Flexible Aggregate Similarity Search

Yang Li, Feifei Li, Ke Yi, Bin Yao, Min Wang

Aggregation Nearest Neighbor (ANN)

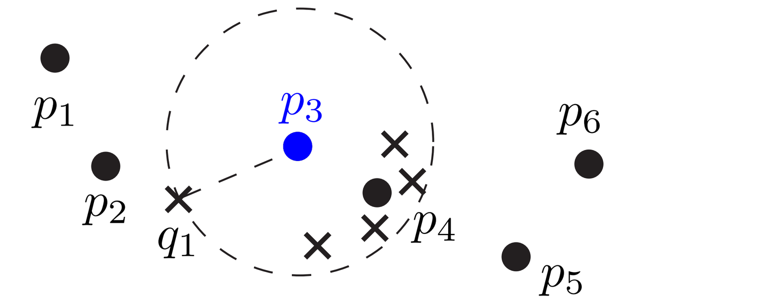
Given an aggregation σ, a similarity/distance function d, a dataset P, and any query group Q:

aggregate similarity distance of p: $r_p = \sigma\{d(p,Q)\} = \sigma\{d(p,q_1),\ldots,d(p,q_{|Q|})\}$, for any p

Find $p^* \in P$ having the smallest r_p value ($r_{p^*} = r^*$).

ANN: $\sigma = \max$

agg= max, $p^* = p_3$, $r^* = d(p_3, q_1)$

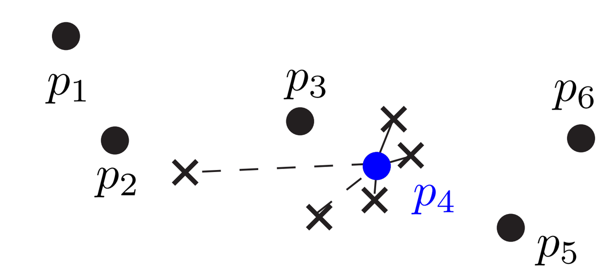


× : group Q of query points

● : dataset P

ANN: $\sigma = \mathrm{sum}$

agg=sum, $p^* = p_4$, $r^* = \sum_{q\in Q} d(p_4, q)$

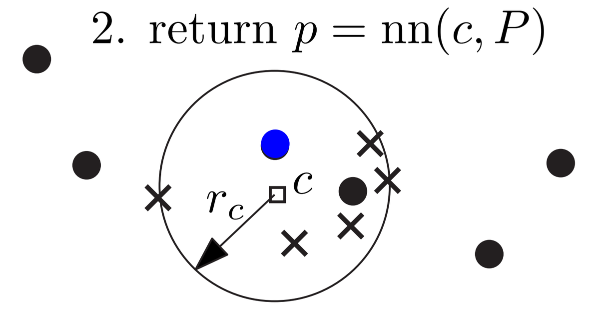


× : group Q of query points

● : dataset P

Our approach for $\sigma = \max$: AMAX1

1. $\mathcal{B}(c, r_c) = \mathrm{MEB}(Q)$
2. return $p = \mathrm{nn}(c, P)$

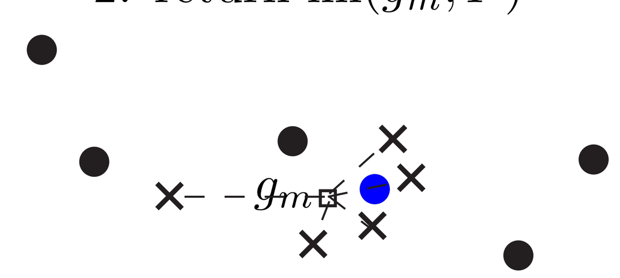


× : group Q of query points

● : dataset P

Our approach for $\sigma = \mathrm{sum}$: ASUM1

1. g_m is the geometric median of Q
2. return $\mathrm{nn}(g_m, P)$

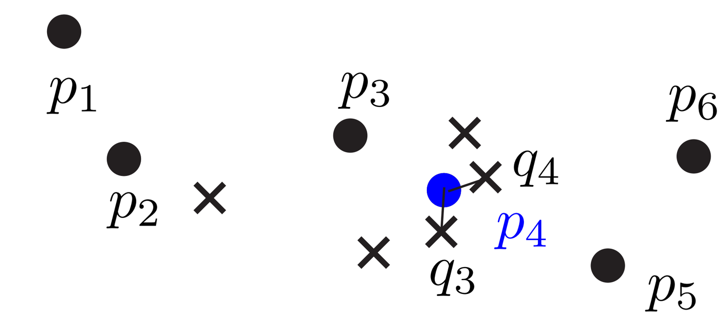


× : group Q of query points

● : dataset P

Flexible aggregate similarity search (FANN)

$\sigma = \mathrm{sum}$, $\phi = 40\%$, $p^* = p_4$,
$r^* = d(p_4, q_3) + d(p_4, q_4)$



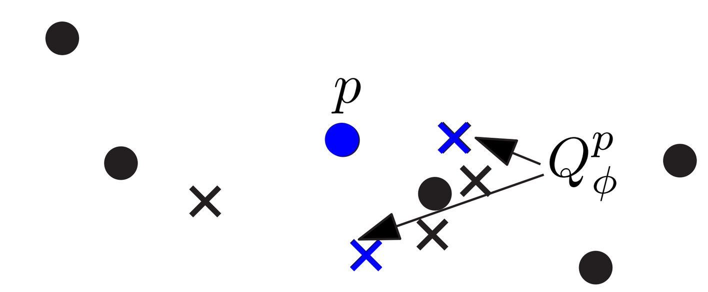
× : group Q of query points

● : dataset P

Given support $\phi \in (0,1]$, find an object in P that has the best aggregate similarity to (*any*) $\phi|Q|$ query objects.

Exact method for FANN

Q^p_ϕ : top $\phi|Q|$ NNs of p in Q



× : group Q of query points

● : dataset P

$\phi = 0.4$, $|Q| = 5$, $\phi|Q| = 2$

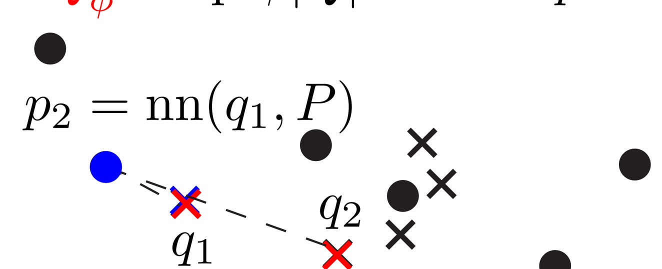
For $\forall p \in P$, $r_p = \sigma(p, Q^p_\phi)$, where Q^p_ϕ is p's $\phi|Q|$ NNs in Q.

The brute-force-search (BFS) approach in any dimension: for each $p \in P$, find out Q^p_ϕ and calculate r_p.

Approximate method for $\sigma = \mathrm{sum}$: ASUM

Q^p_ϕ : top $\phi|Q|$ NNs of p in Q

$p_2 = \mathrm{nn}(q_1, P)$



$r_{p_2} = d(p_2, q_1) + d(p_2, q_2)$

× : group Q of query points

● : dataset P

$\phi = 0.4$, $|Q| = 5$, $\phi|Q| = 2$, $\sigma = \mathrm{sum}$

Repeat this for every $q_i \in Q$, return the p with the smallest r_p.

Theorem 3: In any dimension d, given an exact NN algorithm, ASUM is an 3-approximation. Given an β-approximate NN algorithm, ASUM is an $(\beta+2)$-approximation.

Improvement for ASUM

Theorem 4: For any $0 < \epsilon, \lambda < 1$, executing ASUM algorithm only on a random subset of $f(\phi,\epsilon,\lambda)$ points of Q returns a $(3+\epsilon)$-approximate answer to FANN search in any dimensions with probability at least $1-\lambda$, where

$$f(\phi,\epsilon,\lambda) = \frac{\log\lambda}{\log(1-\phi\epsilon/3)} = O(\log(1/\lambda)/\phi\epsilon).$$

For $|Q| = 1000$, $\phi = 0.4$, $\lambda = 10\%$, $\epsilon = 0.5$, only needs 33 NNs search in any dimension. (much less in practice, $\frac{1}{\phi}$ is enough!)

Independent of dimensionality, $|P|$, and $|Q|$!

Theoretical bounds

Theorem 1: AMAX1 is a $\sqrt{2}$-approximation in any dimension d given (exact) $\mathrm{nn}(c,P)$ and $\mathrm{MEB}(Q)$. Given an α-approximate MEB algorithm and an β-approximate NN algorithm, AMAX1 is an $\sqrt{\alpha^2+\beta^2}$-approximation.

Theorem 2: ASUM1 is a 3-approximation in any dimension d given (exact) geometric median and $\mathrm{nn}(c,P)$. Given an β-approximate NN algorithm, ASUM1 is an 3β-approximation.

Approximate method for $\sigma = \max$: AMAX

Similar to ASUM, but instead, find $c_i = \mathrm{MEB}(Q^{q_i}_\phi)$, then replace $\mathrm{nn}(q_i, P)$ with $\mathrm{nn}(c_i, P)$ where the rest stays the same. But the analysis is much more involved, and we can show:

Theorem 5: In any dimension d, given an exact NN algorithm and an MEB, AMAX is an $1+2\sqrt{2}$-approximation. Given an β-approximate NN algorithm, AMAX is an $(1+2\sqrt{2})\beta$-approximation.

Improvement:

Theorem 6: A random sample from Q of size $O(\log(1/\lambda)/\phi)$ is sufficient to give the same approximation with at least $1-\lambda$ probability. In practice, a sample size of $\frac{1}{\phi}$ is enough (i.e., only needs $\frac{1}{\phi}$ NNs).

Experiments

dataset	number of points	dimensionality
Color	$68,040$	32
MNIST	$60,000$	50
Cortina	$1,088,864$	74

For more results in low dimensions (up to tens of millions of points using OpenStreet Map data), please refer to our paper.