Finding Similar Shapes: Locality Sensitive Hashing of Real World Geometries

Matt Schwennesen [1] Dr. Satish Puri [2]

[1]Michigan Technological University [2]Marquette University

Motivations

Finding similar objects or data records is a fundamental problem in data science. Many tasks such as

- Information Retrieval Systems
- Clustering
- Text Mining

require being able to determine the similarity between objects and often picking the most similar pairs or groups of objects.

Within the existing literature, little attention has been given to similarity searches between polygons, which would enable

- Using shape in other metadata classifiers.
- Geolocation of images without location tags.
- Tracking changes to geographic features over time.

Moreover, while exact similarity searches are possible they are frequently computationally prohibitive. These factors combine to create an opportunity for a new system for finding similar shapes.

Objectives

1. Create a system which is sensitive to the *Jaccard Distance* between polygons.
2. Create a system which can find near neighbors without resorting to linear scans of memory.
3. Create a system which is capable of scaling to large datasets.

Definitions

Locality Sensitive Hashing

A method for approximate nearest neighbor searches which generates hashes of objects in the dataset such that similar objects are more likely to generate the same hash. Searches examine all candidates which shares a hash with the query object.

Geometric Jaccard Distance

Suppose A and B are two different polygons. Then, the Jaccard distance of A and B is

$$J_d(A,B) = 1 - \frac{\text{Area in both } A \text{ and } B}{\text{Area in } A \text{ or } B}$$

Weighted Jaccard Similarity

Defined between two vectors v and u of equal length in [1] as:

$$J_{\mathcal{W}}(v,u) = \frac{\sum \min(v_i, u_i)}{\sum \max(v_i, u_i)}$$

Jaccard Distance v. Jaccard Similarity

These two metrics are inversely related. A polygon has Jaccard *distance* of zero from itself, but a Jaccard *similarity* of one. Both terms are used throughout this work.

Polygons to Sketches

Before being able to generate Locality Sensitive Hashes (LSH), a unique vector representation of each polygon is required. To generate this, center all polygons in the dataset and create a grid over them.

Figure 1. Sample Shape on a Grid

The sketch is a vector where each element is mapped to a cell on the grid. It's value is the percentage of the cell which is filled by the polygon.

Does this preserve the Jaccard Similarity?

The *weighted Jaccard similarity* between two of the sketch vectors approximates the Jaccard similarity of the original polygons.

Consider the contribution of a single cell of the grid to the Jaccard Similarity between a red polygon and a blue one. The minimal and maximal cases are depicted below.

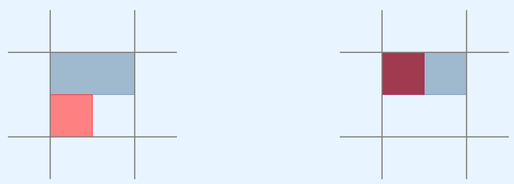

Figure 2. Miminal Jaccard contribution on the left, maximal Jaccard contribution on the right

Notice that:

- Area of intersection cannot exceed smaller polygon fragment.
- Area of union cannot be less than larger polygon fragment.

This causes the weighted Jaccard similarity of the sketches to be a maximizing approximation of the original Jaccard similarity.

Sketches to Locality Sensitive Hashes

The sketches are hashed to a collection of min-hashes as in [1].

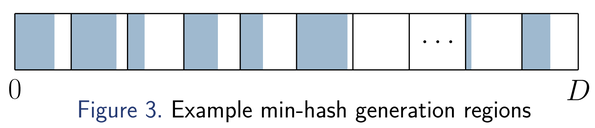

Figure 3. Example min-hash generation regions

Each cell above represents one element of the sketch and is filled to the value of that element in the sketch. Generate random numbers over this range until one lands inside the shaded region, the number of attempts to do so is the min-hash.

The probability that a pair of min-hashes are the same is the Jaccard Similarity of the sketches. Thus low Jaccard distance (high similarity) shapes generate the same hash a high proportion of the time.

Example Query

Figure 4. Returned nearest neighbors and actual nearest neighbors for an object in [2].

Implementation Performance

An implementation using the GEOS library [3] was also developed during this project written in C++20. Multi-threaded stability issues limited current testing to small datasets.

Grid Size	Thres.	# Maps	Hash Len.	Recall	MSE
25×25	3	15	5	0.2184	0.0420

Table 1. Best performance parameters for small test.

Tested with 373 reference polygons and 29 query shapes from [2]. Grid cells with support less than 3 were removed from consideration.

While the recall is low, the mean squared error of the neighbors that are found is also low, suggesting that this dataset has many neighbors which are nearly the same distance from each query.

References

[1] A. Shrivastava, "Simple and Efficient Weighted Minwise Hashing," in *Advances in Neural Information Processing Systems*, vol. 29, Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper/2016/hash/c2626d850c80ea07e7511bbae4c76f4b-Abstract.html (visited on 06/24/2022).

[2] *All water areas in the world from OpenStreetMap*, in collab. with A. Eldawy and M. F. Mokbel, 2019. [Online]. Available: https://doi.org/10.6086/N1668B70#mbr=00bm,zzpg.

[3] GEOS contributors, *GEOS computional geometry library*, version 3.11, Open Source Geospatial Foundation, 2022. [Online]. Available: https://libgeos.org/.

5 August 2022

This material is based upon work supported by the National Science Foundation under Grant #1950826.