Towards a Theory of Homomorphic Compression

Andrew McGregor[*]

University of Massachusetts Amherst
Amherst, MA 01002, USA
mcgregor@cs.umass.edu

Abstract. In this talk we survey recent progress on designing homomorphic fingerprints. Fingerprinting is a classic randomized technique for efficiently verifying whether two files are equal. We will discuss two extensions of this basic functionality: a) verifying whether two text files are cyclic shifts of one another and b) when the files correspond to "address books", verifying whether the resulting social network is connected. Underlying both results is the idea of homomorphic lossy compression, i.e., lossy data compression that supports a range of operations on the compressed data that correspond directly to operations on the original data.

1 Introduction

Fingerprinting is a classic technique for verifying whether two large data sets are equal. Examples include the "rolling" fingerprint of Karp and Rabin [6] and cryptographic hash functions such as MD5 [7]. More generally, *linear sketching* is a form of lossy compression that also enables the "dissimilarity" of non-identical data sets to be estimated. If we represent the data as a vector $\mathbf{x} \in \mathbb{R}^n$, then a linear sketch is just a random projection $A\mathbf{x} \in \mathbb{R}^k$ where $k \ll n$ and we choose the random matrix $A \in \mathbb{R}^{k \times n}$ in such a way that the dissimilarity between files $\mathbf{x}$ and $\mathbf{y}$ can be estimated from $A\mathbf{x} - A\mathbf{y} = A(\mathbf{x} - \mathbf{y})$.

Linear sketches are trivially homomorphic with respect to linear operations in the sense that given sketches $A\mathbf{x}$, $A\mathbf{y}$, and $A\mathbf{z}$ we can also estimate the dissimilarity between $\mathbf{x}$ and $\mathbf{w} = \alpha\mathbf{y} + \beta\mathbf{z}$ for some arbitrary $\alpha, \beta \in \mathbb{R}$ since $A\mathbf{x} - \alpha A\mathbf{y} - \beta A\mathbf{z} = A(\mathbf{x} - \mathbf{w})$. In this talk, we will survey recent work [1–3] on designing fingerprints and sketches that are also homomorphic with respect to other operations. These results have applications in data stream computation and communication complexity.

2 Homomorphic Fingerprints

2.1 Text Misalignment

Many sketches have been proposed for dissimilarity measures that decompose coordinate-wise such as the Hamming distance between alphanumeric strings, or the Euclidean distance between vectors. However, when editing textual data, the appropriate notions of

[*] Supported by NSF CAREER Award CCF-0953754.

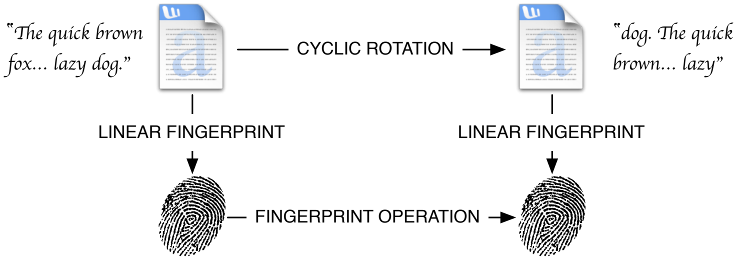

Fig. 1. Fingerprinting is an extreme form of compression that allows the Hamming distance between two files to be estimated given only the compressed form (the "fingerprint") of each file. However, traditional fingerprinting techniques perform poorly when edits result in misaligned characters. However, recent work shows that $n^{O(1/\log \log n)}$ bit fingerprints exist that are homomorphic with respect to both linear and rotation operations, i.e., given only the fingerprint of a file (and not the file itself), we can construct the fingerprint of any cyclic rotation of the file (i.e., the above diagram commutes). Such fingerprints enable us to test whether two files are within a small Hamming distance of being cyclic shifts of one another.

dissimilarity do not decompose coordinate-wise. For example, adding a single character to the start of a file and deleting a character from the end of the file will result in a new file whose Hamming distance to the original file may be proportional to the length of the file. Hence we need fingerprints that are robust to misalignments.

In recent work [3], we designed a linear sketch $L : \mathbb{Z}_m^n \to \{0,1\}^s$ that randomly projected length-n files into $s = D(n) \cdot \operatorname{polylog} n$ dimensions where $D(n)$ is the number of divisors of n. The sketch has the following properties:

1. *Soundness:* Given $L(a)$ and $L(b)$ we can determine whether the file b is a cyclic shift of a (or more generally within a constant Hamming distance of a file that is a cyclic shift of a) with probability at least $2/3$.
2. *Shift homomorphism:* Given $L(a)$ and cyclic shift σ we can compute $L(\sigma(a))$;
3. *Linear homomorphism:* Given $L(a)$ and $L(b)$ we can compute $L(a+b)$.

Furthermore, we showed that the dependence on $D(n)$ was optimal. This is somewhat surprising in the sense that we wouldn't expect a problem that is ostensibly about lossy compression to be so sensitive to a number-theoretic quantity that isn't even monotonic with the size of the data being compressed. The algorithm is based on a modification of the Karp-Rabin fingerprinting technique [6] and analyzed by appealing to properties of cyclotomic polynomials.

2.2 Graph Connectivity

Massive graphs arise in any application where there is data about both basic entities and the relationships between these entities, e.g., web-pages and hyperlinks; neurons and synapses; papers and citations; IP addresses and network flows; people and their

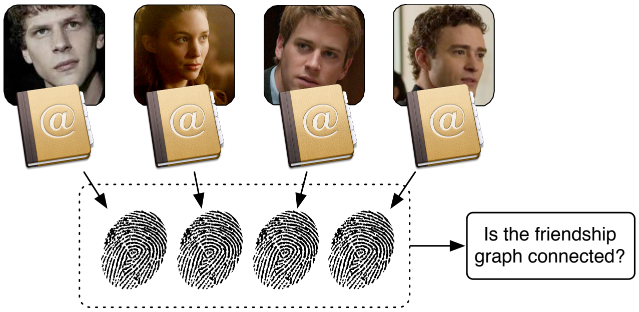

Fig. 2. Each person in a group of n people has an address book that lists their friends in the group. Without any inter-group communication, each person sends a "fingerprint" of their address book to a third party. How many bits must each fingerprint contain for the third party to determine whether the underlying friendship graph is connected with high probability? A trivial upper-bound is n bits and may appear tight. However, recent work shows that even for an arbitrary graph, it suffices for each fingerprint to contain $O(\operatorname{polylog} n)$ bits and we extend the result to approximating the size of all cuts in the graph. An example application is the first sub-linear space data structures for processing dynamic graphs.

friendships. Graphs have become the de facto standard for representing many types of highly-structured data. Relevant properties of these graphs could include dense subgraphs corresponding to tight-knit communities; shortest paths for network routing; hubs and authorities; sparse cuts and natural decompositions of the graph. However, the sheer size of some of these graphs can render classical algorithms useless when it comes to analyzing such graphs. For example, both the web graph and models of the human brain would use around 10^{10} nodes while IPv6 supports 2^{128} possible addresses. For such massive graphs we need efficient streaming and parallel algorithms.

In recent work [1, 2], we designed a fingerprinting algorithm such that, given a $O(\epsilon^{-2} \cdot \operatorname{polylog} n)$ bit fingerprint of each of the n rows of the adjacency matrix of a graph, we can approximate the size of every cut within a factor of $(1 + \epsilon)$ with high probability. See Figure 2. Since these fingerprints are linear, this result immediately implies a $O(\epsilon^{-2} \cdot n \cdot \operatorname{polylog} n)$-space stream algorithm for the dynamic graph connectivity problem (i.e., checking that the graph defined by a sequence of edge insertions and deletions is connected). Subsequently, similar ideas have also yielded data structures for dynamic connectivity with fast update and query time [5]. Underlying these results was the fact that the fingerprints developed were homomorphic to the operation of edge-contraction. The algorithm combines ℓ_0-sampling [4] with an encoding from matroid theory.

References

1. K. J. Ahn, S. Guha, and A. McGregor. Analyzing graph structure via linear measurements. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 459–467, 2012.

2. K. J. Ahn, S. Guha, and A. McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *ACM Principles of Database Systems*, pages 5–14, 2012.
3. A. Andoni, A. Goldberger, A. McGregor, and E. Porat. Homomorphic fingerprints under misalignments. In *ACM Symposium on Theory of Computing*, 2013.
4. H. Jowhari, M. Saglam, and G. Tardos. Tight bounds for lp samplers, finding duplicates in streams, and related problems. In *PODS*, pages 49–58, 2011.
5. B. Kapron, V. King, and B. Mountjoy. Dynamic graph connectivity in polylogarithmic worst case time. In *ACM-SIAM Symposium on Discrete Algorithms*, 2013.
6. R. M. Karp and M. O. Rabin. Efficient randomized pattern-matching algorithms. *IBM Journal of Research and Development*, 31(2):249–260, 1987.
7. R. Rivest. The MD5 message-digest algorithm. *RFC Editor*, 1992.