2(v_A - v_B)|V_{h_2}(X) < \frac{8F_2^2(v_A - v_B)}{k}] < \frac{1}{5}$.*

Proof. We use Chebychev's inequality: $Pr_{h_2}[|X - \frac{1}{pk}\sum_{i=1}^{k}|D_i|| > \frac{\epsilon}{2}F_2(v_A - v_B)|V_{h_2}(X) < \frac{1}{k}8F_2^2(v_A - v_B) < \frac{\frac{1}{k}8F_2^2(v_A - v_B)}{(\frac{\epsilon}{2}F_2(v_A - v_B))^2} = \frac{8}{k(\frac{\epsilon}{2})^2} = \frac{32}{k\epsilon^2}$. We choose $k = \frac{160}{\epsilon^2}$ and get: $Pr_{h_2}[|X - \frac{1}{pk}\sum_{i=1}^{k}|D_i|| > \frac{\epsilon}{2}F_2(v_A - v_B)|V_{h_2}(X) < \frac{8F_2^2(v_A - v_B)}{k}] < \frac{1}{5}$. □

Our goal is to show our estimator $F_2(\widehat{v_A - v_B}) = \frac{1}{pk}\sum_i (a_i - b_i)^2$ is close to $F_2(v_A - v_B)$. If $p = \frac{1}{k}$ then each unique element from $D = (A \cup B) \setminus (A \cap B))$ is mapped to exactly *one* bucket, so $\sum_{i=1}^{k}|D_i| = F_2(v_A - v_B)$. In this case, due to Lemmas 4 and 5, with probability of at least $1 - \frac{1}{4} - \frac{1}{5} > \frac{21}{40}$ our estimator X is within an error factor of at most $\frac{\epsilon}{2}$ from $F_2(v_A - v_B)$. Thus for this case with the required success probability $\frac{21}{40}$ we estimate $F_2(v_A - v_B)$ with an error less than the required accuracy ϵ. We now handle the case where $p < \frac{1}{k}$.

Lemma 6. *The probability that $\frac{1}{pk}\sum_{i=1}^{k}|D_i|$ deviates significantly from $F_2(v_A - v_B)$ is low: $Pr_{h_1}[|\frac{1}{pk}\sum_{i=1}^{k}|D_i| - F_2(v_A - v_B)| > \frac{\epsilon}{2}F_2(v_A - v_B)] < \frac{1}{40}$.*

Proof. If $p < \frac{1}{k}$ then by Definition 2, $p = \frac{2}{(1-t)N}$. Note that $F_2(v_A - v_B) = |(A \cup B) \setminus (A \cap B)| > (1-t)|A \cup B| > (1-t)|A| > \frac{1-t}{2}N$ so $p > \frac{1}{F_2(v_A - v_B)}$. It remains to prove that: $Pr_{h_1}[|\frac{1}{pk}\sum_{i=1}^{k}|D_i| - F_2(v_A - v_B)| > \frac{\epsilon}{2}F_2(v_A - v_B)] < \frac{1}{40}$.

We observe that $\sum_{i=1}^{k}|D_i|$ is a binomial random variable $B(pk, F_2(v_A - v_B))$. Therefore: $E_{h_1}[\sum_{i=1}^{k}|D_i|] = pkF_2(v_A - v_B)$; $V_{h_1}(\sum_{i=1}^{k}|D_i|) = pk(1 - pk)F_2(v_A - v_B)$; $E_{h_1}[\frac{1}{pk}\sum_{i=1}^{k}|D_i|] = F_2(v_A - v_B)$; $V_{h_1}(\frac{1}{pk}\sum_{i=1}^{k}|D_i|) < \frac{1}{pk}F_2(v_A - v_B)$.
As $pF_2(v_A - v_B) > 1$, and applying Chebychev's inequality we get: $Pr_{h_1}[|\frac{1}{pk}\sum_{i=1}^{k}|D_i| - F_2(v_A - v_B)| > \frac{\epsilon}{2}F_2(v_A - v_B)] < \frac{\frac{1}{pk}F_2(v_A - v_B)}{(\frac{\epsilon}{2}F_2(v_A - v_B))^2} = \frac{1}{pF_2(v_A - v_B)k(\frac{\epsilon}{2})^2} < \frac{1}{k(\frac{\epsilon}{2})^2} \leq \frac{1}{40}$.
□

[5]Note that if we chose h_1 from a 4-wise independent family, we could apply Chebychev's inequality here and obtain better constants.

Theorem 3 (Accuracy and confidence for the $F_2(v_A - v_B)$ fingerprint)**.** *Our estimator* $F_2(\widehat{v_A - v_B}) = \frac{1}{pk}\sum_i (a_i - b_i)^2$ *is accurate up to an multiplicative error of* ϵ *with probability at least* $\frac{21}{40}$*:* $Pr_{h_1,h_2}[|F_2(v_A - v_B) - F_2(\widehat{v_A - v_B})| > \epsilon F_2(v_A - v_B)] < \frac{19}{40}$.

Proof. We combine Lemmas 4, 5 and 6. In order for our estimator $F_2(\widehat{v_A - v_B})$ to have a big error, either its variance is high, or it deviates significantly from its expected value $\frac{1}{pk}\sum_{i=1}^{k}|D_i|$ or this expected value deviates significantly from $F_2(v_A - v_B)$. Applying the union bound, our approximation has the required accuracy with probability at least $1 - \frac{1}{4} - \frac{1}{5} - \frac{1}{40} = \frac{21}{40}$. □

Fingerprint size analysis: Our overall desired accuracy is ϵ_J. Denote the required accuracy for the F_2 norm approximation as ϵ. We use a variable length integer encoding, storing an integer x using $2\log x + 1$ bits. [6]

Theorem 4 (Improved Fingerprint For Similar Streams)**.** *When item sets are assumed to be t-similar, if suffices to use a fingerprint of size* $O(\frac{(1-t)^2}{\epsilon_J^2}\log\frac{1}{1-t})$ *to approximate the Jaccard similarity.*

Proof. For all i we have $E_{h_1}[D_i] < \frac{2}{1-t}$. We bound the probability of getting a high D_i using Markov's inequality : $Pr_{h_1}[D_i > c\frac{2}{1-t}] < Pr_{h_1}[D_i > cE[D_i]] < \frac{1}{c}$. Thus we obtain: $\Pr_{h_1}[\log D_i > \log\frac{2}{1-t} + c] < \frac{1}{2^c}$. The expected size of the fingerprint is bounded by $k \cdot 2\log\frac{2}{1-t} + k$, and with high probability it will not be significantly bigger: $Pr_{h_1,h_2}[size(fingerprint) > k \cdot 2\log\frac{2}{1-t} + ck] < \frac{k}{2^{c-1}}$. In order to approximate $\widehat{J(A,B)} = (1 \pm (1-t)\epsilon_J)J(A,B)$, we must use a fingerprint of size $O(k \cdot \log\frac{2}{1-t}) = O(\frac{1}{\epsilon_J^2}\log\frac{1}{1-t})$. Thus, if we want to estimate $\widehat{J(A,B)} = (1 \pm \epsilon_J)J(A,B)$, we require a fingerprint size of $O(\frac{(1-t)^2}{\epsilon_J^2}\log\frac{1}{1-t})$. □

5. Conclusions

We provided a fingerprint for approximating the Jaccard similarity between streams, whose accuracy increases in the similarity between the streams. Our method reduces the Jaccard similarity problem to a restricted case of F_2 norm approximation, which for t-similar streams we solve using fingerprints of size $O(\frac{(1-t)^2}{\epsilon^2}\log\frac{1}{1-t})$. The fingerprint computation requires $O(1)$ time per element, so our approach outperforms state of the art methods in both space and time.

Our key running example was of a highly scalable recommender system, where it is impossible to store the items sets that all users have consumed. In this case, a tractable alternative is storing concise fingerprints of the users' item sets, so that the similarity between any two users can be approximated. We

[6] Though there may exist slightly better encodings, we assume the following encoding for storing x: $1^{\lceil \log|x| \rceil}0 sign(x) binary(x)$

note, however, that the technique is general, and can be used in any domain where we need to approximate the Jaccard similarity between any two highly similar streams.

One takeaway from this work is that restricting our attention to streams that exhibit a certain *relation* (in our case, to streams which are known to be highly similar), can allow designing much more efficient algorithms. In the case of recommender systems, the key insight is that while in theory we may wish to compute the degree of similarity between any two users, in practice we only use this information further down the pipeline if the similarity is high enough (i.e. we only consider users which are highly similar to the target user when choosing which item to recommend); thus, restricting our attention to streams exhibiting the required degree of similarity does not pose a real limitation on the such systems.

Several questions remain open for future research. Are there more efficient fingerprints for t-similar streams? Could our approach be generalized to other fingerprints? Finally, are there other fingerprints which can be improved based on restrictions regarding input properties (such as their similarity, their heavy hitters etc.)?

[1] Alon, N., Matias, Y., Szegedy, M., 1999. The Space Complexity of Approximating the Frequency Moments. J. Computer and System Sciences 58 (1), 137–147.

[2] Andoni, A., Goldberger, A., McGregor, A., Porat, E., 2013. Homomorphic fingerprints under misalignments: Sketching edit and shift distances. In: Proceedings of the forty-fifth annual ACM symposium on Theory of computing. ACM, pp. 931–940.

[3] Bachrach, Y., Ceppi, S., Kash, I. A., Key, P., Radlinski, F., Porat, E., Armstrong, M., Sharma, V., 2014. Building a personalized tourist attraction recommender system using crowdsourcing. In: Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp. 1631–1632.

[4] Bachrach, Y., Finkelstein, Y., Gilad-Bachrach, R., Katzir, L., Koenigstein, N., Nice, N., Paquet, U., 2014. Speeding up the xbox recommender system using a euclidean transformation for inner-product spaces. In: Proceedings of the 8th ACM Conference on Recommender systems.

[5] Bachrach, Y., Graepel, T., Kohli, P., Kosinski, M., Stillwell, D., 2014. Your digital image: factors behind demographic and psychometric predictions from social network profiles. In: Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp. 1649–1650.

[6] Bachrach, Y., Herbrich, R., 2010. Fingerprinting ratings for collaborative filteringtheoretical and empirical analysis. In: String Processing and Information Retrieval. Springer, pp. 25–36.

[7] Bachrach, Y., Herbrich, R., Porat, E., 2009. Sketching algorithms for approximating rank correlations in collaborative filtering systems. In: SPIRE.

[8] Bachrach, Y., Kosinski, M., Graepel, T., Kohli, P., Stillwell, D., 2012. Personality and patterns of facebook usage. In: proceedings of the 3rd annual ACM web science conference. ACM, pp. 24–32.

[9] Bachrach, Y., Porat, E., 2010. Fast pseudo-random fingerprints.

[10] Bachrach, Y., Porat, E., 2013. Sketching for big data recommender systems using fast pseudo-random fingerprints. In: Automata, Languages, and Programming. Springer, pp. 459–471.

[11] Bachrach, Y., Porat, E., Rosenschein, J. S., July 2009. Sketching techniques for collaborative filtering. In: IJCAI. Pasadena, California.

[12] Broder, A., Charikar, M., Frieze, A., Mitzenmacher, M., 2000. Min-wise independent permutations. JCSS 60 (3), 630–659.

[13] Charikar, M., Chen, K., Farach-Colton, M., 2002. Finding frequent items in data streams. Automata, Languages and Programming, 784–784.

[14] Clifford, R., Jalsenius, M., Porat, E., Sach, B., 2012. Pattern matching in multiple streams. In: Combinatorial Pattern Matching. Springer, pp. 97–109.

[15] Cohen, E., Duffield, N., Kaplan, H., Lund, C., Thorup, M., 2007. Sketching unaggregated data streams for subpopulation-size queries. In: PODS. ACM.

[16] Cohen, E., Kaplan, H., 2007. Summarizing data using bottom-k sketches. In: PODC. ACM, pp. 225–234.

[17] Datar, M., Muthukrishnan, S., ???? Estimating rarity and similarity over data stream windows. Algorithms ESA 2002, 323–335.

[18] Dietzfelbinger, M., 1996. Universal hashing and k-wise independent random variables via integer arithmetic without primes. STACS 96, 567–580.

[19] Feigenblat, G., Itzhaki, O., Porat, E., 2010. The frequent items problem, under polynomial decay, in the streaming model. Theoretical Computer Science 411 (34), 3048–3054.

[20] Feigenblat, G., Porat, E., Shiftan, A., 2011. Even better framework for min-wise based algorithms.

[21] Feigenblat, G., Porat, E., Shiftan, A., 2011. Exponential time improvement for min-wise based algorithms. In: SODA.

[22] Feigenblat, G., Porat, E., Shiftan, A., 2012. Exponential space improvement for minwise based algorithms. In: FSTTCS. pp. 70–85.

[23] Gilbert, A., Li, Y., Porat, E., Strauss, M., 2010. Approximate sparse recovery: optimizing time and measurements. In: Proceedings of the 42nd ACM symposium on Theory of computing. ACM, pp. 475–484.

[24] Indyk, P., 2001. A Small Approximately Min-Wise Independent Family of Hash Functions. Journal of Algorithms.

[25] Kane, D., Nelson, J., Porat, E., Woodruff, D., 2011. Fast moment estimation in data streams in optimal space. In: STOC.

[26] Kopelowitz, T., Porat, E., 2005. Efficient bit space and time complexities for bottom-k sketching.

[27] Kosinski, M., Bachrach, Y., Kohli, P., Stillwell, D., Graepel, T., 2014. Manifestations of user personality in website choice and behaviour on online social networks. Machine Learning 95 (3), 357–380.

[28] Kosinski, M., Stillwell, D., Graepel, T., 2013. Private traits and attributes are predictable from digital records of human behavior. Proceedings of the National Academy of Sciences 110 (15), 5802–5805.

[29] Krauthgamer, R., Mehta, A., Raman, V., Rudra, A., 2008. Greedy list intersection.

[30] Li, P., Koenig, C., 2010. b-Bit minwise hashing. In: Proceedings of the 19th international conference on World wide web. ACM, pp. 671–680.

[31] Lipsky, O., Porat, E., 2007. Improved sketching of hamming distance with error correcting. In: CPM. pp. 173–182.

[32] Lovett, S., Porat, E., 2010. A lower bound for dynamic approximate membership data structures. In: Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on. IEEE, pp. 797–804.

[33] Mulmuley, K., 1996. Randomized geometric algorithms and pseudorandom generators. Algorithmica 16 (4), 450–463.

[34] Porat, B., Porat, E., 2009. Exact and approximate pattern matching in the streaming model. In: FOCS.

[35] Theobald, M., Siddharth, J., Paepcke, A., 2008. Spotsigs: robust and efficient near duplicate detection in large web collections. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 563–570.

[36] Thorup, M., 2013. Bottom-k and priority sampling, set similarity and subset sums with minimal independence. In: STOC.

[37] Thorup, M., Zhang, Y., 2004. Tabulation based 4-universal hashing with applications to second moment estimation. In: SODA. pp. 615–624.

[38] Xiao, C., Wang, W., Lin, X., Yu, J., Wang, G., 2011. Efficient similarity joins for near-duplicate detection. ACM Transactions on Database Systems (TODS) 36 (3), 15.