3.6 The Theory of Locality-Sensitive Functions

Locality-sensitive hashing (LSH) is an algorithm that hashes similar inputs into the same buckets with high probability. The objective of LSH is to map high dimensional points into a lower dimensional space in such a way that points that are close to each other.

Besides the minhash functions, this section 3.6 discusses about some other families of functions that can produce candidate pairs efficiently.

There Locality-Sensitive Functions have three conditions in common:

1. They must be more possible to choose close pairs as candidate pairs.
2. They must be statistically independent.
3. They must be efficient to

(a) identify candidate pairs in time much less than the time it takes to look at all pairs.

(b) be combinable to avoid false positives and false negatives.

These functions can apply to the space of sets and the Jaccard distance, or to another space and/or another distance measure.

## 3.6.1 Locality-Sensitive Functions definition

In situation that we don’t want to check all n items, but we want to somehow figure out explore whether two items are similar/ might be close to each other that we could looked like a candidate pair.

The function F needs to be a good hashing function to hash items, and the equality of the results relate to decision of pairs

f(x) = f(y) demonstrates that x and y is a candidate pair.

d(x,y) is the distance between x and y

d1, d2 are two distances metric, where d1 < d2.

P1, p2 are two possibilities that between 0 and 1, and p1< p2

Compare d(x,y) with d1 and d2,

LSF states that a family of functions F is said to be a (d1, d2, p1, p2) locally sensitive family:

1. If d(x,y) ≤ d1 probability of f(x) = f(y) >= p1.

2. If d(x,y) ≥ d2, probability of f(x) = f(y) is <= p2.

The (d1, d2, p1, p2)-sensitive function show as follows:

Line chart

Description automatically generated with medium confidence

3.6.2 Locality-Sensitive Families for Jaccard Distance

We will use minhash functions and d is the Jaccard distance to find a family of Locality-Sensitive functions.

We interpret a minhash function h, if and only if h(x) = h(y), x and y is a candidate pair.

P1 -> 1-d1, p2 -> 1-d2, where0<=d1 <d2 <=1.

SIM(x, y) = 1 − d(x, y) ≥ 1 − d1.

Therefore, we could say the family of minhash functions is a (d1, d2, 1-d1, 1-d2)-sensitive family.

3.6.3 Amplification of Locality-Sensitive Families

A family of Locality-Sensitive functions F as defined above (satisfying p1 > p2) can be combined using AND and OR operations to produce new functions that approach this ideal LSH function. Given a (d1,d2,p1,p2){\displaystyle (d\_{1},d\_{2},p\_{1},p\_{2})}()-sensitive family F{\displaystyle {\mathcal {F}}}FF, we can construct new families {\displaystyle {\mathcal {G}}}by either the AND-construction or OR-construction of F.{\displaystyle {\mathcal {F}}}..

## The AND operation

Given a (d1,d2,p1,p2) sensitive family F, we say F′ is a (d1,d2,(p1)^r,(p2)^r)-sensitive family.

Where r functions h1, h2, . . . hr without replacement from F. Thus, while reducing the probability of a collision, AND amplifies the difference in probabilities of collisions between nearby and far points.

## The OR operation

The OR construction turns a (d1, d2, p1, p2)-sensitive family F into a (d1, d2, 1 − (1 − p1)^b, 1 − (1 − p2)^b)- sensitive family F′.

The OR operation boosts the chances of a collision in F’. Thus while boosting the probability of a collision, OR also boosts the probability of collision more nearby points than for points farther away, and thus is also an amplifying operation for hash functions.

## Concatenation of AND and OR

Concatenation of AND and OR can be used to combine hash functions to produce amplified hash functions that are near-ideal. The following equation shows a family of hash functions H with collision probabilities p transforms after AND and OR in sequence.