Report from Dagstuhl Seminar 17181

Theory and Applications of Hashing

Edited by

Martin Dietzfelbinger[1], Michael Mitzenmacher[2], Rasmus Pagh[3], David P. Woodruff[4], and Martin Aumüller[5]

1 **TU Ilmenau, DE,** `martin.dietzfelbinger@tu-ilmenau.de`
2 **Harvard University – Cambridge, US,** `michaelm@eecs.harvard.edu`
3 **IT University of Copenhagen, DK,** `pagh@itu.dk`
4 **IBM Almaden Center – San Jose, US,** `dpwoodru@us.ibm.com`
5 **IT University of Copenhagen, DK,** `maau@itu.dk`

Abstract

This report documents the program and the topics discussed of the 4-day Dagstuhl Seminar 17181 "Theory and Applications of Hashing", which took place May 1–5, 2017. Four long and eighteen short talks covered a wide and diverse range of topics within the theme of the workshop. The program left sufficient space for informal discussions among the 40 participants.

Seminar May 1–5, 2017 – http://www.dagstuhl.de/17181
1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity, H.3 Information Storage and Retrieval
Keywords and phrases connections to complexity theory, data streaming applications, hash function construction and analysis, hashing primitives, information retrieval applications, locality-sensitive hashing, machine learning applications
Digital Object Identifier 10.4230/DagRep.7.5.1

1 Executive Summary

Martin Dietzfelbinger
Michael Mitzenmacher
Rasmus Pagh
David P. Woodruff
Martin Aumüller

Background

The idea of hashing was proposed in the 1950s as an efficient method for implementing symbol tables in compilers. In succeeding decades it has emerged as an algorithmic tool that goes well beyond its original purpose, providing solutions for a wide range of algorithmic problems. While the theory of hashing may appear mature, in fact, many new advances have been made in recent years. Also, the number of applications has grown to an extent that few people realize how broad the reach of hashing is, or have a comprehensive overview. The aim of this seminar was to bring together researchers with an interest in hashing methods (and more generally random mappings) from various perspectives, spanning theory and a diverse set of application areas. In this way we wanted to identify opportunities for further advancing the field.

Theory

The theoretical side includes aspects such as the design of hash functions, the design of algorithm and data structure primitives, and the mathematical analysis of hashing schemes. For hash function design, since Carter and Wegman proposed universal hashing there has been a fertile research agenda identifying sufficient randomness properties of hash functions for various applications, and on the other hand devising resource-efficient hash functions having these properties. While new simple and efficient hash function constructions with strong theoretical properties keep appearing (seminar participants have contributed to this), there are still many applications of hashing for which no efficient hash function construction is known. At the same time it is increasingly clear that traditional measures of hash function performance like collision probability and independence are inadequate to describe the randomness properties needed in many applications.

While hashing is interesting in its own right, it has also become a foundational building block for higher level algorithms and data structures, each of which can in turn often find uses in a variety of application spaces. Various hash-based sketches laid the groundwork for the field of streaming algorithms, by showing how approximate counting could be done effectively. More recently, hashing algorithms have provided frameworks for similarity measures for text and images, data reconciliation, and even fast sparse Fast Fourier transform algorithms. As we construct richer, more elaborate structures on top of the hashing building blocks, we stretch what we require from hashing further.

Mathematical analysis of hashing schemes, an area enriched by early work of Knuth but grown well beyond, has inspired the development of a number of combinatorial methods and results. In the 1990s it was realized that load balancing using the best of two (or more) random choices leads to a much more even distribution, often referred to as the "power of two choices". Another great success in this area, obtained during the last decade, has been the analysis of cuckoo hashing; several seminar participants were involved in this. On the other hand, as questions become answered, new questions arise; in the case of cuckoo hashing, the effectiveness of variants including double hashing and partial-key cuckoo hashing are not understood. Beyond the hashing schemes themselves, data structures and algorithms that make use of lower-level hashing primitives further require and benefit from mathematical analysis. The fundamental connections between hashing, sparse reconstruction algorithms, and various sketching approaches are only now starting to be realized.

Applications

Hashing is of course heavily used in information storage and retrieval contexts. For example, the "power of two choices" paradigm(where several seminar participants were among the pioneers) has resulted in extremely scalable and robust architectures for distributed hash tables (also known as key-value stores).

Other applications of hashing are appearing at a tremendous rate, as systems-designers and builders become accustomed to a world where approximate answers (as opposed to exact answers) are not only sufficient, they are necessary for efficiency. Indeed, hashing was one of the key methodologies for handling big data well before "big data" was even a widely used term. Since the seminal paper of Flajolet and Martin that showed how to efficiently compute an approximate count of the number of elements in a data stream, hashing has been a central tool in the design of algorithms for data streams, where the inability to store all the information that passes requires approximations. But in recent years the use of hashing has spread to many other settings where data is stored and accessible, but scalability can be achieved only by resorting to approximation algorithms similar to those developed

in the setting of data streams. One early success story in this direction is the method for identifying near-duplicate web pages in Altavista using min-wise hashing. Another is HyperANF, a refined version of the Flajolet-Martin method that was used in 2012 to compute the distance distribution of the Facebook social network, making it by far the largest scale Milgram-like experiment ever performed. Finally, Bloom filters, invented around 1970 to provide a small-memory approximate representation of a set, have become a staple in systems, with countless variations expanding on its initial functionality, such as counts associated with set elements or aging out of set items as new elements are dynamically added.

In the field of machine learning, random mappings of data to lower-dimensional vectors that are easier to handle is of increasing importance for big data applications. This is true in particular since machine learning algorithms often work with kernelized feature vectors whose dimension far exceeds the size of data vectors. Designing randomized mappings that meet the criteria of machine learning applications has been an active research area in recent years. NIPS 2014 awarded one of two best paper awards a paper co-authored by seminar participant Shrivastava that presents a new asymmetric locality-sensitive hash function design for machine learning applications and shows how it leads to significant speedups.

The rich interplay between theory and practice in the field of hashing cannot be overstated. Applications drive the need for new algorithms and data structures based on hashing, as well as the need for more efficient classes of hash function families with provable theoretical guarantees. Specific implementations developed in the field often do not have theoretical guarantees on performance or accuracy, creating new theoretical problems and driving a need to make theory and practice meet.

Industrial relevance. The workshop topic was highly relevant for companies dealing with big data. Three seminar participants are affiliated with Google, one has worked on hashing algorithms at AT&T for a decade, and one is affiliated with VMware. Also, one organizer was affiliated with IBM at the time of the seminar.

Outcome of the seminar

The seminar brought together quite a few leading figures in the area of hashing, mixed with a good fraction of young researchers, some of which are behind many of the most exciting results in the area in recent years. Areas that were particularly well represented were: Analysis of multiple-choice hashing methods, hashing for high-dimensional search problems, applications of hashing in string algorithms, applications of hashing in machine learning, streaming (approximation) algorithms, high-performance hash function construction, and algorithm engineering. Many results in these areas were presented in 18 shorter talks. Four longer talks (by A. Andoni, A. McGregor, U. Wieder, and Q. Zhang) contributed background, overview over new results, and aspects of applications of hashing in industry. Open problems were discussed in an open problem session; four of them are included in this report.

The paper [1] on fillable arrays co-authored by J. Nelson was motivated by a talk given by T. Hagerup (see Section 3.7) and can thus be seen as a first direct result of the seminar.

We, the organizers, would like to thank all participants for their contributions in talks, material, and discussions, and in particular the speakers of the longer talks. Many thanks are due to the Dagstuhl staff both in the offices and in the castle for their support in making this seminar a success.

References

1 Jacob Teo Por Loong, Jelani Nelson, Huacheng Yu: Fillable arrays with constant time operations and a single bit of redundancy. CoRR abs/1709.09574 (2017)

2 Table of Contents

3 Overview of Talks

3.1 Beyond Locality Sensitive Hashing

Alexandr Andoni (Columbia University – New York, US)

Joint work of Piotr Indyk, Thijs Laarhoven, Huy Nguyen, Sasho Nikolov, Ilya Razenshteyn, Ludwig Schmidt, Negev Shekel-Nosatzki, Erik Waingarten

This talk will survey some recent work on approximate nearest neighbor search since 2014, when data-dependent hashing was introduced for the problem.

References

1 Alexandr Andoni, Thijs Laarhoven, Ilya P. Razenshteyn, Erik Waingarten: Optimal Hashing-based Time-Space Trade-offs for Approximate Near Neighbors. SODA 2017: 47-66
2 Alexandr Andoni, Ilya P. Razenshteyn, Negev Shekel Nosatzki: LSH Forest: Practical Algorithms Made Theoretical. SODA 2017: 67-78
3 Alexandr Andoni, Ilya P. Razenshteyn: Optimal Data-Dependent Hashing for Approximate Near Neighbors. STOC 2015: 793-801
4 Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya P. Razenshteyn, Ludwig Schmidt: Practical and Optimal LSH for Angular Distance. NIPS 2015: 1225-1233

3.2 Distance-Sensitive Hashing

Martin Aumüller (IT University of Copenhagen, DK)

Joint work of Martin Aumüller, Tobias Christiani, Rasmus Pagh, Francesco Silvestri
Main reference M. Aumüller, T. Christiani, R. Pagh, F. Silvestri, "Distance-sensitive hashing", arXiv:1703.07867 [cs.DS], 2017.
URL http://arxiv.org/abs/1703.07867

We initiate the study of *distance-sensitive hashing*, a generalization of locality-sensitive hashing that seeks a family of hash functions such that the probability of two points having the same hash value is a given function of the distance between them. More precisely, given a distance space (X, dist) and a "collision probability function" (CPF) $f : \mathbb{R} \to [0, 1]$ we seek a distribution over pairs of functions (h, g) such that for every pair of points $x, y \in X$ the collision probability is $\Pr[h(x) = g(y)] = f(\mathrm{dist}(x, y))$. Locality-sensitive hashing is the study of how fast a CPF can *decrease* as the distance grows. For many spaces f can be made exponentially decreasing even if we restrict attention to the symmetric case where $g = h$. In this paper we study how *asymmetry* makes it possible to achieve CPFs that are, for example, increasing or unimodal. Our original motivation comes from *annulus queries* where we are interested in searching for points at distance approximately r from a query point, but we believe that distance-sensitive hashing is of interest beyond this application.

3.3 Applications of Hashing in Semantic Search

Hannah Bast (Universität Freiburg, DE)

Joint work of Hannah Bast, Florian Bäurle, Björn Buchhold, Elmar Haußmann
Main reference H. Bast, F. Bäurle, B. Buchhold, E. Haußmann, "Easy access to the freebase dataset", in Proc. of the 23rd International World Wide Web Conference (WWW 2014), pp. 95–98, ACM, 2014.
URL http://doi.acm.org/10.1145/2567948.2577016

I will briefly introduce several of our systems for semantic search on (very large) knowledge bases and text, including several live demonstrations. Each of these systems critically rely on very large hash maps, and I will explain which properties are critical for which application. For one of these requirement profiles, we will live-code a relatively simple hash map together (in C++) and show that it beats the hash map from the standard template library by a large margin.

3.4 Set Similarity Search Beyond MinHash

Tobias Christiani (IT University of Copenhagen, DK)

Joint work of Tobias Christiani, Rasmus Pagh
Main reference T. Christiani, R. Pagh, "Set similarity search beyond MinHash", in Proc. of the 49th Annual ACM SIGACT Symp. on Theory of Computing (STOC 2017), pp. 1094–1107, ACM, 2017.
URL http://doi.acm.org/10.1145/3055399.3055443

We consider the problem of approximate set similarity search under Braun-Blanquet similarity $B(x, y) = |x \cap y| / \max(|x|, |y|)$. The (b_2, b_2)-approximate Braun-Blanquet similarity search problem is to preprocess a collection of sets P such that, given a query set q, if there exists $x \in P$ with $B(q, x) \geq b_1$, then we can efficiently return $x' \in P$ with $B(q, x') > b_2$. We present a simple data structure that solves this problem with space usage $O(n^{1+\rho} \log n + \sum_{x \in P} |x|)$ and query time $O(|q| n^{\rho} \log n)$ where $n = |P|$ and $\rho = \log(1/b_1)/\log(1/b_2)$. Making use of existing lower bounds for locality-sensitive hashing by O'Donnell et al. (TOCT 2014) we show that this value of ρ is tight across the parameter space, i.e., for every choice of constants $0 < b_2 < b_1 < 1$. In the case where all sets have the same size our solution strictly improves upon the value of ρ that can be obtained through the use of state-of-the-art data-independent techniques in the Indyk-Motwani locality-sensitive hashing framework (STOC 1998) such as Broder's MinHash (CCS 1997) for Jaccard similarity and Andoni et al.'s cross-polytope LSH (NIPS 2015) for cosine similarity. Surprisingly, even though our solution is data-independent, for a large part of the parameter space we outperform the currently best data-dependent method by Andoni and Razenshteyn (STOC 2015).

3.5 BDDs for Minimal Perfect Hashing: Merging Two State-Space Compression Techniques

Stefan Edelkamp (Universität Bremen, DE)

This talk will merge two different lines of research, namely

a) state-space exploration with binary decision diagrams (BDDs), that was initially proposed for Model Checking and still is state-of-the-art in AI Planning.

b) state-space compaction with (minimal) perfect hashing, which is used in the algorithm community as a memory-based index for big data (often residing on disk).

I will show how BDDs can serve as the internal representation of a perfect hash function with linear-time ranking and unranking, and how it can be used as a static dictionary and an alternative to the recent compression schemes exploiting hypergraph theory. This will also result in a simple method to split a BDD in parts of equal number of satisfying assignments and to generate random inputs for any function represented as a BDD. As a surplus, the BDD hash function is monotone.

In terms of applications, symbolic exploration with BDD constructs a succinct representation of the state space. For each layer of the search, a BDDs is generated and stored, and will later serve as an index to do extra work like the classification of game states. Based on this approach we will show, how to strongly solve Connect-Four in a combination of symbolic and explicit-state space exploration.

3.6 Synchronization Strings: Near-Optimal Codes for Insertions and Deletions

Bernhard Haeupler (Carnegie Mellon University – Pittsburgh, US)

Joint work of Bernhard Haeupler, Amirbehshad Shahrasbi
Main reference B. Haeupler, A. Shahrasbi, "Synchronization strings: codes for insertions and deletions approaching the Singleton bound", in Proc. of the 49th Annual ACM SIGACT Symp. on Theory of Computing (STOC 2017), pp. 33–46, ACM, 2017.
URL https://doi.org/10.1145/3055399.3055498

We introduce synchronization strings as a novel way of efficiently dealing with synchronization errors, i.e., insertions and deletions. Synchronization errors are strictly more general and much harder to deal with than commonly considered half-errors, i.e., symbol corruptions and erasures. For every $\varepsilon > 0$, synchronization strings allow to index a sequence with an $\varepsilon - O(1)$ size alphabet such that one can efficiently transform k synchronization errors into $(1 + \varepsilon)k$ half-errors. This powerful new technique has many applications. In this paper, we focus on designing insdel codes, i.e., error correcting block codes (ECCs) for insertion deletion channels.

While ECCs for both half-errors and synchronization errors have been intensely studied, the later has largely resisted progress. Indeed, it took until 1999 for the first insdel codes with constant rate, constant distance, and constant alphabet size to be constructed by Schulman and Zuckerman. Insdel codes for asymptotically large or small noise rates were given in 2016 by Guruswami et al. but these codes are still polynomially far from the optimal rate-distance

tradeoff. This makes the understanding of insdel codes up to this work equivalent to what was known for regular ECCs after Forney introduced concatenated codes in his doctoral thesis 50 years ago.

A direct application of our synchronization strings based indexing method gives a simple black-box construction which transforms any ECC into an equally efficient insdel code with a slightly larger alphabet size. This instantly transfers much of the highly developed understanding for regular ECCs over large constant alphabets into the realm of insdel codes. Most notably, we obtain efficient insdel codes which get arbitrarily close to the optimal rate-distance tradeoff given by the Singleton bound for the complete noise spectrum.

3.7 On-the-Fly Array Initialization in Less Space

Torben Hagerup (Universität Augsburg, DE)

Joint work of Torben Hagerup, Frank Kammer

We show that for all given $n, t, w \in \{1, 2, \ldots\}$ with $n < 2^w$, an array of n entries of w bits each can be represented on a word RAM with a word length of w bits in at most $nw + \lceil n(t/(2w))^t \rceil$ bits of uninitialized memory to support constant-time initialization of the whole array and $O(t)$-time reading and writing of individual array entries. At one end of this tradeoff, we achieve initialization and access (i.e., reading and writing) in constant time with $nw + \lceil n/w^t \rceil$ bits for arbitrary fixed t, to be compared with $nw + \Theta(n)$ bits for the best previous solution, and at the opposite end, still with constant-time initialization, we support $O(\log n)$-time access with just $nw + 1$ bits, which is optimal for arbitrary access times if the initialization executes fewer than n steps.

3.8 An Adaptive Sublinear Time Block Sparse Fourier Transform

Michael Kapralov (EPFL – Lausanne, CH)

Joint work of Michael Kapralov, Volkan Cevher, Jonathan Scarlett, Amir Zandieh
Main reference V. Cevher, M. Kapralov, J. Scarlett, A. Zandieh, "An adaptive sublinear-time block sparse fourier transform", in Proc. of the 49th Annual ACM SIGACT Symp. on Theory of Computing (STOC 2017), pp. 702–715, ACM, 2017.
URL http://doi.acm.org/10.1145/3055399.3055462

The problem of approximately computing a small number k of dominant Fourier coefficients of a vector of length n quickly, and using few samples in time domain, is known as the Sparse Fourier Transform (sparse FFT) problem. A long line of work on the sparse FFT has resulted in algorithms with $O(k \log n \log(n/k))$ runtime and $O(k \log n)$ sample complexity. These results are proved using non-adaptive algorithms, and the latter sample complexity result is essentially the best possible under the sparsity assumption alone: It is known that even adaptive algorithms must use $\Omega((k \log(n/k))/\log \log n)$ samples. By *adaptive*, we mean being able to exploit previous samples in guiding the selection of further samples.

In this work we revisit the sparse FFT problem with the added twist that the sparse coefficients approximately obey a (k_0, k_1)-block sparse model. In this model, signal frequencies

are clustered in k_0 intervals with width k_1 in Fourier space, where $k = k_0 k_1$ is the total sparsity. Signals arising in applications are often well approximated by this model with $k_0 \ll k$.

We give the first sparse FFT algorithm for (k_0, k_1)-block sparse signals with the sample complexity of $O^*(k_0 k_1 + k_0 \log(1 + k_0) \log n)$ at constant signal-to-noise ratios, and sublinear runtime. A similar sample complexity was previously achieved in the works on *model-based compressive sensing* using random Gaussian measurements, but used $\Omega(n)$ runtime. To the best of our knowledge, our result is the first sublinear-time algorithm for model based compressed sensing, and the first sparse FFT result that goes below the $O(k \log n)$ sample complexity bound.

3.9 Locality Sensitive Distortion

Ravi Kumar (Google Research – Mountain View, US)

Joint work of Flavio Chierichetti, Ravi Kumar, Alessandro Panconesi, Erisa Terolli
Main reference F. Chierichetti, R. Kumar, A. Panconesi, E. Terolli, "The Distortion of Locality Sensitive Hashing", in Proc. of the 8th Innovations in Theoretical Computer Science Conference (ITCS 2017), LIPIcs, Vol. 67, pp. 54:1–54:18, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2017.
URL https://doi.org/10.4230/LIPIcs.ITCS.2017.54

Given a pairwise similarity notion between objects, locality sensitive hashing (LSH) aims to construct a hash function family over the universe of objects such that the probability two objects hash to the same value is their similarity. LSH is a powerful algorithmic tool for large-scale applications and much work has been done to understand LSHable similarities, i.e., similarities that admit an LSH. Our work focuses on similarities that are provably non-LSHable. We propose a notion of distortion to capture the approximation of such a similarity by a similarity that is LSHable. We consider several well-known non-LSHable similarities and show tight upper and lower bounds on their distortion.

3.10 Improved ℓ_2/ℓ_2 Sparse Recovery

Yi Li (Nanyang TU – Singapore, SG)

We study the 'for-each' version of the compressed sensing and related heavy hitters problem: (1) For the compressed sensing problem, under the strongest ℓ_2/ℓ_2 error guarantee, we provide a simple scheme that uses $O(klog(n/k))$ measurements, has $O(klog^3 n)$ decoding time, and achieves $\max\{e^{-k/log^3 k}, (n/k)^{-logk}\}$ failure probability. Our result simultaneously improves the previous best scheme of Gilbert et al. (SICOMP'12) in terms of failure probability, decoding time, and column sparsity. A followup work of Gilbert at al. (ICALP'13) focuses on the very low error probability regime, and our work also improves the number of measurements and failure probability of that scheme, while achieving a similar $O(k^2 polylogn)$ decoding time. A further consequence of our arguments is that we completely resolve the measurement complexity of ℓ_2/ℓ_2 sparse recovery in the popular, average-case spiked-covariance model, providing both a new upper bound and a new lower bound. (2) For the related heavy

hitters problem, under the strongest ℓ_infty/ℓ_2 error guarantee, we provide the first optimal measurement lower bound in terms of the approximation factor eps, n, and the failure probability delta, for the full range of such parameters. Our lower bound shows that the classical Count-Sketch data structure is optimal in all parameters.

3.11 Algorithms for Massive Graphs via Linear Sketches

Andrew McGregor (University of Massachusetts – Amherst, US)

URL https://people.cs.umass.edu/ mcgregor/slides/17-dagstuhl.pdf

In this talk, we will survey recent work on using random linear projections, a.k.a. sketches, to analyze massive graphs. Sketches are useful in a variety of computational models including the dynamic graph stream model were the input is defined by a stream of edge insertions and deletions that need to be processed in small space. A large number of problems have now been studied in this model including edge and vertex connectivity, spectral sparsification, triangle counting, densest subgraph, correlation clustering, vertex cover, and matching.

3.12 Bloom Filters in Adversarial Environments

Moni Naor (Weizmann Institute – Rehovot, IL)

Joint work of Moni Naor, Eylon Yogev
Main reference M. Naor, E. Yogev, "Bloom Filters in Adversarial Environments", IACR Cryptology ePrint Archive, p. 543, IACR, 2015.
URL https://eprint.iacr.org/2015/543

Many efficient data structures use randomness, allowing them to improve upon deterministic ones. Usually, their efficiency and/or correctness are analyzed using probabilistic tools under the assumption that the inputs and queries are independent of the internal randomness of the data structure. In this talk, we consider data structures in a more robust model, which we call the adversarial model. Roughly speaking, this model allows an adversary to choose inputs and queries adaptively according to previous responses. Specifically, we consider Bloom filters and prove a tight connection between Bloom filters in this model and cryptography.

A Bloom filter represents a set S of elements approximately, by using fewer bits than a precise representation. The price for succinctness is allowing some errors: for any x in S it should always answer 'Yes', and for any x not in S it should answer 'Yes' only with small probability.

In the adversarial model, we consider both efficient adversaries (that run in polynomial time) and computationally unbounded adversaries that are only bounded in the amount of queries they can make. For computationally bounded adversaries, we show that non-trivial (memory-wise) Bloom filters exist if and only if one-way functions exist. For unbounded adversaries we show that there exists a Bloom filter for sets of size n and error eps, that is secure against t queries and uses only $O(n \cdot \log(1/\varepsilon) + t)$ bits of memory. In comparison, $n \cdot \log(1/\varepsilon)$ is the best possible even under a non-adaptive adversary.

3.13 Heavy Hitters via Cluster-Preserving Clustering

Jelani Nelson (Harvard University – Cambridge, US)

Joint work of Kasper Green Larsen, Jelani Nelson, Huy L. Nguyen, Mikkel Thorup
Main reference K. Green Larsen, J. Nelson, H. L. Nguyen, M. Thorup, "Heavy Hitters via Cluster-Preserving Clustering", in Proc. of the 57th Annual Symposium on Foundations of Computer Science (FOCS 2016), pp. 61–70, IEEE, 2017.
URL https://doi.org/10.1109/FOCS.2016.16

In the "heavy hitters" or "frequent items" problem, one must process a stream of items and report those items that occur frequently. For example, a telecommunications company may wish to find popular destination IP addresses in a packet stream across one of their links, or a search engine may wish to report popular query words. A more general problem is when there are two streams and we must report those items whose frequencies significantly deviate between them; the former problem is a special case since we can artificially pretend the first of the two streams the empty stream. Such a problem naturally arises in trend detection and anomaly detection. Several algorithms were known in the literature solving these problems, such as the CountMin sketch, CountSketch, Hierarchical CountSketch and others. The goal in designing such algorithms is (1) to guarantee finding frequent items for as lax a definition of "frequent" as possible while still limiting output size, and while having low (2) memory consumption, (3) processing time per stream item, (4) query time to report the list of frequent times, and (5) failure probability. Previous solutions could perform well on various subsets of these metrics, but not on all 5 simultaneously.

We design a new algorithm, ExpanderSketch, which performs well on all 5. Our main innovation is a novel reduction to a new graph-clustering problem we formulate, in which finding most of every cluster guarantees finding all the frequent items. We then solve this problem by devising a novel spectral clustering algorithm potentially of independent interest, based on divide-and-conquer and local search.

3.14 Approximate Near Neighbors for General Symmetric Norms

Ilya Razenshteyn (MIT – Cambridge, US)

Joint work of Alexandr Andoni, Huy L. Nguyen, Aleksandar Nikolov, Ilya Razenshteyn, Erik Waingarten
Main reference A. Andoni, H. L. Nguyen, A. Nikolov, I. P. Razenshteyn, E. Waingarten, "Approximate near neighbors for general symmetric norms", in Proc. of the 49th Annual ACM SIGACT Symp. on Theory of Computing (STOC 2017), pp. 902–913, ACM, 2017.
URL http://dx.doi.org/10.1145/3055399.3055418

We show that every *symmetric* normed space admits an efficient nearest neighbor search data structure with *doubly-logarithmic* approximation. Specifically, for every n, $d = n^{o(1)}$, and every d-dimensional symmetric norm $\|\cdot\|$, there exists a data structure for $\mathrm{poly}(\log\log n)$-approximate nearest neighbor search over $\|\cdot\|$ for n-point datasets achieving $n^{o(1)}$ query time and $n^{1+o(1)}$ space. The main technical ingredient of the algorithm is a low-distortion embedding of a symmetric norm into a low-dimensional iterated product of top-k norms.

We also show that our techniques cannot be extended to *general* norms.

3.15 Dynamic Space Efficient Hash Tables

Peter Sanders (KIT – Karlsruher Institut für Technologie, DE)

Joint work of Tobias Maier, Peter Sanders
Main reference T. Maier, P. Sanders, "Dynamic Space Efficient Hashing", in Proc. of the 25th Annual European Symposium on Algorithms (ESA 2017), LIPIcs, Vol. 87, pp. 58:1–58:14, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2017.
URL https://doi.org/10.4230/LIPIcs.ESA.2017.58

We consider space efficient hash tables that can grow and shrink dynamically and are always highly space efficient, i.e., their space consumption is always close to the lower bound even while growing and when taking into account storage that is only needed temporarily. None of the traditionally used hash tables have this property. We show how known approaches like linear probing and bucket cuckoo hashing can be adapted to this scenario by subdividing them into many subtables or using virtual memory overcommitting. However, these rather straightforward solutions suffer from slow amortized insertion times due to frequent reallocation in small increments.

Our main result is DySECT (**Dy**namic **S**pace **E**fficient **C**uckoo **T**able) which avoids these problems. DySECT consists of many subtables which grow by doubling their size. The resulting inhomogeneity in subtable sizes is equalized by the flexibility available in bucket cuckoo hashing where each element can go to several buckets each of which containing several cells. Experiments indicate that DySECT works well with load factors up to 98%. With up to 2.7 times better performance than the next best solution.

3.16 An Empirical Perspective on Locality-Sensitive Hashing

Ludwig Schmidt (MIT – Cambridge, US)

Joint work of Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, Ludwig Schmidt, Kunal Talwar
Main reference A. Andoni, P. Indyk, T. Laarhoven, I. Razenshteyn, L. Schmidt, "Practical and Optimal LSH for Angular Distance", in Proc. of the 28th Annual Conference on Advances in Neural Information Processing Systems (NIPS 2015), pp. 1225–1233, 2015.
URL http://papers.nips.cc/paper/5893-practical-and-optimal-lsh-for-angular-distance

Locality-Sensitive Hashing offers attractive theoretical guarantees for (approximate) nearest neighbor search. In this talk, I review recent empirical work on nearest neighbor algorithms and how it compares with LSH-based methods. I then describe our work on the cross-polytope hash, which combines good theoretical properties with good practical performance. Finally, I show some experiments with locality-sensitive hashing in the context of deep neural networks.

3.17 Locality Sensitive Hashing in Wild

Anshumali Shrivastava (Rice University – Houston, US)

I will talk about some of my wild explorations with probabilistic hashing algorithms and some of my very recent findings. It turns out there is another feather in the cap for LSH. It can be used for smart sampling and estimations. If we view (K, L) LSH algorithm that adaptively samples (very efficiently) data x with probability $1 - (1 - p^K)^L$ then this can be turned into efficient unbiased estimations algorithms. (p being the collision probability between query and data x). The similar sampling idea can be used to significantly reduce the computations in classical machine learning algorithms such Deep Learning (using our recent success with asymmetric hashing for inner products). In another wild exploration, I will show how Minhash is a better hashing scheme for cosine similarity than Simhash. I will highlight the computational bottleneck, i.e. the hashing time, and will show an efficient variant of minwise hashing. If time permits, I will demonstrate the use of probabilistic hashing for obtaining practical privacy-preserving algorithms.

3.18 Locality-Sensitive Hashing of Curves

Francesco Silvestri (University of Padova, IT)

Joint work of Anne Driemel, Francesco Silvestri
Main reference A. Driemel, F. Silvestri, "Locality-Sensitive Hashing of Curves", in Proc. of the 33rd Int'l Symposium on Computational Geometry (SoCG 2017), LIPIcs, Vol. 77, pp. 37:1-37:16, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2017.
URL https://doi.org/10.4230/LIPIcs.SoCG.2017.37

We study data structures for storing a set of polygonal curves in $\Re^d$ such that, given a query curve, we can efficiently retrieve similar curves from the set, where similarity is measured using the discrete Frechet distance or the dynamic time warping distance. To this end we devise the first locality-sensitive hashing schemes for these distance measures. A major challenge is posed by the fact that these distance measures internally optimize the alignment between the curves. We give solutions for different types of alignments including constrained and unconstrained versions. For unconstrained alignments, we improve over a result by Indyk from 2002 for short curves. Let n be the number of input curves and let m be the maximum complexity of a curve in the input. In the particular case where $m \leq \frac{\alpha}{4d} \log n$, for some fixed $\alpha > 0$, our solutions imply an approximate near-neighbor data structure for the discrete Frechet distance that uses space in $O(n^{1+\alpha} \log n)$ and achieves query time in $O(n^{\alpha} \log^2 n)$ and constant approximation factor. Furthermore, our solutions provide a trade-off between approximation quality and computational performance: for any parameter $k \in [m]$, we can give a data structure that uses space in $O(2^{2k} m^{k-1} n \log n + nm)$, answers queries in $O(2^{2k} m^k \log n)$ time and achieves approximation factor in $O(m/k)$.

3.19 Streaming Complexity of Approximate Pattern Matching

Tatiana Starikovskaya (University Paris-Diderot, FR)

Joint work of Raphaël Clifford, Tatiana Starikovskaya
Main reference R. Clifford, T. Starikovskaya, "Approximate Hamming distance in a stream", in Proc. of the 43rd Int'l Colloquium on Automata, Languages, and Programming (ICALP 2016), LIPIcs, Vol. 55, pp. 20:1–20:14, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2016.
URL http://dx.doi.org/10.4230/LIPIcs.ICALP.2016.20
Main reference T. Starikovskaya, "Streaming and communication complexity of approximate pattern matching", in Proc. of the 28th Symposium on Combinatorial Pattern Matching (CPM 2017), LIPIcs, Vol. 78, pp. 13:1–13:11, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2017.
URL http://dx.doi.org/10.4230/LIPIcs.CPM.2017.13

In the approximate pattern matching problem we are given two strings, a pattern and a text, and must find all approximate occurrences of the pattern in the text. An approximate occurrence of the pattern is a factor of the text such that the distance (Hamming, edit, ...) between it and the pattern is small. This problem is a fundamental problem of text processing and has myriad of applications. However, classical solutions to this problem cannot be used for processing massive data as they demonstrate unfavourable space lower bounds. In the talk I will present several new algorithms for this problem under the streaming model of computation. In this model the input is received as a stream, one item at a time, and the algorithms are not allowed to store a copy of the input without accounting for it, which leads to particularly space-efficient solutions.

3.20 Sample(x)=(a*x<=t) is a Distinguisher with Probability 1/8

Mikkel Thorup (University of Copenhagen, DK)

Main reference M. Thorup, "Sample (x) = (a*x<=t) is a Distinguisher with Probability 1/8", in Proc. of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS 2015), pp. 1277–1291, IEEE, 2015.
URL http://dx.doi.org/10.1109/FOCS.2015.82

A random sampling function Sample : $U \to \{0,1\}$ for a key universe U is a distinguisher with probability P if for any given assignment of values $v(x)$ to the keys $x \in U$, including at least one non-zero $v(x) \neq 0$, the sampled sum, that is, $\text{sum}\{v(x) | x \in U, \text{Sample}(x) = 1\}$, is non-zero with probability at least P. Here the key values may come from any commutative monoid (addition is commutative and associative and zero is neutral). Such distinguishers were introduced by Vazirani [PhD thesis 1986], and Naor and Naor used them for their small bias probability spaces [STOC'90]. Constant probability distinguishers are used for testing in contexts where the key values are not computed directly, yet where the sum is easily computed. A simple example is when we get a stream of key value pairs $(x_1, v_1), (x_2, v_2), ..., (x_n, v_n)$ where the same key may appear many times. The accumulated value of key x is $v(x) = \text{sum}\{v_i \mid x_i = x\}$. For space reasons, we may not be able to maintain $v(x)$ for every key x, but the sampled sum is easily maintained as the single value $\text{sum}\{v_i \mid \text{Sample}(x_i)\}$.

Here we show that when dealing with w-bit integers, if a is a uniform odd w-bit integer and t is a uniform w-bit integer, then $\text{Sample}(x) = [ax \mod 2^w \leq t]$ is a distinguisher with probability 1/8. Working with standard units, that is, w=8, 16, 32, 64, we exploit that w-bit multiplication works modulo 2^w, discarding overflow automatically, and then the sampling

decision is implemented by the C-code `a * x <= t`. Previous such samplers were much less computer friendly, e.g., the distinguisher of Naor and Naor [STOC'90] was more complicated and involved a 7-independent hash function.

3.21 Cuckoo Hashing with Overlapping Buckets

Stefan Walzer (TU Ilmenau, DE)

Cuckoo hashing is the task of finding an injective assignment of m objects to n memory positions, where each object e is only allowed to reside in one of $k \geq 2$ different positions that are chosen for e independently and uniformly at random. It is well understood how the number of choices k affects the critical density $c^* \in (0, 1)$ around which the probability that a valid assignment exists jumps from almost 0 to almost 1.

In order to profit from memory caches when searching for an object, it is reasonable in practice to weaken the requirement that all choices be independent, and instead assign each object k contiguous memory regions, each of size ℓ (totalling $k \cdot \ell$ choices) that can be searched with k cache faults only. In previous work, the possible memory regions formed a partition of the total memory available. Dietzfelbinger and Weidling noticed that space utilisation can be improved if memory regions are allowed to overlap imperfectly, i.e. the choices for each object are the union of k intervals of length ℓ, where the start of an interval is not necessarily a multiple of ℓ. We confirm the experimental results by deriving corresponding thresholds rigorously.

Our main tool is a theorem due to Lelarge, Leconte and Massoulié which uses methods from statistical physics. It allows to derive the thresholds from properties of infinite trees that the underlying hypergraphs converge to.

3.22 Anecdotes on Hashing

Udi Wieder (VMware – Palo Alto, US)

In this talk I will present some anecdotes encountered and some lessons learned from attempts to implement hash tables and approximate set representations in industry. My experience involves implementations with high level languages and not extremely large data sets, and as such differs from the lessons learned in big-data companies.

3.23 Efficient Algorithms for Streaming Datasets with Near-Duplicates

Qin Zhang (Indiana University – Bloomington, US)

Joint work of Djamal Belazzougui, Di Chen, Jiecao Chen, Haoyu Zhang, Qin Zhang

In this talk I will discuss how to analyze data streams with near-duplicates. I will talk about two specific problems: (1) the estimation of the number of distinct elements, and (2) similarity join under edit distance. The messages I would like to deliver are: (1) there is an algorithmic framework for estimating the number of distinct elements in the streaming model, given that the underlying metric space admit a locality sensitive hashing (LSH) with a special property. (2) For those metric spaces X that do not admit LSH (e.g., the edit distance), we can first try to embed X to another metric space which has good LSHs. (3) If we want to handle exact distance thresholds (for defining near-duplicates) then we may need to use sketches.

References

1 Di Chen and Qin Zhang, "Streaming Algorithms for Robust Distinct Elements", SIGMOD 2016
2 Djamal Belazzougui and Qin Zhang, "Edit Distance: Sketching, Streaming and Document Exchange", FOCS 2016
3 Haoyu Zhang and Qin Zhang, "EmbedJoin: Efficient Edit Similarity Joins via Embeddings", KDD 2017

4 Open Problems

4.1 The Longest Chain with $(ax + b) \bmod p$ Hashing

Mathias Bæk Tejs Knudsen (University of Copenhagen, DK)

Let p, n be integers with $p > n$ and p a prime. Let $h : \{0, 1, \ldots, p-1\} \to \{0, 1, \ldots, n-1\}$ be the hash function defined by $h(x) = ((ax + b) \bmod p) \bmod n$ where $a, b \in \{0, 1, \ldots, p-1\}$ are chosen independently and uniformly random. Consider using h to build a hash table of size n with n keys. That is, we let $X \subset \{0, 1, \ldots, p-1\}$ be a set of size n, and for $x \in X$ we put x into the i'th linked list in the table if $h(x) = i \in \{0, 1, \ldots, n-1\}$. We let $L(X)$ denote the length of the longest chain and note $L(X)$ is a random variable defined by:

$$L(X) = \max_{i \in \{0,1,\ldots,n-1\}} \{x \in X \mid h(x) = i\} .$$

The worst case time it takes to search for an element in the hash table is $\Theta(L(X))$, which motivates the study of $\mathrm{E}[L(X)]$. We let M denote the maximal expected value over all choices of X, i.e.

$$M = \max \{\mathrm{E}[L(X)] \mid X \subset \{0, 1, \ldots, p-1\}, |X| = n\} .$$

We are interested in finding the value of M.

Open question: What is the smallest non-negative real number c such that $M \leq n^{c+o(1)}$. Is it true that $c > 0$? Is it true that $c < \frac{1}{3}$? (It is an easy exercise to prove that such a c exists)

Known results: It is known that $c \in \left[0, \frac{1}{3}\right]$. The upper bound is from [1].

References

1 Knudsen, M. B. T. Linear Hashing is Awesome. Proc. 57th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 345-352. 2016.

4.2 Voronoi Choice Games

Michael Mitzenmacher (Harvard University – Cambridge, US)

Consider the unit torus. We consider a game with n players (we think of n as growing large). Each player is given $m > 1$ points on the torus chosen independently and uniformly at random, giving mn total points. (We think of m as a small constant; for example, consider the case $m = 2$.) You may assume all players know all of points given to all of the players. In the game, each player must simultaneously choose one of their m points; using those n points, we construct a Voronoi diagram, and each player obtains a score corresponding to the size of the area of the cell of the Voronoi diagram containing their point.

Open question: Is there with probability $1 - o(1)$ a pure Nash equilibrium (e.g., a selection of points for all of the players where no single individual player wants to change their decision to another point) when the goal is for each player to maximize (respectively minimize) their score?

Known results: There is little or nothing known about this specific variant, but one can come up with all sorts of variations of the problem; see [1] for details, and in particular for how individual players having their own sets of points differentiates these problems from previous Voronoi games. A more readily analyzed variant – but still very difficult – has each player obtaining points chosen uniformly at random on the boundary of the unit circle, and after each player selects one of their points, their score corresponds to the length of the clockwise arc starting at their point until the next point for another player is reached. It is known that the number of pure Nash equilibrium when each player tries to maximize their arc length in this setting is between $(0.19)^{m-1}m$ and m; for even this simpler variant, these results do not settle the question of whether a pure Nash equilibrium exists with high probability.

References

1 Boppana, M., Hod, R., Mitzenmacher, M., Morgan, T. Voronoi Choice Games. ICALP 2016: 23:1-23:13.

4.3 Space Complexity of Monotone Minimal Perfect Hashing

Rasmus Pagh (IT University of Copenhagen, DK)

Given a set S of n elements from $\{1, \ldots, u\}$ we wish to store a function f such that for each $x \in S, f(x) = \text{rank}_S(x)$, where $\text{rank}_S(x) = |\{y \in S \mid y \leq x\}|$. Such a function is referred to as a monotone minimal perfect hash function (MMPHF) for S. We want a family F of smallest possible size such that for every S of size n there exists $f \in F$ that is a MMPHF for S. This will allow us to store f in $\log_2 |F|$ bits of space. To simplify the problem, you may assume free access to a random function $h\colon \mathbb{N} \to \{0, 1\}$; the space usage should then hold with high probability over the randomness in h.

Open question: What is the best possible value of $\log_2 |F|$ for an MMPHF family F?

Known results: Lower bound of $\Omega(n)$ bits [2]. Holds even for minimal perfect hashing without monotonicity. Upper bound of $O(n \min(\log n, \log\log\log u))$ bits [1].

References

1 Belazzougui, D., Boldi, P., Pagh, R., & Vigna, S. Monotone minimal perfect hashing: searching a sorted table with $O(1)$ accesses. In Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2009), pages 785-794, 2009. SIAM.
2 Mehlhorn, K. On the program size of perfect and universal hash functions. In Proceedings of the 23th Annual Symposium on Foundations of Computer Science (FOCS 1982), pages 170–175, 1982. IEEE.

4.4 Approximate Pattern Matching in a Stream

Tatiana Starikovskaya (University Paris-Diderot, FR)

In the approximate pattern matching problem we are given a text T and a pattern P of length n, and we must decide for each prefix of T if it ends with a string that is at the edit distance at most k from P. If this is the case, we must output the edit distance and the corresponding edit operations.

Open question: What is the complexity of the problem in the streaming model? (In the streaming model we assume that the pattern and the text arrive as streams, first the pattern, then the stream. The space is all the space used by the algorithm, included the space needed to store the information about the input. The time is the worst-case time the algorithm spends for processing a prefix of the text.)

Known results: There is a streaming algorithm that uses $O(k^8\sqrt{n}log^6 n)$ space and $O((k^2\sqrt{n} + k^{13}) \log n)$ time per arrival[1]. It is the first sublinear-space algorithm for this well-known problem. The main tool is a new sketch for computing the edit distance between two strings suggested by Belazzougui and Zhang in FOCS 2016 [2].

References

1 Tatiana Starikovskaya. Communication and streaming complexity of approximate pattern matching. To appear in CPM 2017.
2 Djamal Belazzougui, Qin Zhang: Edit Distance: Sketching, Streaming, and Document Exchange. FOCS 2016: 51-60.

Participants

- Alexandr Andoni
Columbia University – New York, US
- Martin Aumüller
IT University of Copenhagen, DK
- Hannah Bast
Universität Freiburg, DE
- Djamal Belazzougui
CERIST – Algiers, DZ
- Vladimir Braverman
Johns Hopkins University – Baltimore, US
- Andrei Z. Broder
Google Inc. – Mountain View, US
- Tobias Christiani
IT University of Copenhagen, DK
- Artur Czumaj
University of Warwick – Coventry, GB
- Søren Dahlgaard
University of Copenhagen, DK
- Martin Dietzfelbinger
TU Ilmenau, DE
- Stefan Edelkamp
Universität Bremen, DE
- Tom Friedetzky
Durham University, GB
- Andreas Goerdt
TU Chemnitz, DE
- Bernhard Haeupler
Carnegie Mellon University – Pittsburgh, US
- Torben Hagerup
Universität Augsburg, DE
- Michael Kapralov
EPFL – Lausanne, CH
- Mathias Bæk Tejs Knudsen
University of Copenhagen, DK
- Ravi Kumar
Google Research – Mountain View, US
- Yi Li
Nanyang TU – Singapore, SG
- Andrew McGregor
University of Massachusetts – Amherst, US
- Michael Mitzenmacher
Harvard University – Cambridge, US
- Moni Naor
Weizmann Institute – Rehovot, IL
- Jelani Nelson
Harvard University – Cambridge, US
- Rasmus Pagh
IT University of Copenhagen, DK
- Konstantinos Panagiotou
LMU München, DE
- Ely Porat
Bar-Ilan University – Ramat Gan, IL
- Ilya Razenshteyn
MIT – Cambridge, US
- Peter Sanders
KIT – Karlsruher Institut für Technologie, DE
- Tamás Sarlós
Google Inc. – Mountain View, US
- Ludwig Schmidt
MIT – Cambridge, US
- Anshumali Shrivastava
Rice University – Houston, US
- Francesco Silvestri
University of Padova, IT
- Christian Sohler
TU Dortmund, DE
- Tatiana Starikovskaya
University Paris-Diderot, FR
- Mikkel Thorup
University of Copenhagen, DK
- Stefan Walzer
TU Ilmenau, DE
- Udi Wieder
VMware – Palo Alto, US
- Philipp Woelfel
University of Calgary, CA
- David P. Woodruff
IBM Almaden Center – San Jose, US
- Qin Zhang
Indiana University – Bloomington, US