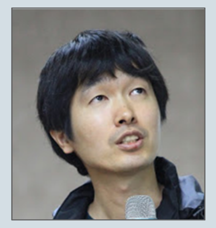


Wanbin Son
Aarhus University

MADALGO
CENTER FOR MASSIVE DATA ALGORITHMICS

Danmarks Grundforskningsfond
Danish National Research Foundation

Group Nearest Neighbor Queries in the L_1 plane

Introduction

We propose an efficient algorithm to compute the group k nearest neighbor (NN) points.

- Motivation
  - The nearest neighbor query problem is one of the fundamental problems in computer science

  - For a set of query points, how can we find k-NN points from them?

- Problem
  - Input (in the L_1 plane)
    - a set P of n data points
    - a set Q of m query points ($m \le n$)
    - a positive integer k
  - $dist_Q(p) = \sum_{q \in Q} dist(p, q)$
  - Output : the k closest points in P with respect to Q

We want to construct a query structure of P to support GNN queries.

- We assume that all points are on the L_1 Plane
  - $dist(p, q) = |p.x - q.x| + |p.y - q.y|$

Straightforward Algorithm

Straightforward Algorithm to compute group k-NN points:

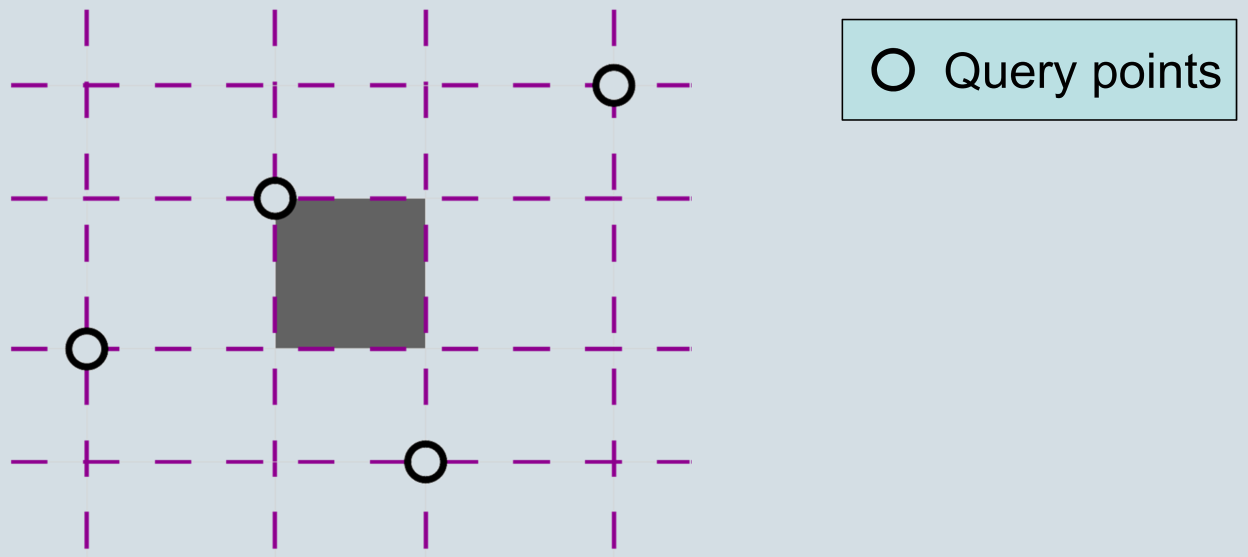
1. Compute sum-of-distances for all data points from Q
2. Report the k closest points by using selection algorithm

➔ $O(mn)$ time

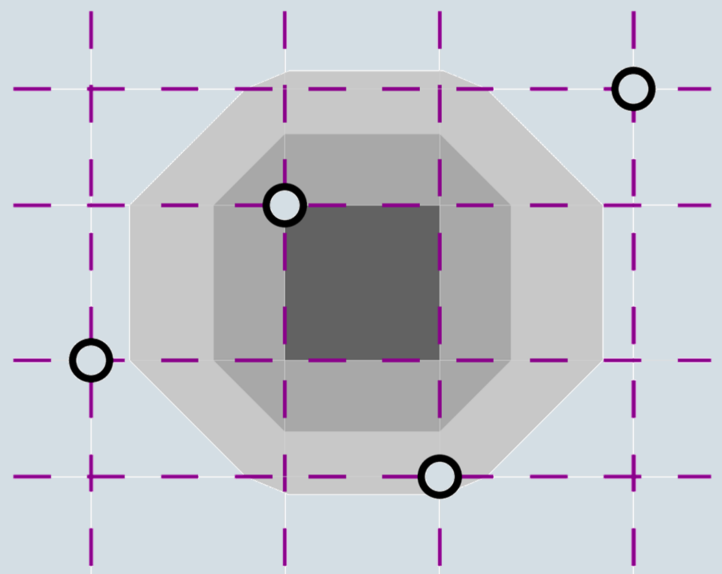
for small m and k, inefficient

Main Idea

1. Compute the region that minimizes the distance from Q
   - The median cell of the grid of Q has the minimum distance



2. Expand this region by increasing the sum-of-distance value



   - The region is a convex polygon
     $A_Q(c) = \{x \mid dist(x, Q) \le c\}$ is a sublevel set of $dist(x, Q)$, and $dist(x, Q)$ is a convex function, so A_Q is convex
   - Function $dist_Q$ is linear in each cell g, and a slope of the boundary of $A_Q(c) \cap g$ is as follows.

$$\text{slope(g)} = \frac{m_r(g) - m_l(g)}{m_b(g) - m_t(g)}$$

($m_l(g)$, $m_r(g)$, $m_t(g)$ and $m_b(g)$ are # of query points that are to the left, right, above, and below of any point in g, respectively)

3. Report the points in P hit by the region until the k-th point
   - *Top-k orthogonal range query* or *segment dragging query* is used to find points hit by the region

Results

RNGALGO (algorithm based on top-k orthogonal range query)

- Query time : $O(T_{min} \log n + T_{max}(\log\log n + \log m))$
  - $T_{min} = \min\{k + m, m^2\}$, $T_{max} = \max\{k + m, m^2\}$
- Preprocessing time : $O(m^2 n \log^2 n)$
- Space : $O(m^2 n \log^2 n)$

SGMTALGO (algorithm based on segment dragging query)

- Query time : $O\big((k + m)\log^2 n + m^2(\log^\varepsilon n + \log m)\big)$
- Preprocessing time : $O(m^2 n \log n)$
- Space : $O(m^2 n)$

➔ for small m and k, efficient

It is unlikely that we achieve an $o(n)$ time algorithm without any preprocessing because of the lower bound for the selection problem

Our approach can be extended for group k-farthest neighbor query and the weighted group k-NN query

References

[1] S.W. Bae, M. Korman, and T. Tokuyama. *All farthest neighbors in the presence of highways and obstacles*. In Proc. the 3rd International Workshop on Algorithms and Computation, 2009.

[2] S. W. Bent and J. W. John. *Finding the median requires 2n comparisons*. In Proc. the 17th annual ACM symposium on Theory of computing, 1985.

[3] S. Rahul, P. Gupta, R. Janardan, and K. S. Rajan. *Efficient top-k queries for orthogonal ranges*. In Proc. the 5th International Conference on WALCOM: Algorithms and Computation, 2011.

[4] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1996.

MADALGO – Center for Massive Data Algorithmics, a Center of the Danish National Research Foundation