Lecture 10

TTIC 41000: Algorithms for Massive Data

Toyota Technological Institute at Chicago

Spring 2021

Instructor: Sepideh Mahabadi

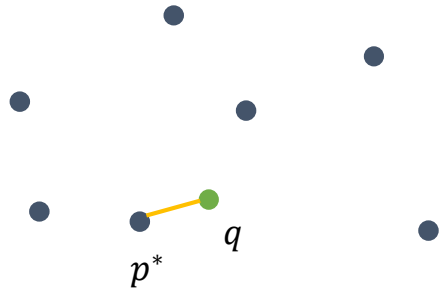
This Lecture

- Approximate Nearest Neighbor

Nearest Neighbor

Dataset of n points P in a metric space, e.g. $\mathbb{R}^d$

A query point q comes online



Goal:

- Find the nearest data point p^*
- Do it in sub-linear time and small space

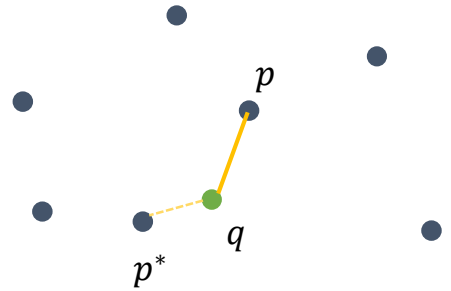
All existing algorithms for this problem

- Either space or query time depending exponentially on d
- Or assume certain properties about the data, e.g., bounded intrinsic dimension

Approximate Nearest Neighbor

Dataset of n points P in a metric space, e.g. $\mathbb{R}^d$

A query point q comes online

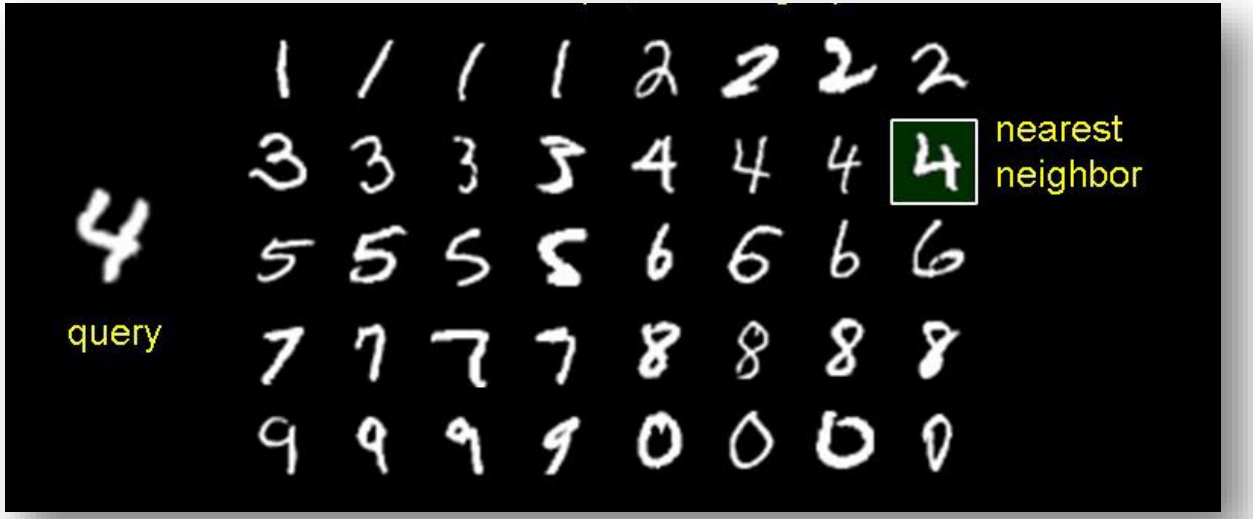


Goal:

- Find the nearest data point p^*
- Do it in sub-linear time and small space
- Approximate Nearest Neighbor
  - If optimal distance is r, report a point in distance cr for $c = (1+\epsilon)$
  - For Hamming (and Manhattan) query time is $n^{1/O(c)}$ [IM98]
  - and for Euclidean it is $n^{\frac{1}{O(c^2)}}$ [AI08]

Applications of NN

Searching for the closest object

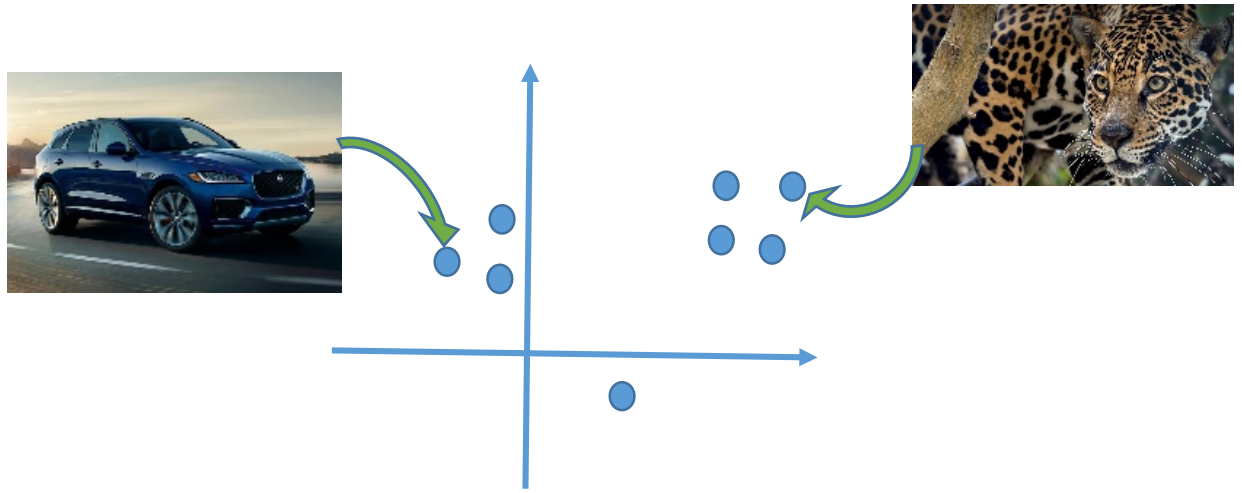


Modeling the Search Problem

Objects (documents, images, etc)

Feature Vectors

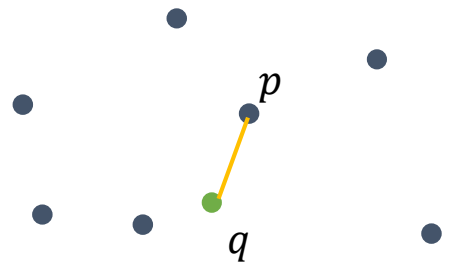
Points in a high dimensional space



Approximate Near Neighbor

Dataset of n points P in a metric space, e.g. $\mathbb{R}^d$, and a parameter r

A query point q comes online



Goal:

- If there exists a point within distance r
- Find one point within distance cr

- Approximate Nearest Neighbor can be reduced to polylog instances of Approximate Near Neighbor

Locality Sensitive Hashing (LSH)

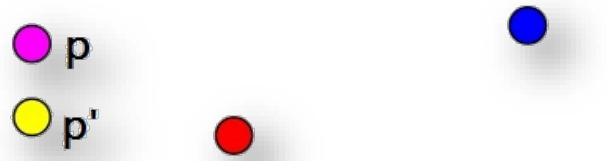
One of the main approaches to solve the Nearest Neighbor problems

Locality Sensitive Hashing (LSH) [Indyk, Motwani'98]

Hashing scheme s.t. close points have higher probability of collision than far points

Hash functions: $g_1, \dots, g_L$

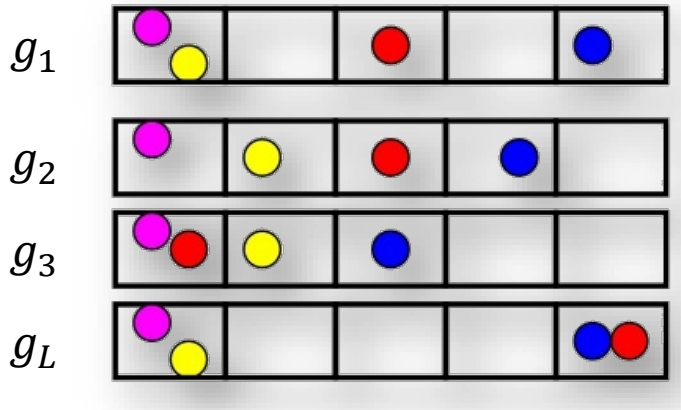
- g_i is an independently chosen hash function
- Concatenation of several randomly chosen hash functions from $\mathcal{H}$



If $||p - p'|| \leq r$, they collide w.p. $\geq P_{high}$

If $||p - p'|| \geq cr$, they collide w.p. $\leq P_{low}$

For $P_{high} \geq P_{low}$



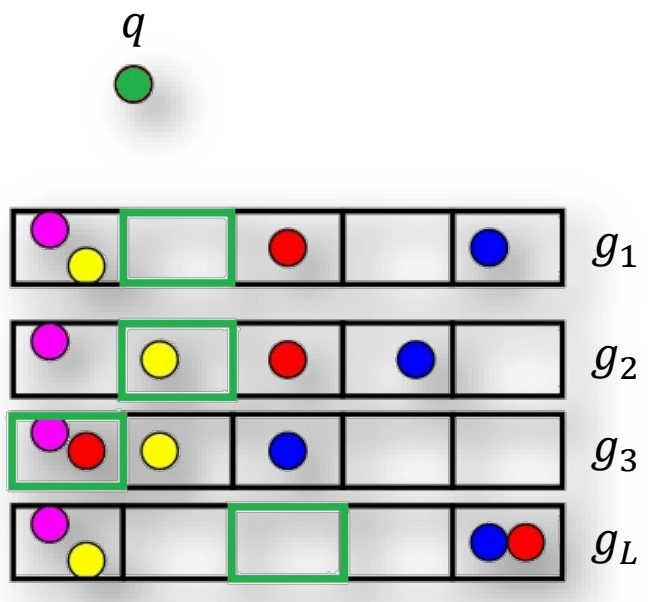
➢ $\mathcal{H}$ is a $(r, cr, p_{high}, p_{low})$ –sensitive family of Hash Functions

Locality Sensitive Hashing (LSH) [Indyk, Motwani'98]

Retrieval:

- The union of the query buckets is roughly the neighborhood of q

- $T = \bigcup_i B_i\left(g_{i(q)}\right)$ is roughly the neighborhood i.e., with a constant prob
  - $N(q, r) \subseteq T$
  - $|T \setminus N(q, cr)| \leq O(L)$



Details

- $k = \lceil \log_{1/p_{low}} n \rceil$
- $L = n^{\rho}/p_{high}$
- Let $\rho = \frac{\log 1/p_{high}}{\log 1/p_{low}}$

- Assume we have access to $\mathcal{H}$ which is is a $(r, cr, p_{high}, p_{low})$ −sensitive family of Hash Functions
- $\boldsymbol{L}$ is the number of hash functions: $g_1, \cdots, g_L$
- Each g_i is a concatenation of $\boldsymbol{k}$ randomly chosen functions from $\mathcal{H}$, e.g. $h_{i,1}, \cdots, h_{i,k}$ and $g_i(a) = g_i(b)$ iff $\forall j: h_{i,j}(a) = h_{i,j}(b)$
- If $dist(a,b) \leq r$: $\Pr[g_i(a) = g_i(b)] \geq (p_{high})^k$
  - $\Pr[g_i(a) = g_i(b)] = (p_{high})^k \geq (p_{high})^{k+1} = p_{high} \cdot n^{-\rho}$
  - $\Pr[\exists i:\ g_i(a) = g_i(b)] \geq 1 - \left(1 - (p_{high})^k\right)^L \geq 1 - \left(1 - p_{high} \cdot n^{-\rho}\right)^L \geq 1 - 1/e$
- If $dist(a,b) > cr$: $\Pr[g_i(a) = g_i(b)] < (p_{low})^k$
  - $\mathbb{E}[|\{a \in P, i \leq L: dist(a,q) > cr, g_i(a) = g_i(q)\}|] \leq L \cdot n \cdot (p_{low})^k \leq L \cdot n \cdot (p_{low})^{\log_{1/p_{low}} n} \leq L$
  - By Markov: $\Pr[\#\ outliers > 3L] < 1/3$

➢ With constant probability $(1 - \frac{1}{e} - \frac{1}{3})$, all is good! (captured at least one close point, and not too many outliers)

➢ Again with logarithmic repetition, the probability of success can be boosted to high probability, i.e., $1 - 1/n$

Retrieval Algorithm

- $k = \lceil \log_{1/p_{low}} n \rceil$
- $L = n^{\rho}/p_{high}$
- Let $\rho = \frac{\log 1/p_{high}}{\log 1/p_{low}}$

- Given a query q,
- For $i = 1$ to L
  - Inspect the points in $g_i(q)$ one by one
  - If one of them has distance closer than cr to the query, report it
  - If more than $3L$ points are inspected, abort (try with the next data structure).

➢ Query time: $O(kL) \approx n^{\rho}$

➢ Space Usage: $O(nd + nL) \approx n^{1+\rho}$

How to get a $(r, cr, p_{high}, p_{low})$ –sensitive family of Hash Functions

Hamming Metric

- $P \subseteq \{0,1\}^d$, $dist$ is the hamming distance between the points
- $\mathcal{H}$ is the family of projections onto a single coordinate, i.e., $h_i(p) = p_i$
- Claim: $\mathcal{H}$ is $\left(r, cr, 1 - \frac{r}{d}, 1 - \frac{cr}{d}\right)$ −sensitive
  - If $dist(p_1, p_2) \leq r$, then the probability of sampling from those different bits is at most $\frac{r}{d}$
  - If $dist(p_1, p_2) \geq cr$, then the probability of sampling from those different bits is at least cr/d
- Need to compute $\rho = \frac{\log 1/p_{high}}{\log 1/p_{low}}$

Hamming Metric

- $k = \lceil \log_{1/p_{low}} n \rceil$
- $L = n^{\rho}/p_{high}$
- Let $\rho = \frac{\log 1/p_{high}}{\log 1/p_{low}}$

- $P \subseteq \{0,1\}^d$, $dist$ is the hamming distance between the points
- $\mathcal{H}$ is the family of projections onto a single coordinate, i.e., $h_i(p) = p_i$
- Claim: $\mathcal{H}$ is $\left(r, cr, 1 - \frac{r}{d}, 1 - \frac{cr}{d}\right)$ –sensitive, $p_{high} = 1 - \frac{r}{d}$, $p_{low} = 1 - \frac{cr}{d}$
- Need to compute $\rho = \frac{\log 1/p_{high}}{\log 1/p_{low}} = \frac{\log p_{high}}{\log p_{low}} = \frac{\log(1-x)}{\log(1-tx)} \leq \frac{1}{t} = \frac{1-p_{high}}{1-p_{low}} = \frac{1}{c}$
  - $t = \frac{1-p_{low}}{1-p_{high}}$, $x = 1 - p_{high}$
  - For $x \in [0,1)$ and $t \geq 1$ such that $1 - tx > 0$, we have $\frac{\log(1-x)}{\log(1-tx)} \leq \frac{1}{t}$
    - Since $\log(1 - tx) \leq 0$, we need to show $t \log(1 - x) \geq \log(1 - tx)$
    - Equivalent to show that $f(x) = (1 - tx) - (1 - x)^t \leq 0$ which is true for $x = 0$
    - Take derivative: $f'(x) = -t + t(1 - x)^{t-1}$ which is non-positive for $x \in [0,1)$ and $t \geq 1$
      - Equivalent to show $-1 + (1 - x)^{t-1} \leq 0$ or $(1 - x)^{t-1} > 1$

Other distances

- $k = \lceil \log_{1/p_{low}} n \rceil$
- $L = n^{\rho}/p_{high}$
- Let $\rho = \frac{\log 1/p_{high}}{\log 1/p_{low}}$

❑ L_1-distance:

- Impose a randomly shifted grid of side length proportional to r
- Discretize the grid -> the points will be in $\{0, \dots, M\}^d$
- Map them using a unary mapping to $\{0,1\}^{dM}$
  - $unary((x_1, \dots, x_d)) = unary(x_1) \dots unary(x_d)$
  - $unary(x) = 111100000$ (there are x 1's followed by $M - x$ 0's)
- The distance is now changed to Hamming distance
- Get $\rho \approx 1/c$

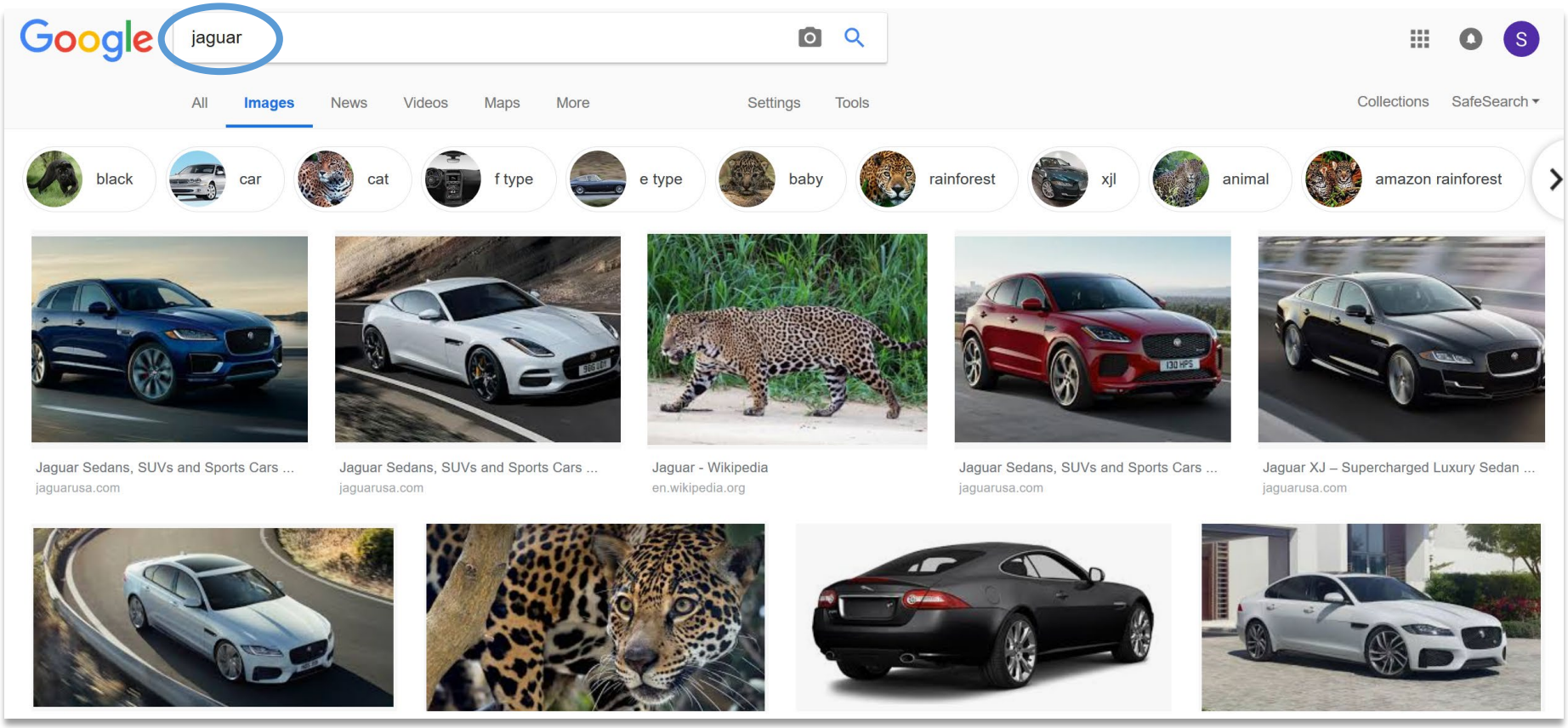
❑ L_2- (Euclidean) distance

- Project points on a random line, divide the line randomly randomly shifted into segments of size w, the segments will be the buckets
- Get $\rho \approx 1/c^2$

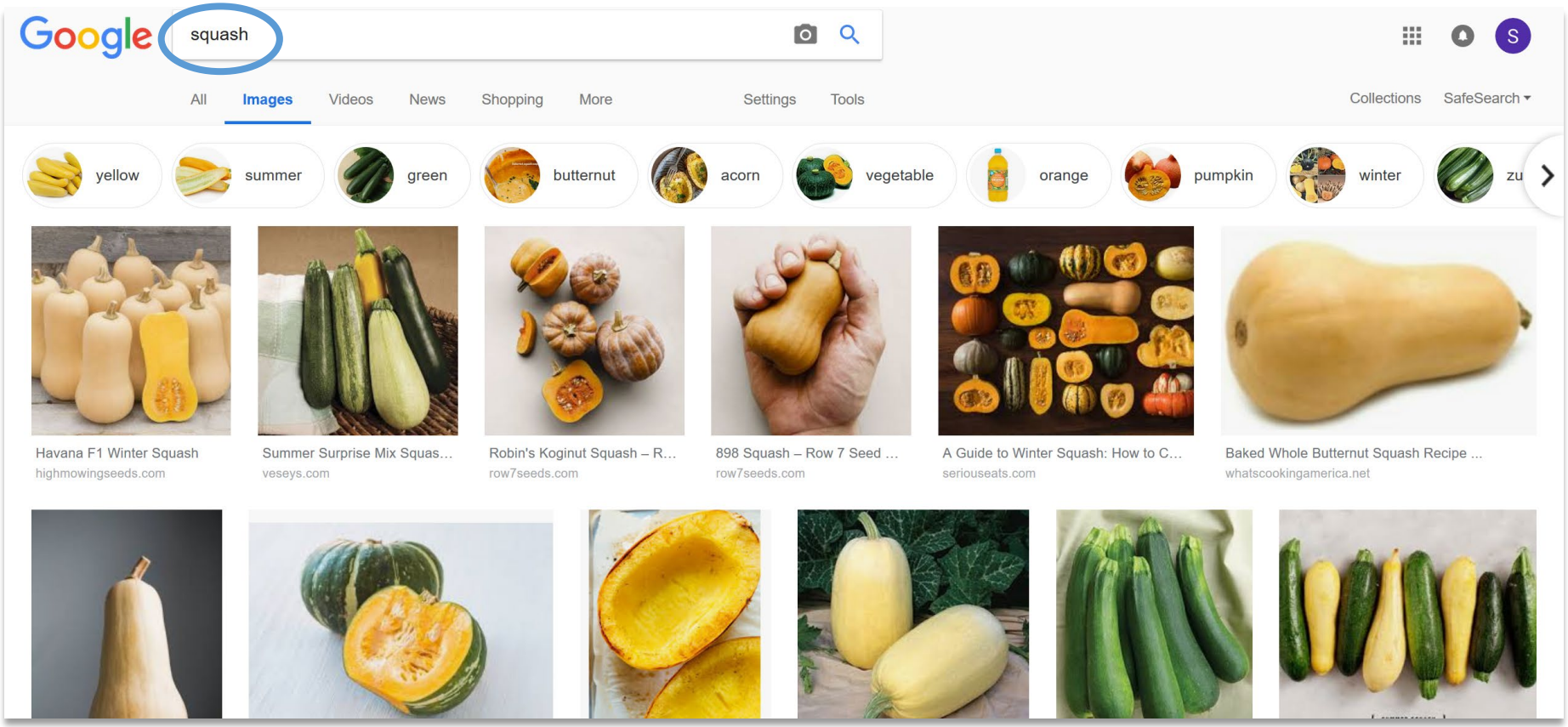
❑ L_p distances, Jaccard Similarity: $\frac{|A \cap B|}{|A \cup B|}$

Variants of NN

Report diverse results for a search query



Report diverse results for a search query



Be fair among all qualified candidates

Applications:

- ❑ Removing noise, k-NN classification
- ❑ Anonymizing the data
- ❑ Counting the neighborhood size

Laughing

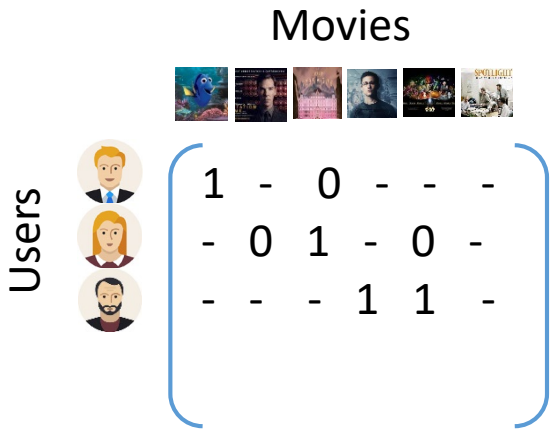
$\frac{1}{2}$ $\frac{1}{2}$

Some part of the data is noisy, incomplete or irrelevant

The data points are:

- **corrupted, noisy**
  - Image denoising

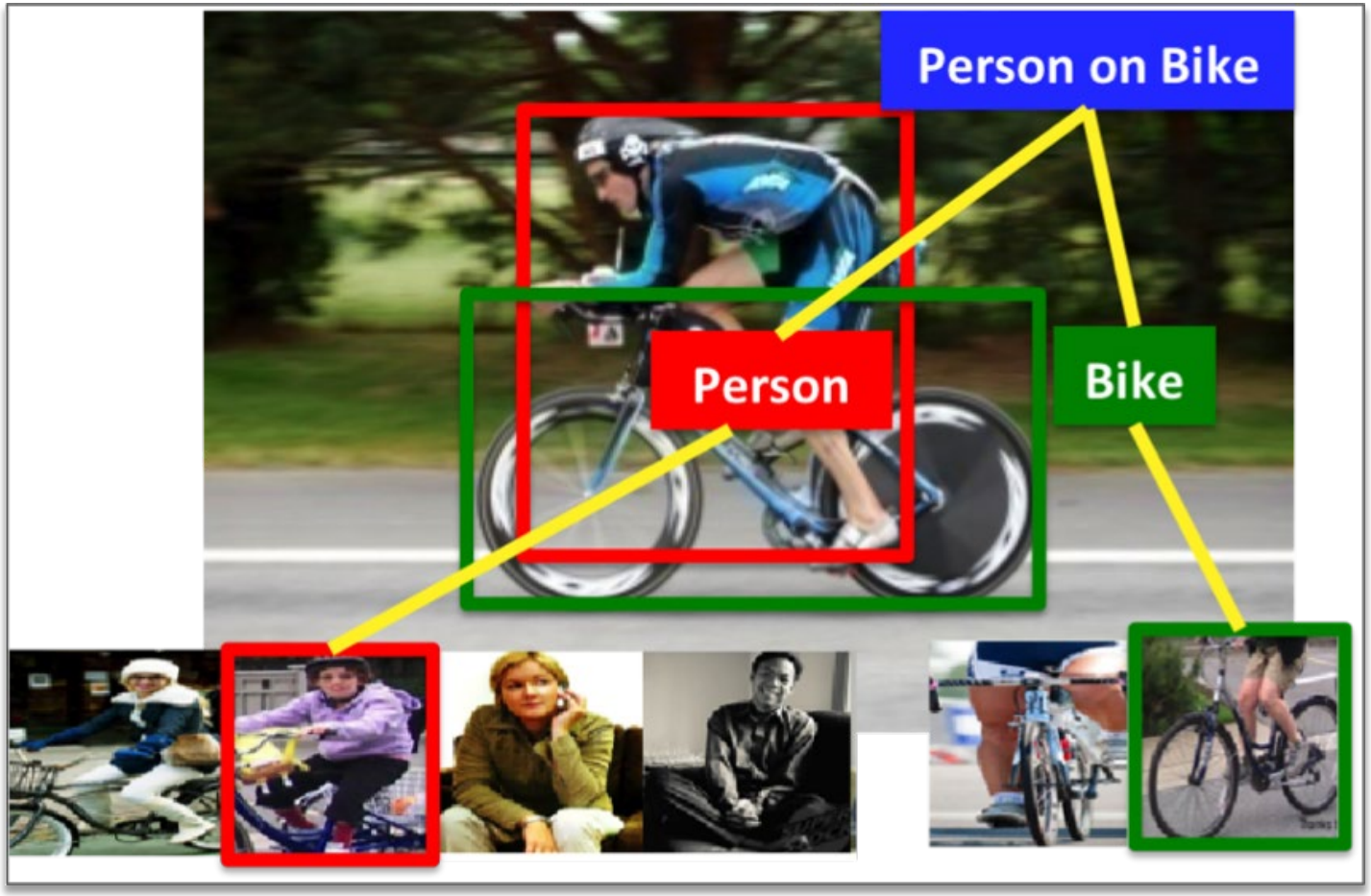
- **Incomplete**
  - Recommendation: Sparse matrix



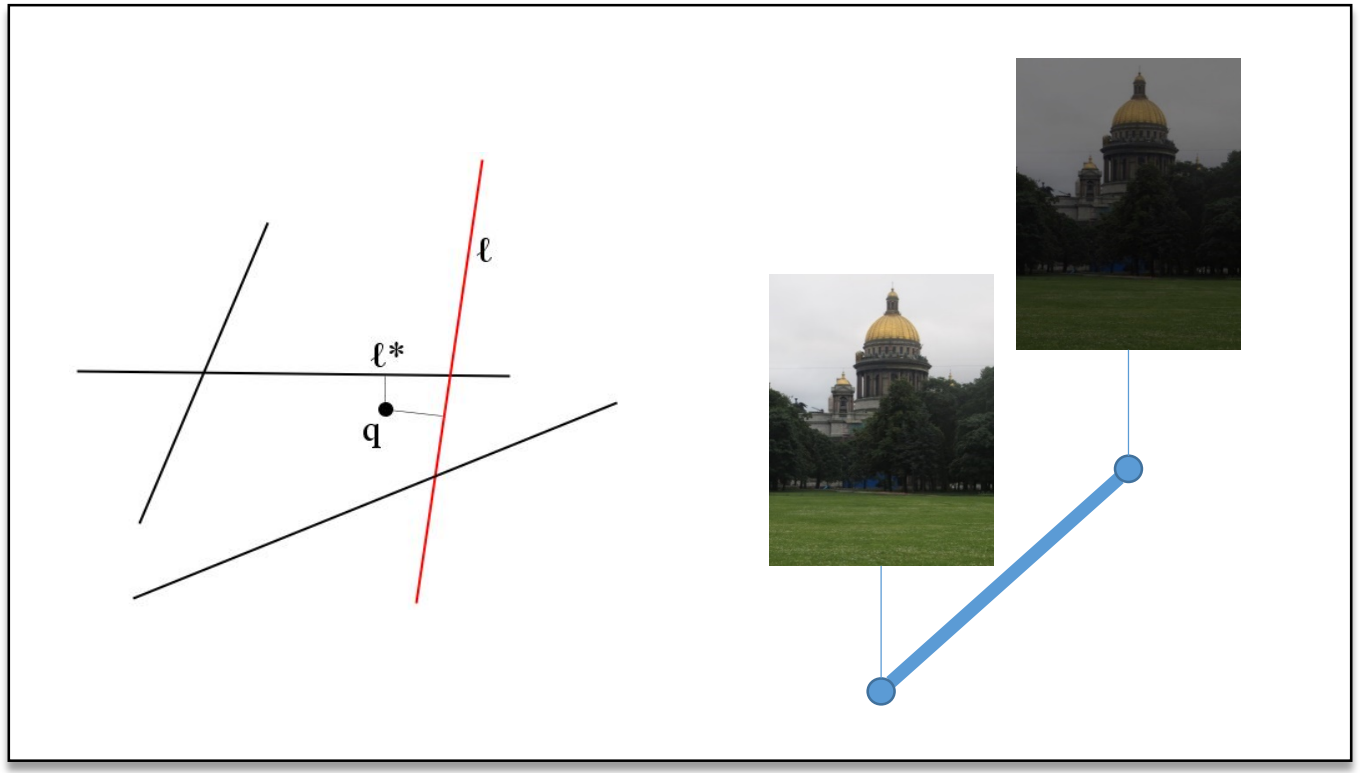
- **Irrelevant**
  - Occluded image

➢ **Need Robustness**

Multiple Related Queries



Dataset contains more complex objects



E.g. Affine subspaces are used to model data under linear variations.

Other directions

Data Dependent LSH

- Standard LSH is oblivious to the data (many advantages)
- Constants in the exponents can be improved using existing structures in the data
  - If some parts of the data is more dense, a ball can be carved
  - Otherwise the data looks random -> improved LSH bounds

Lower Dimensional Case

- Many work on the lower dimensional case, i.e., using kd-trees