Report for CSE 5339 2018 — (OTMLSA)
Optimal Transport in Machine Learning and Shape Analysis

Fast Image Retrieval via Embeddings

Tim Carpenter

1 Introduction

The central question this paper addresses is how to build a data structure that quickly identifies the images that are closest to a query image. The authors note that early work represents images as points in multidimensional space, and uses a norm to define the distances between points. To improve the quality of the results, a variety of other metrics (such as the Earth Movers Distance (EMD)) were proposed [RTG00], but for unnormed metrics like EMD, nearest neighbor data structures such as kd-trees or R-trees cannot be used.

The main contributions of Indyk and Thaper in [IT03] is a "low distortion" embedding of EMD into $\mathbb{R}^d$ with the ℓ_1 norm, and a data structure to solve approximate nearest neighbor on this space.

2 Definitions

The follow definitions and assumptions are made in the description of the embedding algorithm.

- Let $P, Q \subset \mathbb{R}^k$ be two point sets of cardinality s, and $V = P \cup Q$.
- Assume that the smallest inter-point distance is 1, and let Δ be the diameter of V.
- For any pair $p \in P, q \in Q$, the weight of (p, q) is the Euclidean distance between p and q.
- The EMD metric $D_M(P, Q)$ is the cost of the minimum weight edge matching in the bipartite graph consisting of all edges between points in P and Q.

3 Embedding into ℓ_1

The embedding from $\mathbb{R}^k$ into ℓ_1 is constructed by the following steps:

1. Impose grids on $\mathbb{R}^k$ of side lengths $\frac{1}{2}, 1, 2, 4, \ldots, 2^i, \ldots, \Delta$. Let G_i be the grid of side length 2^i.
2. Translate each grid by a vector chosen uniformly at random from $[0, \Delta]^k$.
3. For each G_i, construct a vector $v_i(P)$ with one coordinate per cell, where each coordinate counts the number of points in the corresponding cell.
4. Define an embedding f by setting $f(P)$ to be the vector

$$v_{-1}(P)/2, v_0(P), 2v_1(P), 4v_2(P), \ldots, \Delta v_{\log \Delta}(P)$$

Two properties of this embedding should be immediately clear. First, $v(P)$ is an $O(\Delta^k)$-dimensional vector. Second, $v(P)$ has only $O(\log(\Delta) \cdot |P|)$ entries. The distortion of this embedding is bounded by the following two Lemmas.

Lemma 3.1 *For $k = 2$, there is a constant C such that for any P, Q, we have $D_M(P < Q) \leqslant C \cdot |v(P) - v(Q)|_1$.*

Lemma 3.2 *For $k = 2$, there is a constant C such that, for a fixed pair P, Q, if we shift the grids randomly, then the expected value of $|v(P) - v(Q)|_1$ is at most $C \cdot D_M(P, Q) \log \Delta$.*

The proof of these Lemmas follow by considering an optimal matching on the point sets, and comparing this matching to the matching induced by the grid structure imposed on$\mathbb{R}^k$.

Due to the high dimension of the embedding space, standard techniques for finding nearest neighbors would be too time consuming to use, and so they authors devise a scheme using locality sensitive hashing.

4 Nearest Neighbor in high dimension ℓ_1 space

To find the nearest neighbor in high-dimension ℓ_1 space, the authors use a data structure solving the (R, c)-PLEB problem.

Definition 4.1 ((R, c)-PLEB Problem) *Given n radius R balls centered at $P = \{p_1, \ldots, p_n\}$ in $\mathcal{M} = (X, D)$, devise a data structure which for any query point $q \in X$ does the following:*

- *if there exists $p \in P$ with $q \in B(p, R)$ then return YES and a point $p' \in P$ such that $q \in B(p', cR)$,*
- *if $q \notin B(p, cR)$ for all $p \in P$ then return NO,*
- *if for the point p closest to q we have $R \leqslant D(p, q) \leqslant cR$ then return either YES or NO.*

(R, c)-PLEB is a decision version of the approximate Nearest Neighbor (NN) problem. Using a binary search based approach, when $c = 1 + \epsilon$ the approximate NN problem can be reduced to $O(\log(n/\epsilon))$ instances of (R, c)-PLEB. One way to do this [IM98]:

1. Let Π be the ratio between the largest and smallest inter-point distances in the pointset $\{p_1, p_2, \ldots, p_n\}$.
2. For each $\ell \in \{(1+\epsilon)^0, (1+\epsilon)^1, \ldots, \Pi\}$ generate a sequence of balls $B_1^\ell, B_2^\ell, \ldots, B_n^\ell$ of radius ℓ centered at $p_1, p_2, \ldots, p_n$.
3. Given a query q, use binary search to find the minimal ℓ such that $q \in B_i^\ell$ and return p_i as an approximate nearest neighbor.

This method is simple, but the authors improve on it by using locality sensitive hashing (LSH).

Definition 4.2 (LSH Family) *For a domain S with distance measure D, a family $\mathcal{H} = \{h : S \to U\}$ is called (r_1, r_2, p_1, p_2)-sensitive for D if for any $v, q \in S$*

- *if $v \in B(q, r_1)$ then $\mathbf{Pr}_{\mathcal{H}}[h(q) = h(v)] \geqslant p_1$,*
- *if $v \notin B(q, r_2)$ then $\mathbf{Pr}_{\mathcal{H}}[h(q) = h(v)] \leqslant p_2$.*

A useful LSH family will satisfy $p_1 > p_2$ and $r_1 < r_2$. If we have a method of generating useful LSH families, we can apply the following Theorem to solve the PLEB, an therefore approximate NN, problem.

Theorem 4.3 *Suppose there is a (R, cR, p_1, p_2)-sensitive family $\mathcal{H}$ for a distance measure D. Then there exists an algorithm for (R, c)-PLEB under measure D which uses $O(dn+n^{1+\rho})$ space, with query time dominated by $O(n^\rho)$ distance computations, and $O(n^\rho \log_{1/p_2} n)$ evaluations of hash functions from $\mathcal{H}$, where $\rho = \frac{\ln 1/p_1}{\ln 1/p_2}$.*

To find useful LSH families, we can use *p-stable distributions*.

Definition 4.4 *A distribution $\mathcal{D}$ over $\mathbb{R}$ is called* p-stable *if there exists $p \geqslant 0$ such that for any n real numbers $v_1, \ldots, v_n$ and i.i.d. variables $X_1, \ldots, X_n$ with distribution $\mathcal{D}$, the random variable $\sum_i v_i X_i$ has the same distribution as the variable $(\sum_i |v_i|^p)^{1/p} X_i$, where X is a random variable with distribution $\mathcal{D}$.*

It is known that p-stable distributions exists for any $p \in (0, 2]$. For example, the Cauchy distribution defined by the density function $c(x) = \frac{1}{\pi(1+x^2)}$ is 1-stable. We define hash functions from $\mathbb{R}^d$ to $\mathbb{N}$ in the following way. For $v \in \mathbb{R}^d$, let $h_{\mathbf{a},b}(v) = \lfloor \frac{a.v+b}{r} \rfloor$, where

- $\mathbf{a}$ is a d-dimensional vector with entries chosen independently from a p-stable distribution,
- b is a real number chosen uniformly from the range $[0, r]$,
- r is a real number.

5 Experiments

To evaluate their image retrieval algorithm, the authors implemented and tested it on a collection of 20,000 color images from the Corel Stock Photo Library. The images were first transformed into the CIE-Lab color space.

5.1 Implementation

The authors identified the following issues that came when implementing their algorithms

ℓ_1 embedding. The embedding process described by the authors is randomized, and in particular the low-distortion guarantee is only in expectation. To increase the probability of getting good results, the authors compute 5 embeddings. This increases the query time by a factor of 5, but the authors note that their algorithm is still significantly faster than a linear scan. It also causes a 5-fold increase in the amount of space needed, but due to the sparse nature of the the embedding and the memory use being less important than the retrieval time this is not a problem.

***p*-stable LSH.** To keep low the number of random bits needed to represent the hash functions, the authors use the following approach.

- For each hash function g_j, store one random variable r_j.
- For each index i, let $u_{ji} = (ir_j \mod p)/p$ for some large prime p.
- For each u_{ji}, let $a_{ji} = \tan(\pi(u_{ji} - 1/2))$.
- For hash function $g_j = (h_{j1}, h_{j2}, \ldots, h_{jk})$, h_{ji} is defined using a_{ji}.

Furthermore, since the number of buckets in each hash function may be large, the buckets are compresses using another layer of hashing.

Parameter setting. The algorithm described by the authors has 3 parameters to set; the number of projections per hash value (call this k), the umber of hash tables (l), and the width of the projection (r).

The authors note that as the value of k is increased, the probability of two points colliding decreases exponentially (false positives), but it also increases the probability that near neighbors are placed in different buckets by the hash function (false negatives). Increasing l decreases the number of false negatives, and decreasing r decreases the number of false positives. Therefore k, l, and r should all be tuned together. For values of k, the optimal value of l can be computed directly. Through experiments on different values of k and r, the authors settled on $k = 6$ and $r = 5.0$ as giving the best tradeoff between query time and result correctness.

5.2 Experimental results

In their results, they measure the retrieval time and the accuracy of the answer, compared to the answers of a linear scan.

The first set of experiments was on 100 randomly selected query images. For their algorithms, the authors used parameters $k = 6$ and $r = 5.0$. In this experiment, the average speedup (and median) over the naive EMD computation and linear scan was 90 (resp. 59). Here the median rank of the images retrieved was 3.

In these experiments, for some of the images the rank of the retrieved image is very high (for example, in one case the retrieved image had rank 204). The authors examined the rank 204 case, and found that for all values of c, the query image had a particularly large number of c-approximate nearest neighbors compared to the other query images.

The second set of experiments conducted by the authors, they tested the effect the number of images in the database affects query time by varying the number of images from 5,000 to 20,000. These experiments demonstrates that that the embedding algorithm scales much better than the naive EMD computation and linear scan.

References

[IM98] P. Indyk and R. Motwani. Approximate nearest neighbor: toward removing the curse of dimensionality. *Proceeding of the Symposium on Theory of Computing,* 1998.

[IT03] P. Indyk and N. Thaper. Fast Image Retrieval via Embeddings. *In 3rd Intl Wkshp on Statistical and Computational Theories of Vision,* Nice, France, 2003.

[RTG00] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. 2000. The Earth Mover's Distance as a Metric for Image Retrieval. Int. J. Comput. Vision 40, 2 (November 2000), 99-121. DOI: https://doi.org/10.1023/A:1026543900054