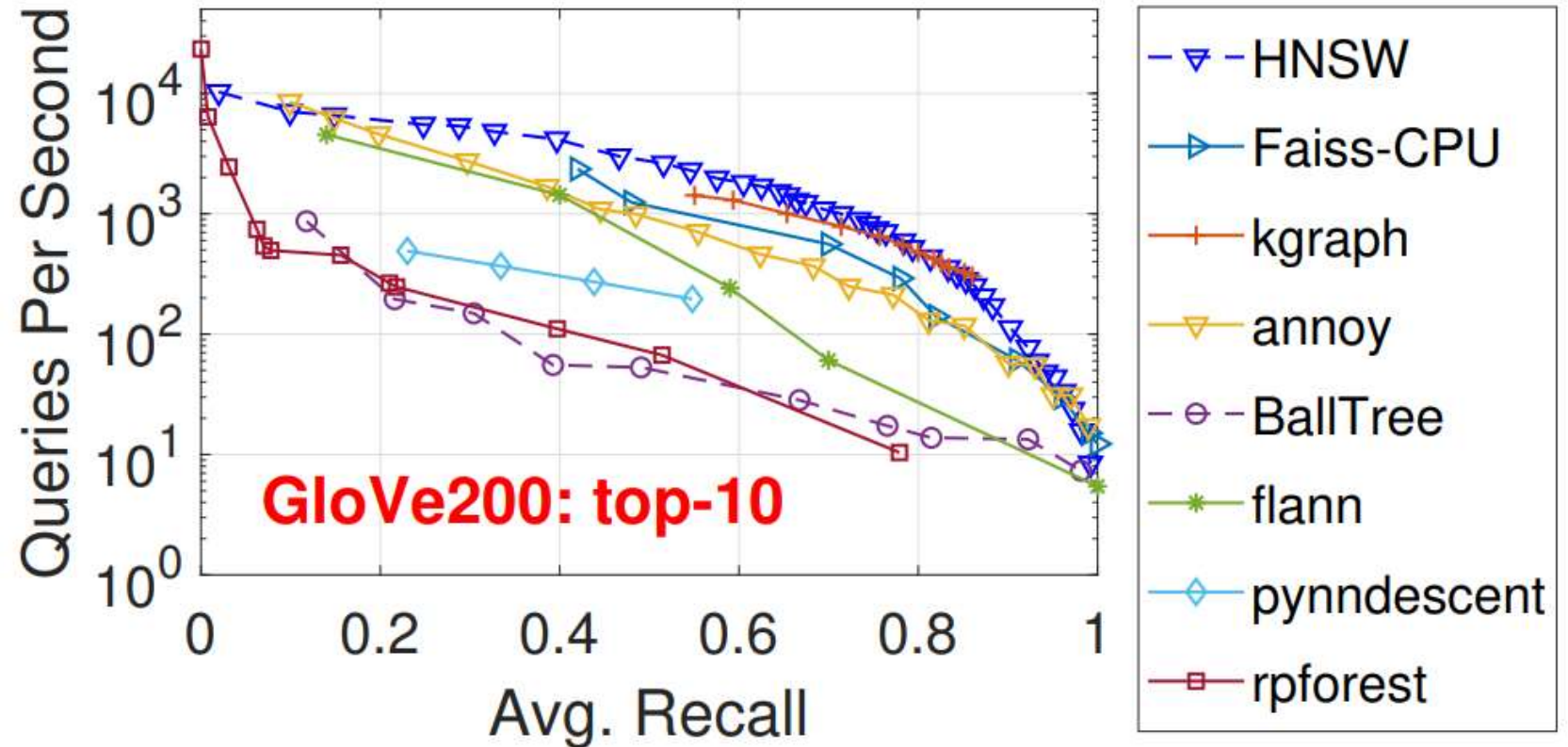# Approximate Nearest Neighbor Search

Weijie Zhao

Approximate Nearest Neighbor Search

- Tree
- Hashing
- Quantization
- Graph

K-D Tree

```
function kdtree (list of points pointList, int depth)
{
    // Select axis based on depth so that axis cycles through all valid values
    var int axis := depth mod k;

    // Sort point list and choose median as pivot element
    select median by axis from pointList;

    // Create node and construct subtree
    node.location := median;
    node.leftChild := kdtree(points in pointList before median, depth+1);
    node.rightChild := kdtree(points in pointList after median, depth+1);
    return node;
}
```
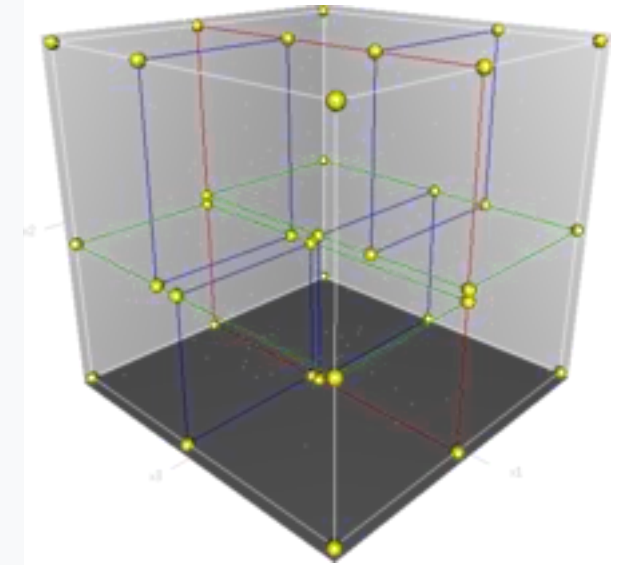
Locality-Sensitive Hashing (LSH)

- Random Projection

- MinHash

- Consistent Weighted Sampling

Algorithm 1 Generalized consistent weighted sampling (GCWS) for hashing the pGMM kernel.

Input: Data vector u_i $(i = 1$ to $D)$

Generate vector $\tilde{u}$ in $2D$-dim by (1).

For i from 1 to $2D$
$$r_i \sim Gamma(2,1), \ c_i \sim Gamma(2,1), \ \beta_i \sim Uniform(0,1)$$
$$t_i \leftarrow \lfloor p\frac{\log \tilde{u}_i}{r_i} + \beta_i \rfloor, \ a_i \leftarrow \log(c_i) - r_i(t_i + 1 - \beta_i)$$
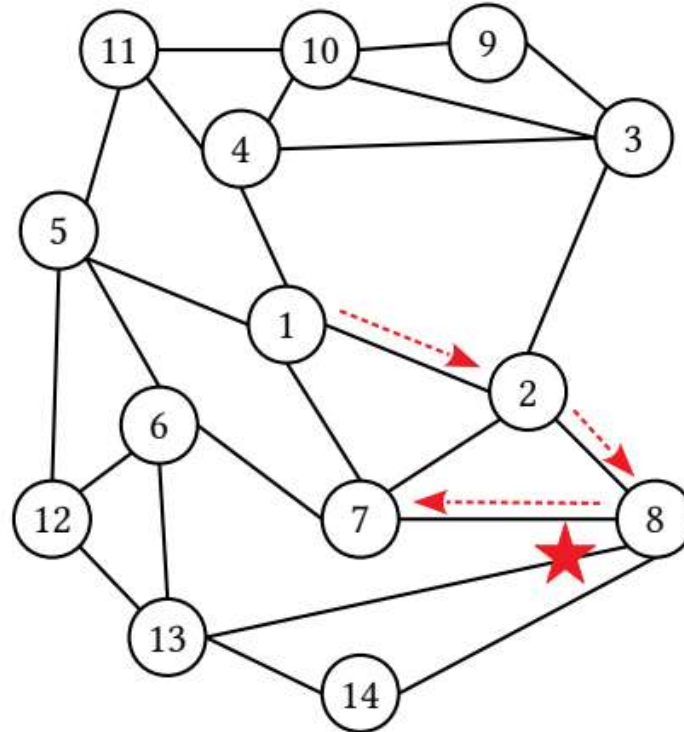End For

Output: $i^* \leftarrow arg\min_i \ a_i, \qquad t^* \leftarrow t_{i^*}$

Quantization

- Map high dimensional vector to low dimensional integer
- K means
- Product Quantization (PQ)

Graph-Based ANN

- Hierarchical Small World Graph (HNSW)

- Search ON Graph (SONG)

- BipartitE Graph Indices (BEGIN)