

Lecture 4: Nearest Neighbor Search: RAG

Instructors: *Alex Andoni, Edo Liberty*Scribes: *Zoga Duka (zd2377)*

1 Overview

This lecture opens the nearest neighbor search (NNS) part of the course. The motivating problem comes from the data stage of the LLM pipeline: given a large corpus of documents, find all pairs of documents that are near duplicates. The same primitive, “find the items closest to a query”, is the core of retrieval (RAG), so we treat it as a computational problem in its own right.

The lecture develops the following chain of ideas.

1. Represent documents as sets and measure similarity by the Jaccard index.
2. Compress each set into a short *sketch* (minhash) from which Jaccard similarity can be estimated; this reduces near-duplicate detection to comparing n^2 short sketches, in time $\Theta(n^2k)$ for sketches of length k .
3. Ask how small k can be (Chebyshev gives $k = \Theta(1/(\varepsilon^2\delta))$, Hoeffding gives $k = \Theta(\varepsilon^{-2} \log(1/\delta))$), and whether the n^2 can be beaten (not exactly, assuming SETH).
4. Reformulate the task as a data-structure problem, r -near neighbor search, and relax it to the c -approximate version.
5. Introduce locality-sensitive hashing (LSH) as the tool for approximate NNS, analyze an idealized version, and state the Indyk–Motwani theorem, whose proof is the subject of the next lecture.

2 Near-duplicate detection

2.1 The problem

Problem 1 (Near-duplicate detection). *Given a corpus C of n documents, find all pairs of documents that are near matches.*

To make this precise we need a notion of similarity between documents. The lecture’s choice is to forget the order of words and view a document as a set: a document A becomes a subset of a universe $[U] = \{1, \dots, U\}$ of tokens (words, or w -grams of words). The reason is pragmatic: near duplicates typically share most of their tokens even when small edits move words around, and sets admit a very simple similarity measure.

Definition 2 (Jaccard similarity). *For nonempty $A, B \subseteq [U]$,*

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \in [0, 1].$$

$J(A, B) = 1$ iff $A = B$, and $J(A, B) = 0$ iff A and B are disjoint. We call $1 - J(A, B)$ the *Jaccard distance*; it is a metric on subsets of $[U]$, which is what allows us to phrase everything below in the language of metric spaces.

Computing $J(A, B)$ directly costs time $O(|A| + |B|)$ per pair (with a hash set), so the trivial algorithm for near-duplicate detection costs $O(n^2 \cdot \max_A |A|)$. The first goal of the lecture is to reduce the per-pair cost to something that does not depend on the document size.

2.2 Minhash

Definition 3 (Minhash [1]). Let $\pi : [U] \rightarrow [U]$ be a uniformly random permutation. The minhash of a nonempty set $A \subseteq [U]$ is

$$h(A) = h_\pi(A) = \min_{x \in A} \pi(x).$$

Thus $h_\pi : 2^{[U]} \setminus \{\emptyset\} \rightarrow [U]$ maps a set to a single element.

The whole construction rests on the following property, which was stated in class; we include the proof.

Claim 4. For all nonempty $A, B \subseteq [U]$, $\Pr_\pi [h_\pi(A) = h_\pi(B)] = J(A, B)$.

Proof. Let $x^* = \arg \min_{x \in A \cup B} \pi(x)$ be the element of $A \cup B$ with the smallest image under π . If $x^* \in A \cap B$ then $h_\pi(A) = h_\pi(B) = \pi(x^*)$. If $x^* \in A \setminus B$, then $h_\pi(A) = \pi(x^*)$ while $h_\pi(B) = \min_{x \in B} \pi(x) > \pi(x^*)$, because every $x \in B$ is an element of $A \cup B$ other than x^* ; the case $x^* \in B \setminus A$ is symmetric. Hence $h_\pi(A) = h_\pi(B)$ iff $x^* \in A \cap B$. Since π is a uniformly random permutation, x^* is a uniformly random element of $A \cup B$, so the probability is $|A \cap B| / |A \cup B|$. $\square$

So a single random minhash is a coin whose bias is exactly the Jaccard similarity. This is the sense in which minhash “sketches” J .

2.3 Sketching

Definition 5 (Sketch). A sketch for Jaccard similarity is a (randomized) map $S : 2^{[U]} \rightarrow \{0, 1\}^s$ together with an estimator $R : \{0, 1\}^s \times \{0, 1\}^s \rightarrow [0, 1]$ such that for all A, B ,

$$R(S(A), S(B)) \approx J(A, B).$$

The point of a sketch is *functional compression*: we do not need to recover A from $S(A)$, only to compute one function of pairs. Each document is sketched once, and afterwards every pairwise comparison touches only $2s$ bits.

Definition 6 (Minhash sketch). Fix $k \in \mathbb{N}$ and independent uniformly random permutations $\pi_1, \dots, \pi_k$ of $[U]$ with minhashes $h_1, \dots, h_k$. The sketch of A is

$$S(A) = (h_1(A), h_2(A), \dots, h_k(A)) \in [U]^k,$$

and the estimator is the fraction of coordinates that agree:

$$\hat{J}(A, B) = \frac{1}{k} \sum_{i=1}^k \mathbb{I}[h_i(A) = h_i(B)].$$

By Claim 4 and linearity of expectation, $\mathbb{E}[\hat{J}(A, B)] = J(A, B)$: the estimator is unbiased. The sketch occupies $s = k \lceil \log_2 U \rceil$ bits.

Algorithm 7 (Near-duplicate detection via sketches). Given a similarity threshold τ :

1. For every $A \in C$ compute $S(A)$. This takes time $O(k \cdot |A|)$ per document, i.e. $O(k \sum_{A \in C} |A|)$ in total.
2. For every pair $A, B \in C$ compute $\hat{J}(A, B)$ in time $O(k)$ and report the pair if $\hat{J}(A, B) \geq \tau$. This takes time $\Theta(n^2 k)$.

Step 2 dominates, and its cost is n^2 times the sketch length. So the question is how small k can be.

2.4 How large should k be?

We want, for a given accuracy ε and failure probability δ ,

$$\Pr \left[|\hat{J}(A, B) - J(A, B)| \leq \varepsilon \right] \geq 1 - \delta \quad \text{for all } A, B. \quad (1)$$

The lecture gave a first bound with Chebyshev's inequality and then observed that a concentration bound gives an exponentially better dependence on δ . We do both; the reason for starting with Chebyshev is that it only needs the variance, and the variance computation is where the independence of the h_i enters.

Lemma 8 (Chebyshev). *For any random variable X with finite variance and any $\lambda > 0$, $\Pr \left[|X - \mathbb{E}[X]| \geq \lambda \right] \leq \text{Var}(X)/\lambda^2$.*

Theorem 9. *Let $k \geq \frac{1}{\varepsilon^2 \delta}$. Then (1) holds.*

Proof. Fix A, B and write $J = J(A, B)$. Let $X_i = \mathbb{1}[h_i(A) = h_i(B)]$, so $\hat{J} = \frac{1}{k} \sum_{i=1}^k X_i$ and $\mathbb{E}[X_i] = \mathbb{E}[\hat{J}] = J$ by Claim 4. Since $\hat{J}$ has mean J , Chebyshev with $\lambda = \varepsilon$ gives $\Pr[|\hat{J} - J| \geq \varepsilon] \leq \text{Var}(\hat{J})/\varepsilon^2$, so it suffices to show $\text{Var}(\hat{J}) \leq \varepsilon^2 \delta$.

The X_i are independent, because the permutations π_i are. Hence

$$\text{Var}(\hat{J}) = \frac{1}{k^2} \text{Var} \left(\sum_{i=1}^k X_i \right) = \frac{1}{k^2} \sum_{i=1}^k \text{Var}(X_i) \leq \frac{1}{k^2} \sum_{i=1}^k \mathbb{E}[X_i^2] = \frac{1}{k} (J \cdot 1^2 + (1-J) \cdot 0^2) = \frac{J}{k} \leq \frac{1}{k} \leq \varepsilon^2 \delta. \quad \square$$

The dependence on δ is $1/\delta$, which is poor: driving the failure probability to n^{-10} , say, costs a factor n^{10} in k . Replacing Chebyshev by a Chernoff-type bound fixes this.

Lemma 10 (Hoeffding). *Let $X_1, \dots, X_k$ be independent random variables with values in $[0, 1]$ and common mean μ . Then for every $\varepsilon > 0$, $\Pr \left[\left| \frac{1}{k} \sum_{i=1}^k X_i - \mu \right| \geq \varepsilon \right] \leq 2e^{-2k\varepsilon^2}$.*

Theorem 11. *Let $k \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$. Then (1) holds.*

Proof. Apply Hoeffding to the independent indicators $X_i = \mathbb{1}[h_i(A) = h_i(B)]$, which have mean $\mu = J(A, B)$: the failure probability is at most $2e^{-2k\varepsilon^2} \leq \delta$ for the stated k . $\square$

Remark 12 (All pairs at once). *Both theorems concern one fixed pair. Algorithm 7 uses the sketches for all $\binom{n}{2}$ pairs; by a union bound over the pairs, Theorem 11 with $\delta/\binom{n}{2}$ in place of δ , i.e. $k = \Theta(\varepsilon^{-2} \log(n/\delta))$, makes every estimate ε -accurate simultaneously with probability $1 - \delta$. With Theorem 9 the same union bound would require $k = \Theta(n^2/(\varepsilon^2 \delta))$, which is why the Chernoff-type bound matters here.*

Remark 13 (Sketch size). *The sketch has $s = k \lceil \log_2 U \rceil$ bits. This can be reduced to $s = O(k \log(1/\varepsilon))$: replace each $h_i(A)$ by $g(h_i(A))$ for a random hash $g : [U] \rightarrow \{0, 1\}^b$ with $b = \lceil \log_2(1/\varepsilon) \rceil$. Two unequal minhash values then collide with probability $2^{-b} \leq \varepsilon$, so each coordinate agrees with probability between J and $J + \varepsilon$, and the estimator's error grows by at most ε .*

2.5 Can the n^2 be avoided?

With $k = \Theta(\varepsilon^{-2} \log(n/\delta))$, Algorithm 7 runs in time $\tilde{O}(n^2)$, and the sketching step is linear in the input. The natural question is whether the quadratic comparison step can be avoided. The answer given in class is: not for the *exact* problem, conditioned on the following hypothesis from fine-grained complexity.

Assumption 14 (Strong Exponential Time Hypothesis, SETH [6]). *For every $\varepsilon > 0$ there is a k such that k -SAT on n variables cannot be solved in time $O((2 - \varepsilon)^n)$. (In particular there is no 1.999^n -time algorithm for CNF-SAT.)*

Under SETH, no algorithm decides in time $n^{2-\varepsilon}$, given two collections $\mathcal{A}, \mathcal{B}$ of n documents each with $\omega(\log n)$ distinct tokens per document, whether there is a pair $A \in \mathcal{A}, B \in \mathcal{B}$ of Jaccard similarity at least $1/2$. This is the *bichromatic* version of the problem; the single-corpus version is not directly covered by the argument (see Remark 31). The precise statement and its proof are the subject of Section 4. The consequence for the design of algorithms is that subquadratic (and, in the data-structure setting, sublinear) running times require *approximation*, which is the theme of the rest of the lecture.

3 Nearest neighbor search

3.1 Problem definitions

We now fix a metric space (M, d) . The two examples in the lecture are subsets of $[U]$ with the Jaccard distance $d = 1 - J$, and Euclidean space $\mathbb{R}^d$ with $\|\cdot\|_2$. (We use d both for the metric and for the dimension; the meaning is always clear from context.)

Problem 15 (r -near neighbor). *Given $r > 0$, preprocess a dataset $D \subseteq M$ of n points so that, given a query $q \in M$, one can find a point $p \in D$ with $d(q, p) \leq r$, if one exists.*

Remark 16. *Near-duplicate detection reduces to r -near neighbor search in two steps: preprocess the corpus C , then query the structure with each $q \in C$. The point of the reformulation is that a good data structure gives the near-dedup problem for free, while also solving the online retrieval problem.*

The naive solution scans the dataset: query time $O(n \cdot d)$ in $\mathbb{R}^d$, and $O(n^2 d)$ for near-dedup, i.e. nothing is gained. The goal is query time $\ll n$. In view of Section 2.5 this is impossible for the exact problem, so we relax it.

Problem 17 (c -approximate r -near neighbor). *Given $c > 1$ and $r > 0$, preprocess D so that, given $q \in M$: if there is $p \in D$ with $d(q, p) \leq r$, report some $p' \in D$ with $d(q, p') \leq cr$.*

The structure may return a point that is a little farther than the nearest one, by a factor c (Figure 1). If no point is within distance r the structure may answer anything. The relaxation is natural in applications, where r is itself a heuristic threshold, and it is exactly what makes sublinear query time possible.

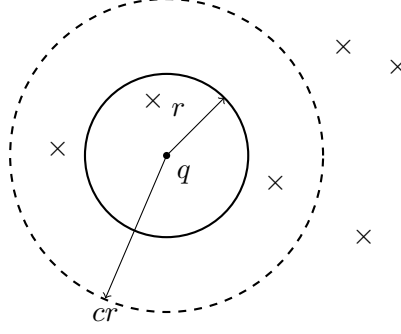


Figure 1: A query q with a point inside the ball of radius r . Any of the three points inside the dashed ball of radius cr is an acceptable answer; the three points outside it are not.

3.2 Locality-sensitive hashing

The idea of LSH [2] is to hash points so that close points collide with noticeably higher probability than far points; then the bucket of $h(q)$ is a good place to look for near neighbors.

Definition 18 (Locality-sensitive family). *A family $\mathcal{H}$ of functions $h : M \rightarrow [U]$ is (r, cr, p_1, p_2) -sensitive if, for h drawn uniformly from $\mathcal{H}$ and all $p, q \in M$:*

1. if $d(p, q) \leq r$ then $\Pr_h[h(p) = h(q)] \geq p_1$;
2. if $d(p, q) > cr$ then $\Pr_h[h(p) = h(q)] \leq p_2$.

The family is useful only when $p_1 > p_2$.

Claim 19. *The minhash family $\{h_\pi\}$ is $(r, cr, 1 - r, 1 - cr)$ -sensitive for the Jaccard distance, for every $0 < r < cr < 1$.*

Proof. By Claim 4, $\Pr[h_\pi(A) = h_\pi(B)] = J(A, B) = 1 - d(A, B)$. If $d(A, B) \leq r$ this is at least $1 - r$; if $d(A, B) > cr$ it is less than $1 - cr$. $\square$

To see why locality-sensitive families lead to data structures, consider first an idealized family.

Algorithm 20 (Basic LSH structure). Let $\mathcal{H}$ be (r, cr, p_1, p_2) -sensitive.

- *Preprocessing.* Pick $h \in \mathcal{H}$ at random; compute $h(p)$ for every $p \in D$ and store the points in a hash map keyed by $h(p)$.
- *Query q .* Compute $h(q)$, retrieve the bucket of all $p \in D$ with $h(p) = h(q)$, and scan it, computing $d(q, p)$ for each point, until a point with $d(q, p) \leq cr$ is found; return it. If the bucket is exhausted, return “none”.

Theorem 21. *Suppose $\mathcal{H}$ is $(r, cr, 1, 0)$ -sensitive. Then Algorithm 20 solves the c -approximate r -near neighbor problem with probability 1, uses space $O(nd)$ (the points) plus $O(n)$ (the hash map), and answers a query with one evaluation of h and at most one distance computation.*

Proof. Correctness. The algorithm can fail only if some $p^* \in D$ has $d(q, p^*) \leq r$ but the scan does not return a point within cr . Since $p_1 = 1$, $h(p^*) = h(q)$ with probability 1, so p^* is in the bucket; the scan stops at the first point within distance cr , and p^* is such a point, so the scan returns a valid answer.

Query time. The scan computes the distance to each far point ($d(q, p) > cr$) in the bucket, plus at most one distance to the point at which it stops. Let F be the number of far points in the bucket. By linearity of expectation, $\mathbb{E}[F] \leq p_2 \cdot n$, which is 0 for $p_2 = 0$; so $F = 0$ almost surely and at most one distance is computed. $\square$

Unfortunately such families do not exist in the spaces we care about.

Fact 22. *For $M = \mathbb{R}^d$ with the Euclidean metric and any $c > 1$, there is no $(r, cr, 1, 0)$ -sensitive family.*

Proof. Let $m = \lceil c \rceil + 1$ and consider the points $p_j = q + j r e_1$ for $j = 0, \dots, m$. Consecutive points are at distance r , so $p_1 = 1$ forces $h(p_j) = h(p_{j+1})$ with probability 1 for each j , hence $h(p_0) = h(p_m)$ with probability 1. But $d(p_0, p_m) = mr > cr$, so $p_2 = 0$ forces $\Pr[h(p_0) = h(p_m)] = 0$, a contradiction. $\square$

For minhash the gap is visible directly: $p_1 = 1 - r < 1$ and $p_2 = 1 - cr > 0$. In general we have $p_1 < 1$ and $p_2 > 0$, and the analysis of Algorithm 20 becomes: the query succeeds with probability at least p_1 , and the expected number of far points examined is at most $p_2 n$. Neither is good enough on its own; the next lecture shows how to *amplify* the gap between p_1 and p_2 (concatenating hash functions to shrink p_2 , and using several independent tables to restore the success probability), which yields the following theorem.

Theorem 23 (Indyk–Motwani [2]). *Let $\mathcal{H}$ be an (r, cr, p_1, p_2) -sensitive family with $p_1 > p_2$, and let*

$$\rho = \frac{\log p_1}{\log p_2} = \frac{\log(1/p_1)}{\log(1/p_2)} \in (0, 1).$$

Then there is a data structure for the c -approximate r -near neighbor problem with space $O(n^{1+\rho} + nd)$ and query time $O(n^\rho \cdot d)$, up to logarithmic factors, that succeeds with constant probability.

The exponent ρ is the whole story: it is determined by the family alone, and lowering it is what research on LSH is about. For Euclidean space the known values are

$$\rho = \frac{1}{c} \text{ (e.g. } \rho = \frac{1}{2} \text{ for } c = 2) \text{ [2, 3],} \quad \rho = \frac{1}{c^2} + o(1) \text{ [4],}$$

and the latter is optimal for LSH-based methods [5].

4 SETH-hardness of exact near-duplicate detection [not covered in class]

This chapter fills in the one-line remark from the lecture that time $\ll n^2$ is impossible for exact near-duplicate detection under SETH. We make the statement precise and prove it, and then explain why it is exactly approximation that lets LSH escape the barrier.

4.1 Fine-grained hypotheses

Throughout, $A, B \subseteq \{0, 1\}^d$ are multisets of N vectors each, and $\langle a, b \rangle = \sum_i a_i b_i$ is the number of coordinates where both are 1.

Definition 24 (Orthogonal Vectors, OV). *Given $A, B \subseteq \{0, 1\}^d$ with $|A| = |B| = N$, decide whether there exist $a \in A$, $b \in B$ with $\langle a, b \rangle = 0$.*

The trivial algorithm tries all N^2 pairs in time $O(N^2 d)$. The following theorem says that, for dimension only slightly above $\log N$, nothing substantially better is possible under SETH. We use the sparsification lemma of Impagliazzo, Paturi and Zane [7]: for every k and every $\eta > 0$, a k -CNF on n variables can be written, in time $O(2^{\eta n})$, as the OR of $2^{\eta n}$ k -CNFs on the same variables, each having at most $c_{k,\eta} \cdot n$ clauses for a constant $c_{k,\eta}$.

Theorem 25 (Williams [8]). *Assume SETH. Then for every $\varepsilon > 0$ and every function $d(N) = \omega(\log N)$, OV in dimension $d(N)$ cannot be solved in time $O(N^{2-\varepsilon})$.*

Proof. Suppose OV in dimension $d(N) = \omega(\log N)$ is solvable in time $O(N^{2-\varepsilon})$. Fix k ; we solve k -SAT in time $O(2^{(1-\varepsilon/3)n})$, which contradicts SETH once k is the value SETH provides for $\varepsilon/3$.

Let φ be a k -CNF on variables $x_1, \dots, x_n$. Apply the sparsification lemma with $\eta = \varepsilon/6$: it suffices to decide satisfiability of each of the $2^{\eta n}$ sparse formulas $\varphi_1, \dots, \varphi_{2^{\eta n}}$, each with $m \leq cn$ clauses ($c = c_{k,\eta}$).

Fix one sparse formula ψ with clauses $C_1, \dots, C_m$. Split the variables into $V_1 = \{x_1, \dots, x_{n/2}\}$ and $V_2 = \{x_{n/2+1}, \dots, x_n\}$. For every assignment $\alpha \in \{0, 1\}^{V_1}$ define $u_\alpha \in \{0, 1\}^m$ by

$$u_\alpha[j] = 1 \iff \text{no literal of } C_j \text{ over } V_1 \text{ is satisfied by } \alpha,$$

and define $v_\beta \in \{0, 1\}^m$ for $\beta \in \{0, 1\}^{V_2}$ symmetrically. Then $u_\alpha[j] v_\beta[j] = 1$ iff clause C_j is satisfied by neither half, i.e. iff the combined assignment (α, β) falsifies C_j . Hence

$$\langle u_\alpha, v_\beta \rangle = 0 \iff (\alpha, \beta) \text{ satisfies } \psi.$$

So ψ is satisfiable iff the OV instance $A = \{u_\alpha\}$, $B = \{v_\beta\}$ has an orthogonal pair. Here $N = 2^{n/2}$ and the dimension is $m \leq cn = 2c \log_2 N$. Since $d(N) = \omega(\log N)$, for all large N we have $m \leq d(N)$, and we pad the vectors with zeros (which does not change inner products). Building the instance takes $O(Nm)$ time and solving it takes $O(N^{2-\varepsilon}) = O(2^{(1-\varepsilon/2)n})$. Over all $2^{\eta n}$ sparse formulas the total time is $O(2^{(\eta+1-\varepsilon/2)n}) = O(2^{(1-\varepsilon/3)n})$, as claimed. $\square$

The conclusion of Theorem 25 is called the *Orthogonal Vectors Conjecture* (OVC). Note why the dimension must be $\omega(\log N)$ rather than $C \log N$ for a fixed C : the constant $c_{k,\eta}$ grows with k , and SETH only gives hardness for larger and larger k .

4.2 From OV to near-duplicates

We now show that OV embeds into the exact version of the problem from the lecture. To keep the statement clean we use the *bichromatic* variant, in which the pairs to be found go across two collections; the two-step reduction of near-dedup to r -near neighbor search (preprocess, then query each point) is this variant.

Definition 26 (Exact bichromatic near-pair). *Given two collections $\mathcal{A}, \mathcal{B}$ of n subsets of a universe $[U]$ and a threshold $\tau \in (0, 1]$, decide whether there exist $A \in \mathcal{A}, B \in \mathcal{B}$ with $J(A, B) \geq \tau$ (equivalently, Jaccard distance $1 - J(A, B) \leq 1 - \tau$).*

Theorem 27. *Assume SETH. For every $\varepsilon > 0$ and every $U(n) = \omega(\log n)$, the exact bichromatic near-pair problem with threshold $\tau = 1/2$ over universe $[U(n)]$ cannot be solved in time $O(n^{2-\varepsilon})$. The same holds for exact bichromatic closest pair in Hamming distance on $\{0, 1\}^{U(n)}$.*

Note that Theorem 27 is a statement about the bichromatic problem only (two separate collections). It does not directly give hardness for the single-corpus (monochromatic) formulation of near-duplicate detection; see Remark 31.

Proof. Let (A, B) be an OV instance in dimension $d = \omega(\log N)$ with $|A| = |B| = N$. For a vector x write $\bar{x}$ for its bitwise complement and $|x|$ for its Hamming weight. Define maps into $\{0, 1\}^{3d}$ (three blocks of d coordinates each):

$$\phi(a) = (a, \bar{a}, 0^d), \quad \psi(b) = (\bar{b}, 1^d, b).$$

We compute weights and inner products block by block:

$$\begin{aligned} |\phi(a)| &= |a| + (d - |a|) + 0 = d, \\ |\psi(b)| &= (d - |b|) + d + |b| = 2d, \\ \langle \phi(a), \psi(b) \rangle &= \langle a, \bar{b} \rangle + \langle \bar{a}, 1^d \rangle + \langle 0^d, b \rangle = (|a| - \langle a, b \rangle) + (d - |a|) + 0 = d - \langle a, b \rangle. \end{aligned}$$

Writing $t = \langle a, b \rangle \in \{0, 1, \dots, d\}$, and viewing $\phi(a), \psi(b)$ as subsets of $[3d]$,

$$J(\phi(a), \psi(b)) = \frac{\langle \phi(a), \psi(b) \rangle}{|\phi(a)| + |\psi(b)| - \langle \phi(a), \psi(b) \rangle} = \frac{d - t}{3d - (d - t)} = \frac{d - t}{2d + t}.$$

This is a strictly decreasing function of t . It equals $\frac{1}{2}$ when $t = 0$ and is at most $\frac{d-1}{2d+1} < \frac{1}{2}$ when $t \geq 1$. Hence

$$\exists a \in A, b \in B : \langle a, b \rangle = 0 \iff \exists A' \in \phi(A), B' \in \psi(B) : J(A', B') \geq \frac{1}{2}.$$

The map is computable in $O(Nd)$ time, produces $n = N$ sets on each side over a universe of size $U = 3d = \omega(\log n)$, and $(\phi(A), \psi(B))$ is a valid input for any target universe size $U(n) \geq 3d$ (pad with unused elements). An $O(n^{2-\varepsilon})$ algorithm for the near-pair problem would therefore solve OV in time $O(N^{2-\varepsilon})$, contradicting Theorem 25.

For Hamming distance, the same gadget gives

$$\text{ham}(\phi(a), \psi(b)) = |\phi(a)| + |\psi(b)| - 2\langle \phi(a), \psi(b) \rangle = d + 2d - 2(d - t) = d + 2t,$$

so the bichromatic closest pair has distance exactly d if an orthogonal pair exists and at least $d + 2$ otherwise. $\square$

4.3 Consequences for the lecture

The lecture's goal was a data structure with query time $\ll n$. For the *exact* r -near neighbor problem this is impossible without a quadratic preprocessing cost:

Corollary 28. *Assume SETH and let $U(n) = \omega(\log n)$. There is no data structure for the exact r -near neighbor problem in Jaccard distance ($r = 1/2$) over sets in $[U(n)]$ with preprocessing time $O(n^{2-\varepsilon})$ and query time $O(n^{1-\varepsilon})$ for some $\varepsilon > 0$.*

Proof. Given such a structure, solve the bichromatic near-pair problem in two steps: preprocess $\psi(B)$ in $O(n^{2-\varepsilon})$ time and query it with each of the n sets $\phi(a)$, $a \in A$, in total $n \cdot O(n^{1-\varepsilon}) = O(n^{2-\varepsilon})$ time. Report yes iff some query returns a point at Jaccard distance $\leq 1/2$. This contradicts Theorem 27. $\square$

In particular Algorithm 7, which sketches and then compares all $\binom{n}{2}$ pairs in $\Theta(n^2 k)$ time, cannot be improved to $n^{2-\varepsilon}$ time by any *exact* method when documents contain $\omega(\log n)$ distinct tokens. The LSH structures of the next lecture achieve $n^{1+\rho}$ preprocessing and n^ρ query with $\rho < 1$, which by Corollary 28 is only possible because they solve the c -approximate problem.

4.4 Why approximation escapes the barrier

It is instructive to look at how small the gap in the reduction is. In Jaccard distance the two cases are

$$\text{yes: } 1 - \frac{1}{2} = \frac{1}{2}, \quad \text{no: } \geq 1 - \frac{d-1}{2d+1} = \frac{d+2}{2d+1} = \frac{1}{2} \cdot \frac{2d+4}{2d+1}.$$

So the reduction only distinguishes distance $r = \frac{1}{2}$ from distance cr with $c = 1 + \frac{3}{2d+1} = 1 + o(1)$, because $d = \omega(\log n)$. An algorithm for the c -approximate r -near neighbor problem with any constant $c > 1$ (such as LSH, where the gap is a fixed constant c and $\rho < 1$) is simply not addressed by Theorem 27: it may legitimately answer “yes” in the “no” case. This is the precise sense in which the lecture’s “no without approximation” should be read.

Two further remarks on where the true boundary lies.

Remark 29 (Approximation does not help arbitrarily). *Rubinstein [10] showed, using a considerably more involved gap amplification, that under SETH the hardness persists even for approximation: for every $\delta > 0$ there is a constant $\varepsilon > 0$ such that no $O(n^{2-\delta})$ -time algorithm computes a $(1 + \varepsilon)$ -approximate bichromatic closest pair in Hamming space of dimension $\omega(\log n)$. Thus the approximation factor c in LSH cannot be taken arbitrarily close to 1 while keeping a fixed polynomial saving; this matches the fact that $\rho \rightarrow 1$ as $c \rightarrow 1$ in all known LSH families.*

Remark 30 (Dimension is essential). *The lower bound needs $d = \omega(\log n)$. For $d = c \log n$ with c constant, Alman and Williams [9] gave an exact algorithm for Hamming closest pair running in time roughly $n^{2-1/O(c \log^2 c)}$, i.e. genuinely subquadratic, via the polynomial method. So exact near-duplicate detection is subquadratic when documents have $O(\log n)$ distinct tokens, and (conditionally) quadratic as soon as they have slightly more.*

Remark 31 (Monochromatic version). *Our reduction is bichromatic: sets of $\phi(A)$ may themselves be near-duplicates of each other (e.g. $J(\phi(a), \phi(a')) = \frac{d-h}{d+h}$ where $h = \text{ham}(a, a')$, which exceeds $\frac{1}{2}$ whenever $h < d/3$), so it does not directly apply to the single-corpus formulation of near-dedup. Karthik C. S. and Manurangsi [11] showed that the monochromatic closest-pair problem is nevertheless SETH-hard as well, in essentially the same dimension regime, via a more delicate gadget that keeps same-side points far apart.*

References

- [1] A. Z. Broder. On the resemblance and containment of documents. *Compression and Complexity of Sequences*, 1997.
- [2] P. Indyk, R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. *STOC*, 1998.
- [3] M. Datar, N. Immorlica, P. Indyk, V. Mirrokni. Locality-sensitive hashing scheme based on p -stable distributions. *SoCG*, 2004.
- [4] A. Andoni, P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *FOCS*, 2006.
- [5] R. O’Donnell, Y. Wu, Y. Zhou. Optimal lower bounds for locality-sensitive hashing (except when q is tiny). *ACM Trans. Comput. Theory*, 2014.
- [6] R. Impagliazzo, R. Paturi. On the complexity of k -SAT. *J. Comput. Syst. Sci.*, 2001.
- [7] R. Impagliazzo, R. Paturi, F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 2001.
- [8] R. Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 2005.
- [9] J. Alman, R. Williams. Probabilistic polynomials and Hamming nearest neighbors. *FOCS*, 2015.
- [10] A. Rubinfeld. Hardness of approximate nearest neighbor search. *STOC*, 2018.
- [11] Karthik C. S., P. Manurangsi. On closest pair in Euclidean metric: monochromatic is as hard as bichromatic. *ITCS*, 2019.