Finding HSP Neighbors via an Exact, Hierarchical Approach

Brown University
School of Engineering

Cole Foster, Edgar Chávez, Benjamin Kimia
{Cole_Foster, Benjamin_Kimia}@brown.edu, elchavez@cicese.edu.mx

CICESE

sisap

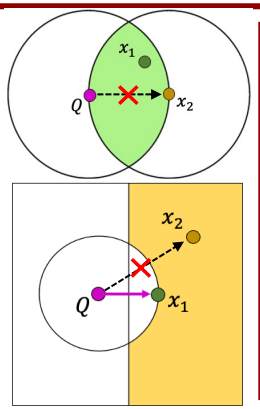
The Half-Space Proximal (HSP) Graph is a sparse proximity graph with a wide array of **applications**:

- Routing and edge pruning
- Instance-based classification
- Intrinsic dimensionality estimation
- Representing chemical networks
- Defining a notion of hubness/centrality

The HSP Graph is defined geometrically by the same inequalities as the Relative Neighborhood Graph (RNG), making it a super-set of linear complexity.

The HSP Algorithm on a query Q begins with a list L initialized with all points in the dataset and iteratively:

1. Finds the next HSP neighbor x_1 as the closest point to Q in L.
2. Removes point $x_2 \in L$ if both:
   $d(Q, x_1) < d(Q, x_2)$
   $d(x_1, x_2) < d(Q, x_2)$

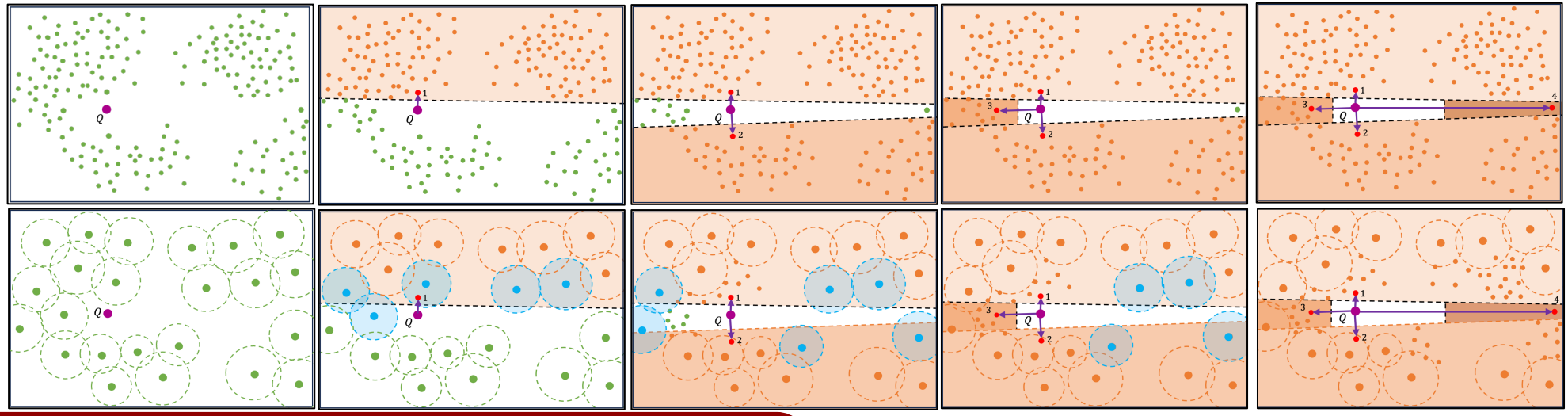


Importance of Exactness:

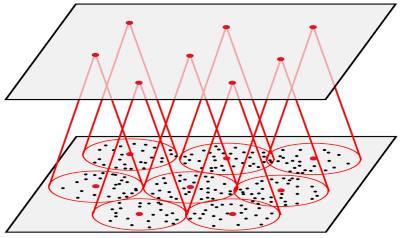
- Existing approaches are **approximate**, missing long range links by constraining the HSP algorithm to a local neighborhood.
- The exact HSP Graph is **monotonic**, a highly desired property in graph-based similarity search.
- The **t-spanner** conjecture of the HSP Graph requires exactness and leads to its application as a distance oracle.

Original HSP

Hierarchical HSP

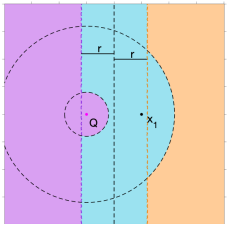


The Hierarchical HSP (HHSP) algorithm leverages a hierarchical partitioning of the dataset to speedup the two core processes of the HSP Algorithm.



Nearest Neighbor Search by pivots and the triangle inequality is used to efficiently find HSP neighbors.

Validating Entire Groups of Points at a time by metric-space inequalities establishes three cases for any pivot and its partition: **(1) invalidated**, **(2) safe**, or **(3) indeterminant.**



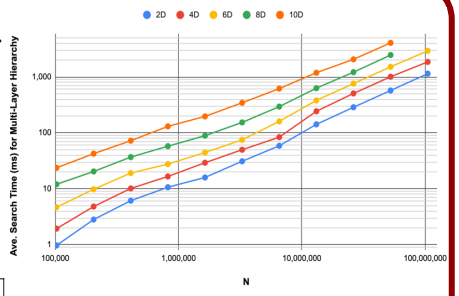
All points within a radius r of pivot p_2 cannot be an HSP neighbor of Q if both:

$$\begin{cases} d^2(Q, p_2) - d^2(x_1, p_2) > 2r\ d(Q, x_1) \\ d(Q, x_1) < d(Q, p_2) - r, \end{cases}$$

Results on synthetic data shows the HHSP provides orders of magnitude savings over brute force HSP, enabling *exact* HSP search on datasets with hundreds of millions of points.

Search time (ms) on 1.6M uniformly distributed points.

Dimension	Brute Force HSP	Hierarchical HSP	Ratio
2	187.516	0.016	11,749.51
4	181.696	0.029	6,209.85
6	132.222	0.045	2,960.31
8	135.741	0.089	1,522.53
10	135.353	0.197	686.70



Search time (ms) on uniformly distributed data, scaling to hundreds of millions of points and suggesting experimentally sublinear complexity.