DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF COPENHAGEN

Practical Hash Functions for Similarity Estimation and Dimensionality Reduction

Søren Dahlgaard[1,2], Mathias Bæk Tejs Knudsen[1,2], Mikkel Thorup[1]

[1]*Department of Computer Science, University of Copenhagen,* [2]*SupWiz –* `www.supwiz.com`

Objective

The aim is to determine which hash functions to use when performing similarity estimation and dimensionality reduction. Specifically:

- Do hash functions with strong theoretical properties work better in practice?
- Can we achieve no bias and strong concentration guarantees without sacrificing speed?

Conclusions

We demonstrate experimentally on both synthetic and real-world data that for FH, OPH, and similarity search with LSH:

- Multiply-shift and 2-independent hashing exhibit bias and poor concentration.
- Mixed tabulation gives the desired concentration, confirming the theory.
- Mixed tabulation is roughly 40% faster than all hash functions with similar performance in our experiments.

Hash functions

We consider Mixed Tabulation hashing of [DKRT15] comparing it with several popular hash functions.
We first perform a comparison of evaluation times for:

- Hashing 10^7 randomly chosen integers.
- Performing feature hashing [WDL+09] on the News20 data set with different hash functions.

Hash function	time ($1..10^7$)	time (News20)
Multiply-shift	7.72 ms	55.78 ms
2-wise PolyHash	17.55 ms	82.47 ms
3-wise PolyHash	42.42 ms	120.19 ms
MurmurHash3	59.70 ms	159.44 ms
CityHash	59.06 ms	162.04 ms
Blake2	3476.31 ms	6408.40 ms
Mixed tabulation	42.98 ms	90.55 ms

Applications

We consider three different applications of hash functions in this paper.

- **Feature Hashing (FH)** introduced by Weinberger, et al. [WDL+09].
- **One Permutation Hashing (OPH)** introduced by Li, et al. [LOZ12].
- **Locality-Sensitive Hashing (LSH)** introduced by Indyk and Motwani [IM98]

For LSH, we implement it using OPH with densification of Shrivastava and Li [SL14].

Data

We consider both synthetic and real-world data.

Synthetic data

For a parameter, n, we create two sets A, B such that:

- $A \cap B$ is a dense subset of $\{1, \ldots, n\}$
- $A \triangle B$ consists of $|A \cap B|$ random numbers greater than n.

This is used for OPH to test that the estimated Jaccard similarity is indeed $\frac{1}{2}$.
For feature hashing we consider one of the sets, A as a normalized indicator vector.

Real-world data

Data set	Data points	Unique features	Avg. features
MNIST	70,000	728	150
News20	19,996	1,355,191	454

For LSH we use the standard partition of 60,000 database points and 10,000 query points for MNIST and a random partition of roughly 10,000 database and query points for News20.

Results

Feature hashing

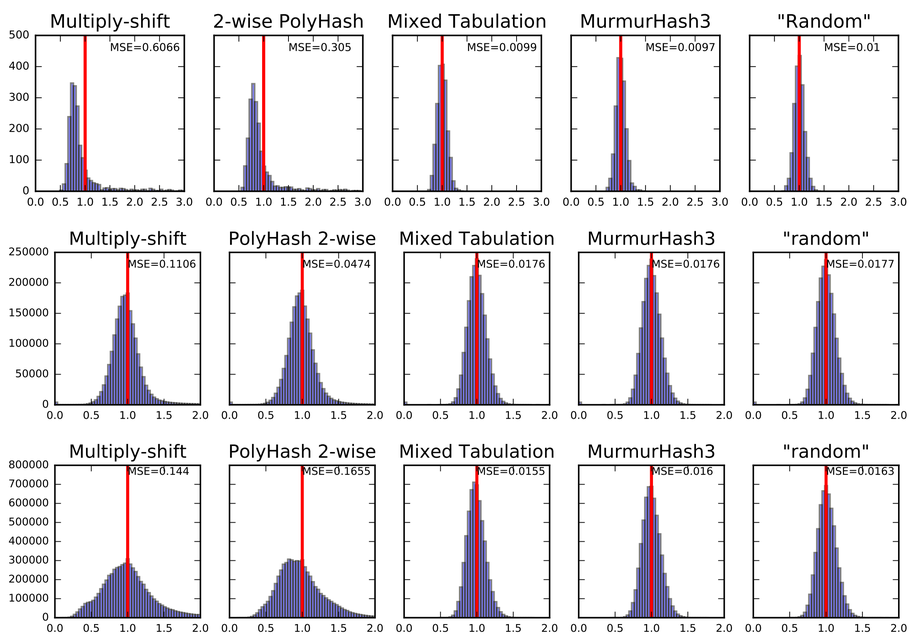

Figure 1: All experiments perform feature hashing on a set of vectors with norm 1 and plot the norms of the hashed vectors. Top: Synthetic data, Middle: News20 data set, Bottom: MNIST data set.

One permutation hashing

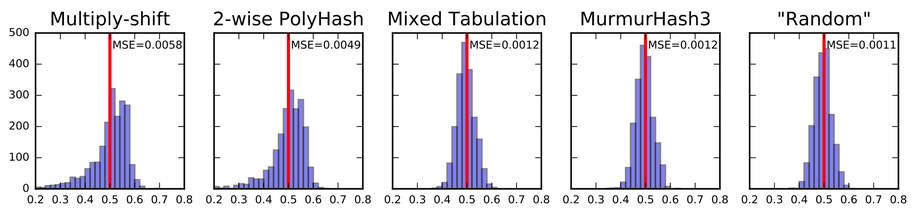

Figure 2: Set similarity estimation on synthetic data with actual similarity 1/2.

Locality-sensitive hashing

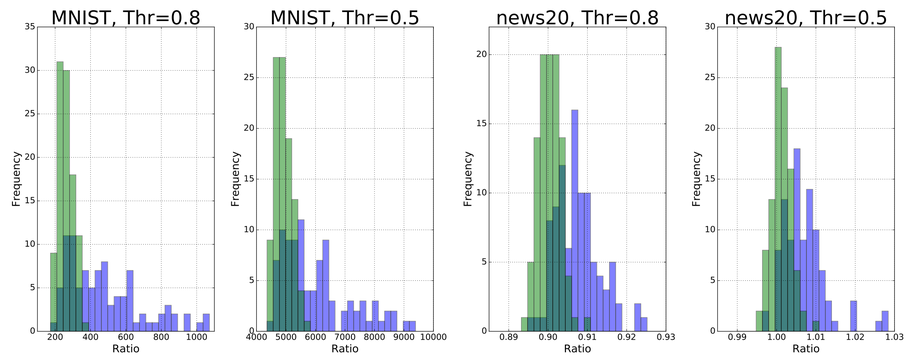

Figure 3: LSH experiments with Mixed Tabulation (green) and Multiply-shift (blue) at different similarity thresholds on MNIST (left two) and News20 (right two). The reported quantity is the number of retrieved data points divided by the fraction of recalled points (lower is better).

Acknowledgements

The authors gratefully acknowledge support from Mikkel Thorup's Advanced Grant *DFF-0602-02499B* from the *Danish Council for Independent Research* as well as the *DABAI project*. Mathias Bæk Tejs Knudsen gratefully acknowledges support from the *FNU project AlgoDisc*.

References

[DKRT15] Søren Dahlgaard, Mathias Bæk Tejs Knudsen, Eva Rotenberg, and Mikkel Thorup. Hashing for statistics over k-partitions. In *Proc. 56th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1292–1310, 2015.

[IM98] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. 13th ACM Symposium on Theory of Computing (STOC)*, pages 604–613, 1998.

[LOZ12] Ping Li, Art B. Owen, and Cun-Hui Zhang. One permutation hashing. In *Proc. 26th Advances in Neural Information Processing Systems*, pages 3122–3130, 2012.

[SL14] Anshumali Shrivastava and Ping Li. Improved densification of one permutation hashing. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI 2014, Quebec City, Quebec, Canada, July 23-27, 2014*, pages 732–741, 2014.

[WDL+09] Kilian Q. Weinberger, Anirban Dasgupta, John Langford, Alexander J. Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proc. 26th International Conference on Machine Learning (ICML)*, pages 1113–1120, 2009.