Review and Critical Analysis

Practical and Optimal LSH for Angular Distance

Andoni, Indyk, Laarhoven, Razenshteyn, & Schmidt (NIPS 2015)
https://arxiv.org/abs/1509.02897

---

1 Synthesized Summary

Locality-Sensitive Hashing (LSH) for angular distance (cosine similarity) has long suffered from a difference between theory and practice. The standard industry solution, *Hyperplane LSH* (Charikar, 2002), is computationally efficient but theoretically suboptimal. Conversely, *Spherical LSH* (Andoni et al., 2014) established the theoretical upper bound for performance but involved complex hash functions that were too slow for real-world deployment.

In this work, Andoni et al. bridge this gap by introducing Cross-Polytope LSH. They demonstrate that hashing unit vectors to the nearest vertex of a randomly rotated cross-polytope (octahedron) achieves the same asymptotic optimality as Spherical LSH. Crucially, the authors convert this theoretical construct into a practical algorithm by replacing expensive Gaussian rotations with fast pseudo-random rotations (Fast Hadamard Transform) and introducing a multiprobe query scheme. The result is an algorithm that is provably optimal among partition-based LSH families and empirically faster than Hyperplane LSH by up to $10\times$ on high-dimensional datasets.

2 The Theoretical Core: Cross-Polytope LSH

The fundamental innovation of the paper is the geometric hashing scheme. The authors propose partitioning the unit sphere S^{d-1} using the Voronoi cells of a randomly rotated cross-polytope.

2.1 Geometric Intuition

A cross-polytope in d dimensions is the convex hull of the standard basis vectors $\{\pm e_1, \ldots, \pm e_d\}$. While Hyperplane LSH essentially splits the sphere into two hemispheres (1 bit of information per hash), Cross-Polytope LSH splits the sphere into $2d$ distinct regions. A point x is hashed to the basis vector most aligned with it after a random rotation A:

$$h(x) = \arg \max_{i \in \{\pm 1, \ldots, \pm d\}} \langle Ax, e_i \rangle \tag{1}$$

The authors prove that this scheme achieves a sensitivity exponent $\rho \approx \frac{1}{2c^2-1}$ (where c is the approximation ratio). This is a strictly superior exponent compared to Hyperplane LSH ($\rho = 1/c$), meaning the Cross-Polytope method requires significantly fewer hash tables to achieve the same collision probability for approximate nearest neighbors.

2.2 Fine-Grained Optimality

The paper goes beyond proving the performance of their specific algorithm; it establishes a fundamental limit for the field. In Section 4, the authors derive a lower bound for *any* LSH family that partitions the sphere. They demonstrate that the trade-off between the "near" collision probability (p_1) and "far" collision probability (p_2) exhibited by Cross-Polytope LSH is essentially optimal. This implies that no future partition-based LSH scheme can significantly outperform this method, effectively solving the problem for this class of algorithms.

3 From Theory to Practice: Algorithm Engineering

A naive implementation of the theoretical Cross-Polytope LSH is actually *slower* than Hyperplane LSH due to the $O(d^2)$ cost of the rotation matrix A. The paper's second major contribution is the rigorous engineering required to make the optimal theory practical.

3.1 Fast Pseudo-Random Rotations

To circumvent the $O(d^2)$ matrix-vector multiplication, the authors approximate the Gaussian rotation using Fast Hadamard Transforms (FHT). They define the rotation as $HD_3HD_2HD_1$, where H is the Hadamard matrix and D_i are random diagonal sign matrices. This reduces the projection time to $O(d \log d)$, making the hash computation feasible even for high-dimensional data.

3.2 Multiprobe Scheme

LSH typically requires many hash tables (L) to ensure high recall, which consumes excessive memory. The authors adapt the "Multiprobe" technique to the cross-polytope geometry. Instead of checking only the single optimal basis vector, the query algorithm checks the top-k basis vectors that are closest to the rotated query point.

- **Impact:** This reduces the number of required hash tables significantly.
- **Implementation:** The paper describes an efficient sorting mechanism to prioritize these alternative probes based on the absolute values of the rotated vector coordinates.

3.3 Feature Hashing for Sparsity

For extremely high-dimensional sparse data (e.g., TF-IDF vectors where $d \approx 10^5$), even $O(d \log d)$ is too slow. The authors integrate "Feature Hashing" to reduce the dimensionality to a manageable subspace before applying the FHT, ensuring the algorithm scales to sparse text data.

4 Critical Evaluation of Results

The experimental section provides a nuanced view of where the proposed method succeeds.

1. **High-Dimensional Speedups:** The algorithm shines on high-dimensional data. On the PubMed and NYT text datasets (sparse, $d \approx 10^5$), Cross-Polytope LSH yields a **3.4**$\times$ – **4.0**$\times$ speedup over Hyperplane LSH. This confirms that the better ρ exponent dominates overheads when d is large.

2. **Diminishing Returns on Dense Data:** On the SIFT dataset ($d = 128$), the speedup was only **1.2**$\times$. The authors correctly attribute this to the fact that SIFT nearest neighbors are already quite close; when the "gap" between near and far points is large, simple Hyperplane LSH is already very efficient, and the complex machinery of Cross-Polytope offers less marginal gain.
3. **The Necessity of Multiprobe:** The experiments validate that the theoretical improvements alone are insufficient. Without the multiprobe optimization, the memory overhead of Cross-Polytope LSH would make it slower than Hyperplane LSH. The combination of theory (better ρ) and engineering (Multiprobe) is required for the win.

5 Conclusion

"Practical and Optimal LSH for Angular Distance" is a landmark paper because it successfully closes the loop on angular LSH. By proving a lower bound, the authors define the limit of what is theoretically possible; by engineering the Cross-Polytope scheme with FHT and Multiprobe, they provide a practical implementation that hits that limit. While it introduces implementation complexity compared to the simple Hyperplane LSH, it represents the state-of-the-art for high-dimensional cosine similarity search.