More LSH Families

Cosine Distance and Random Hyperplanes
Euclidean Distance

Mining of Massive Datasets
Leskovec, Rajaraman, and Ullman
Stanford University

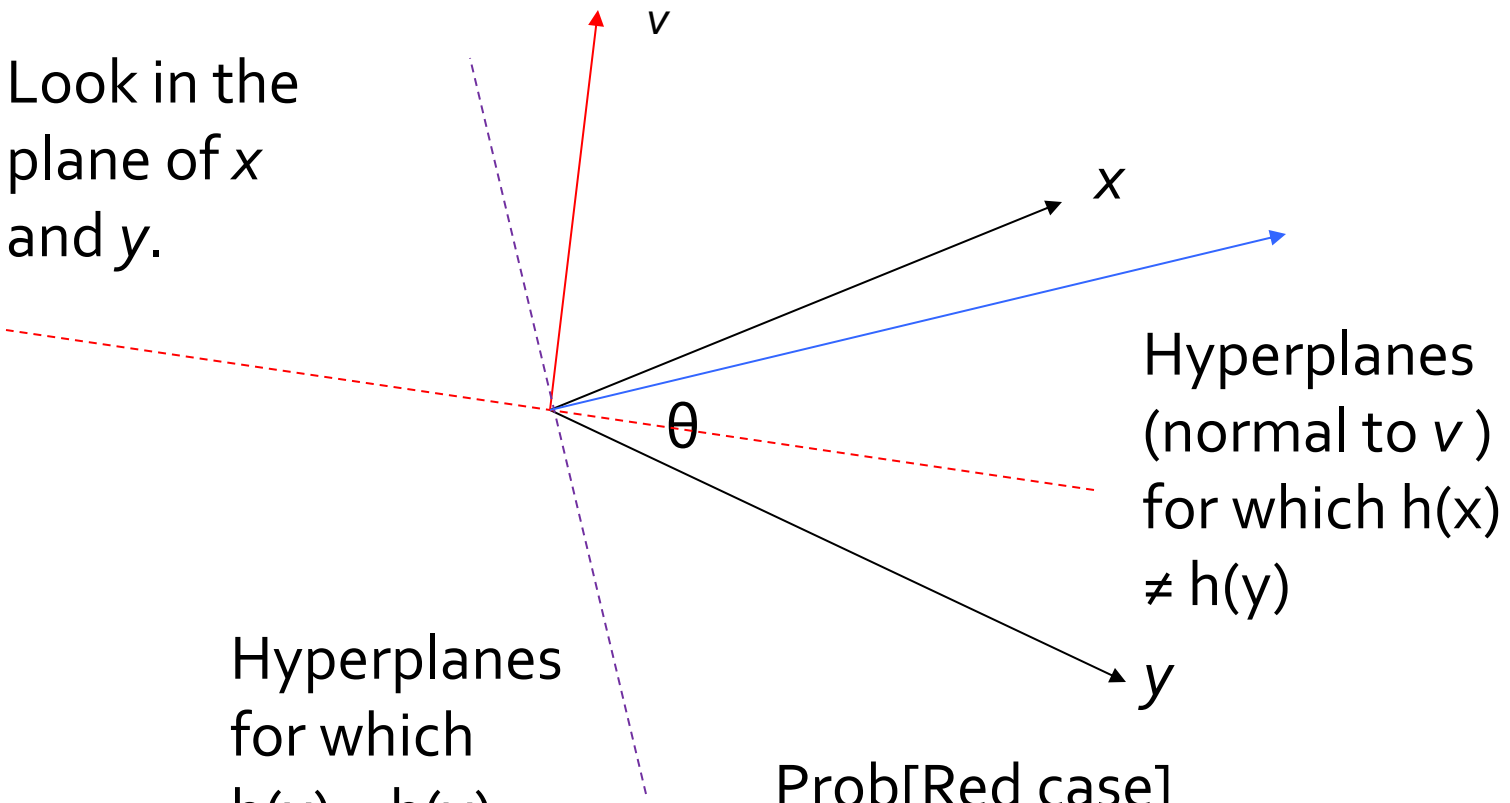
An LSH Family for Cosine Distance

- For cosine distance, there is a technique analogous to minhashing for generating a $(d_1,d_2,(1-d_1/180),(1-d_2/180))$-sensitive family for any d_1 and d_2.
- Called *random hyperplanes*.

Random Hyperplanes

- Each vector *v* determines a hash function h_v with two buckets.
- $h_v(x) = +1$ if $v.x > 0$; $= -1$ if $v.x < 0$.
- LS-family **H** = set of all functions derived from any vector.
- Claim: Prob[h(x)=h(y)] = 1 – (angle between *x* and *y* divided by 180).

Proof of Claim



Signatures for Cosine Distance

- Pick some number of vectors, and hash your data for each vector.
- The result is a signature (*sketch*) of +1's and –1's that can be used for LSH like the minhash signatures for Jaccard distance.
- But you don't have to think this way.
- The existence of the LSH-family is sufficient for amplification by AND/OR.

Simplification

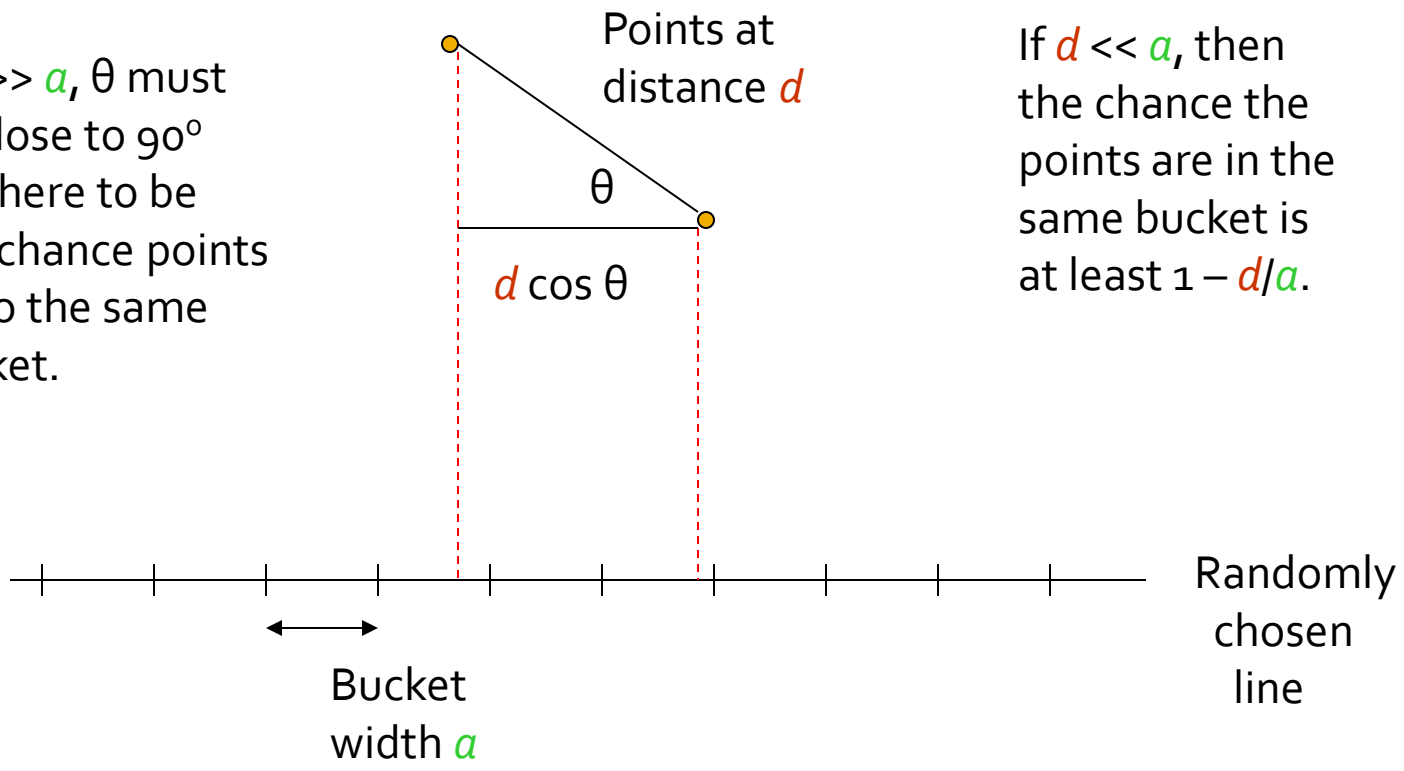
- We need not pick from among all possible vectors v to form a component of a sketch.
- It suffices to consider only vectors v consisting of +1 and –1 components.

LSH for Euclidean Distance

- Simple idea: hash functions correspond to lines.
- Partition the line into buckets of size a.
- Hash each point to the bucket containing its projection onto the line.
- Nearby points are always close; distant points are rarely in same bucket.

Projection of Points

If $d >> a$, θ must be close to 90° for there to be any chance points go to the same bucket.



If $d << a$, then the chance the points are in the same bucket is at least $1 - d/a$.

An LS-Family for Euclidean Distance

- If points are distance $\geq 2a$ apart, then $60 \leq \theta \leq 90$ for there to be a chance that the points go in the same bucket.
  - I.e., at most 1/3 probability.
- If points are distance $\leq a/2$, then there is at least ½ chance they share a bucket.
- Yields a $(a/2, 2a, 1/2, 1/3)$-sensitive family of hash functions.

Fixup: Euclidean Distance

- For previous distance measures, we could start with a (d, e, p, q)-sensitive family for any $d < e$, and drive p and q to 1 and 0 by AND/OR constructions.
- Here, we seem to need $e \geq 4d$.

Fixup – (2)

- But as long as $d < e$, the probability of points at distance d falling in the same bucket is greater than the probability of points at distance e doing so.
- Thus, the hash family formed by projecting onto lines is a (d, e, p, q)-sensitive family for some $p > q$.