Graph-based Nearest Neighbor Search: From Practice to Theory

Liudmila Prokhorenkova, Aleksandr Shekhovtsov

Yandex Research

ICML 2020

Nearest neighbor search

- Dataset $\mathcal{D} = \{\mathsf{x}_1, \ldots \mathsf{x}_n\}$, $\mathsf{x}_i \in \mathbb{R}^d$
- For a given query q let $\mathsf{x} \in \mathcal{D}$ be its nearest neighbor
- Exact NNS: find x
- c-ANN: find such x' that $\rho(q, \mathsf{x}') \leq c\rho(q, \mathsf{x})$

Graph-based algorithms

Main idea:

- Construct a proximity graph, where each element of $\mathcal{D}$ is connected to its nearest neighbors
- For a given query q, take an element in $\mathcal{D}$ and make greedy steps towards q on the graph
- At each step, check the neighbors of the current node

Malkov Y., Yashunin D. "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs". IEEE transactions on pattern analysis and machine intelligence, 2018.

Graph-based algorithms

Main idea:

- Construct a proximity graph, where each element of $\mathcal{D}$ is connected to its nearest neighbors
- For a given query q, take an element in $\mathcal{D}$ and make greedy steps towards q on the graph
- At each step, check the neighbors of the current node

Additional heuristics:

- Adding shortcut edges
- Beam search: maintaining a dynamic list of several candidates instead of just one optimal point
- Diversification of neighbors

Malkov Y., Yashunin D. "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs". IEEE transactions on pattern analysis and machine intelligence, 2018.

Overview of our work

- Graph-based methods are known to outperform other approaches in many large-scale applications, but they do not have much theoretical support[1]
- We fill this gap assuming the uniform distribution of data
- We mostly focus on the dense regime ($d \ll \log n$)
- We show the effect of:
  - Local kNN edges
  - Properly distributed long edges
  - Beam search
- We empirically motivate our assumptions about dense regime and uniform distribution

[1] We are aware of one related study: Laarhoven, T. "Graph-based time-space trade-offs for approximate near neighbors". SoCG 2018.

Dense and sparse regimes

- **Dense regime**: $d \ll \log n$
- **Sparse regime**: $d \gg \log n$

Assuming the uniform distribution over a d-dimensional sphere:

- **Dense regime**: the nearest neighbor is at distance $n^{-1/d} \to 0$
- **Sparse regime**: the nearest neighbor is at distance $\approx \sqrt{2}$, as other elements

Dense and sparse regimes

Complexity of known exact algorithms scales exponentially in d, which is a problem in sparse regime $d \gg \log n$.

While real-world datasets may have large d, they usually have lower *intrinsic dimension*.

Fortunately, most graph-based algorithms do not care about the original dimension.

Plan:

- NN graphs in dense regime
- Shortcut edges
- Beam search
- Empirical illustrations

Plain NN graphs in dense regime

For any constant $M > 1$, let $G(M)$ be a graph obtained by connecting $\mathbf{x}_i$ and $\mathbf{x}_j$ iff $\rho(\mathbf{x}_i, \mathbf{x}_j) \le \arcsin\left(M\, n^{-1/d}\right)$.

Theorem (simplified)

Let $d \gg \log\log n$ and $M > \sqrt{2}$.
Then, with probability $1 - o(1)$, $G(M)$-based NNS solves the NN problem.
Time complexity is $\Theta\left(d^{1/2} \cdot n^{1/d} \cdot M^d\right)$.
Space complexity is $\Theta\left(n \cdot d^{-1/2} \cdot M^d \cdot \log n\right)$.

- The expected number of neighbors is $\Theta(d^{-1/2} \cdot M^d)$
- So, the complexity of one step is $\Theta(d^{1/2} \cdot M^d)$
- The number of steps is $\Theta\left(n^{1/d}\right)$

Long edges on a lattice

Kleinberg's result:

- Consider a 2-dimensional grid
- Each node has local edges + one random long link
- The probability of a link from u to v is proportional to $\rho(u, v)^{-r}$
- If $r = 2$, the greedy graph-based search finds the target element in $O\left(\log^2 n\right)$ steps
- Any other r gives at least n^φ with $\varphi > 0$

Kleinberg J. "The small-world phenomenon: An algorithmic perspective". ACM symposium on Theory of computing, 2000.

Long edges in our setting

$$P(\text{edge from } u \text{ to } v) = \frac{\rho(u,v)^{-d}}{\sum_{w \neq u} \rho(u,w)^{-d}} \, .$$

Theorem

Sampling one long edge for each node reduces the number of steps to $O(\log^2 n)$ (with high probability).

- Importantly, we allow $d \to \infty$

Long edges in our setting

$$P(\text{edge from } u \text{ to } v) = \frac{\rho(u,v)^{-d}}{\sum_{w \neq u} \rho(u,w)^{-d}}\,.$$

Theorem

Sampling one long edge for each node reduces the number of steps to $O(\log^2 n)$ (with high probability).

- Importantly, we allow $d \to \infty$

- Long edges can guarantee $O\left(\log^2 n\right)$ steps
- Plain NN graphs give $\Theta\left(n^{1/d}\right)$ steps
- So, reducing the number of steps is reasonable if $d < \frac{\log n}{2 \log \log n}$

Dimension-independent probabilities

Dimension-independent probabilities

Let $\rho_{rank}(u, v) = (k/n)^{1/d}$ if v is the k-th neighbor of u

$\rho(u, v) \propto \rho_{rank}(u, v)$ for uniform datasets (the number of nodes at distance ρ grows as ρ^d)

$$P(\text{edge to } k\text{-th neighbor}) \propto \frac{1}{k}$$

This distribution is *dimension-independent*

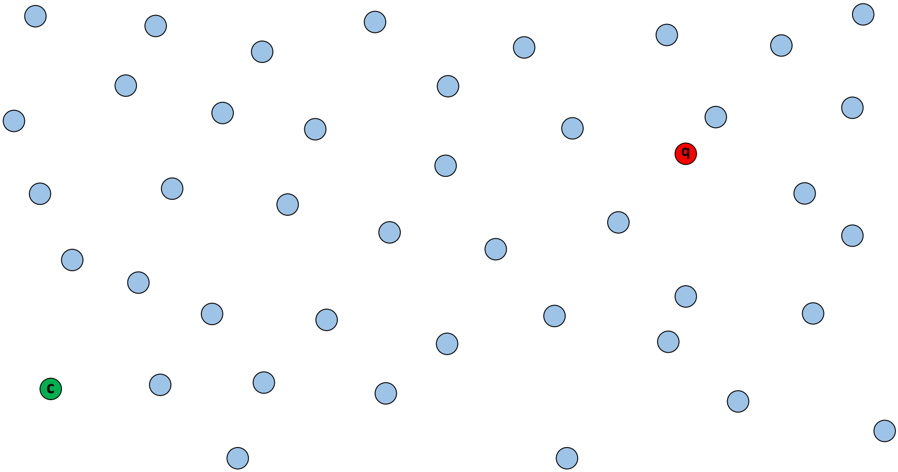
Beam search

Theorem (informal)

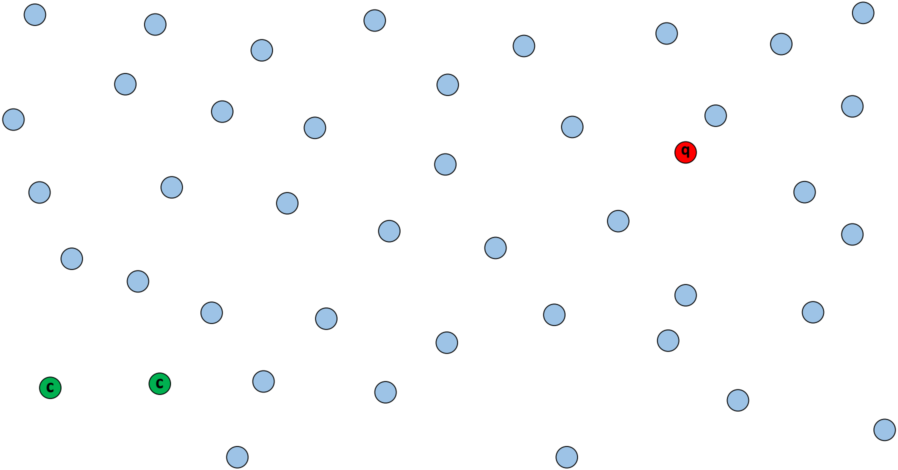
Using beam search allows to get space complexity L^d and time complexity $R^d L^d$, where $L, R > 1$ and $L^2 \left(1 - \frac{L^2}{4R^2}\right) > 1$.

- In particular, time complexity can be reduced to $\left(\frac{27}{16}\right)^{d/2}$
- Without beam search we can get only $2^{d/2}$

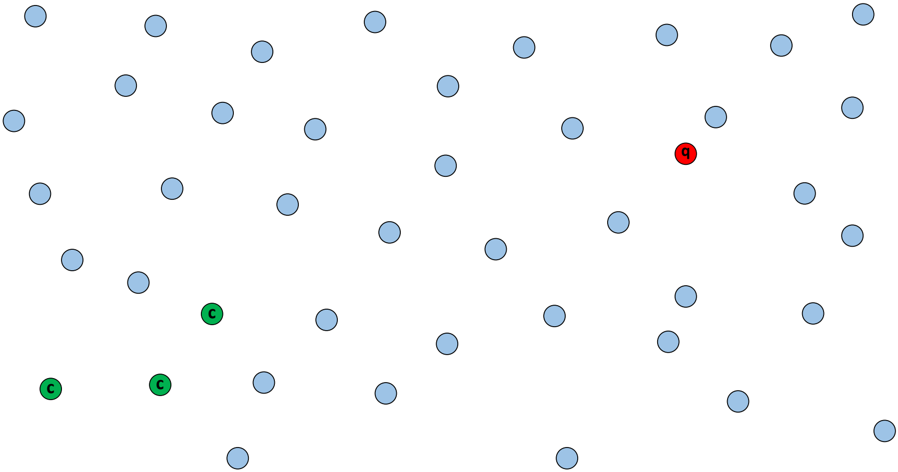
Beam search



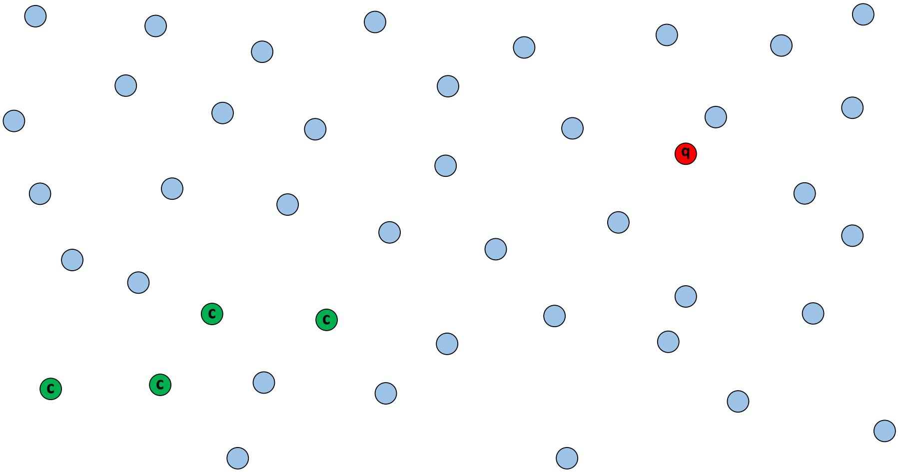
Beam search



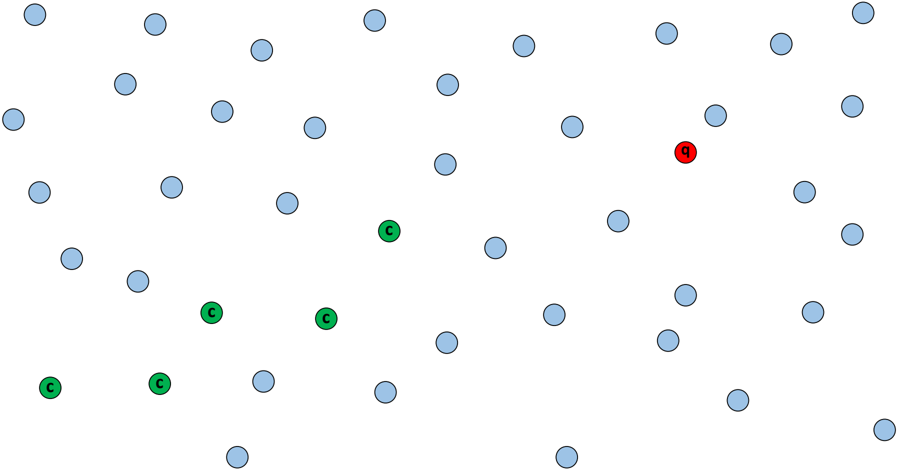
Beam search



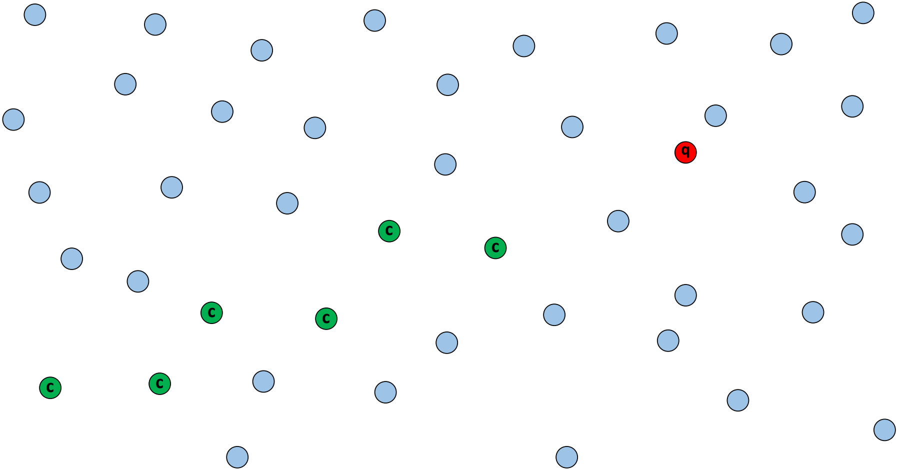
Beam search



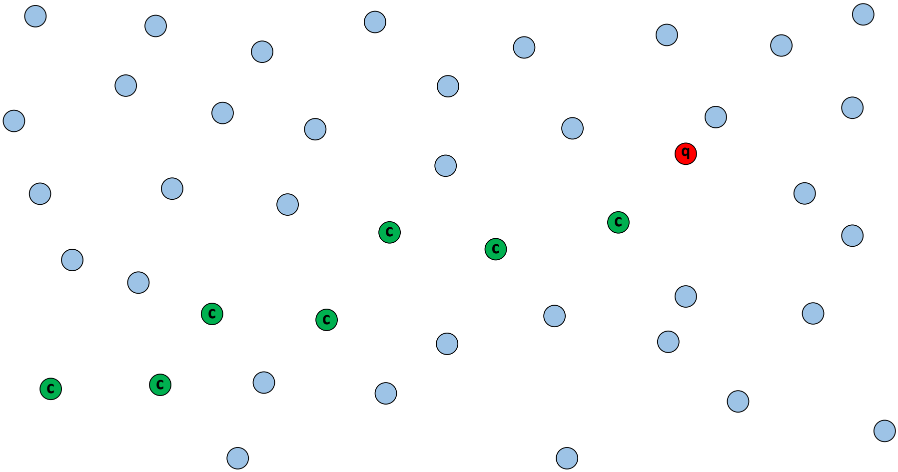
Beam search



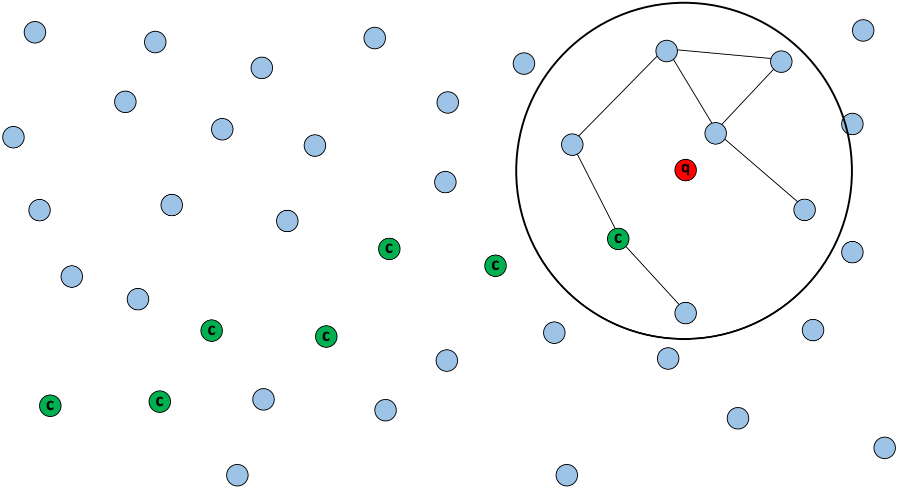
Beam search



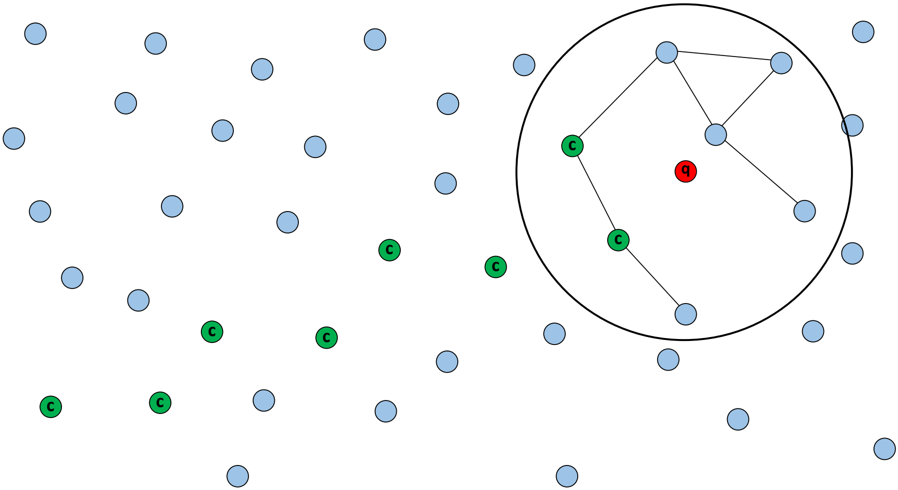
Beam search



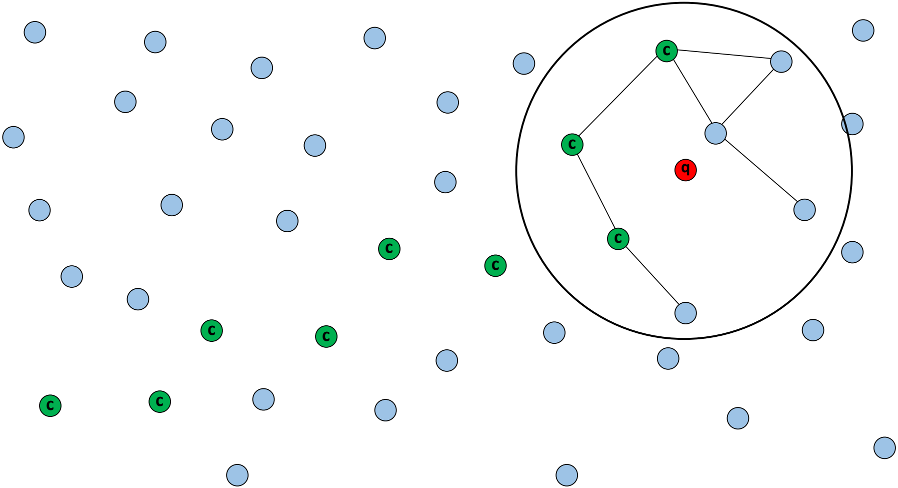
Beam search



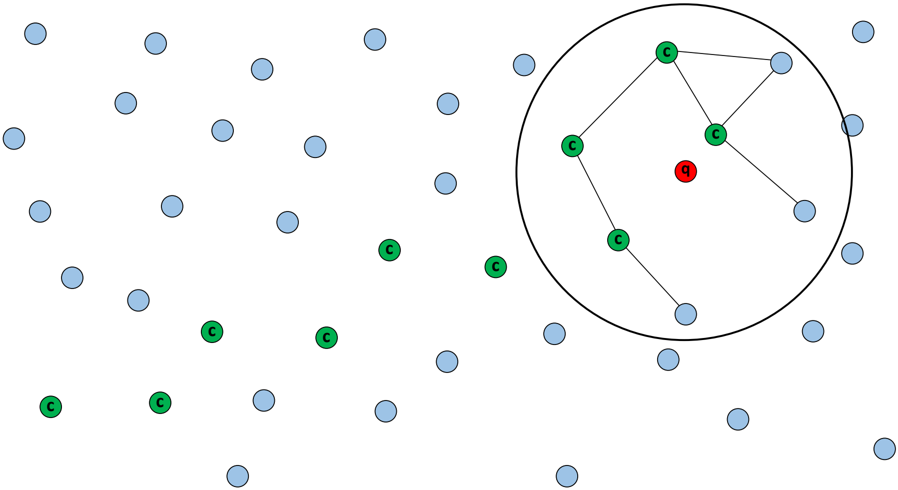
Beam search



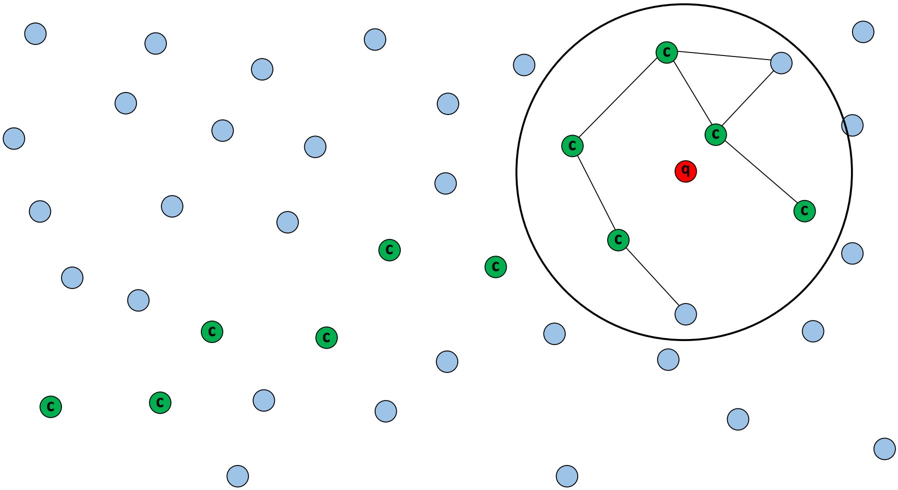
Beam search



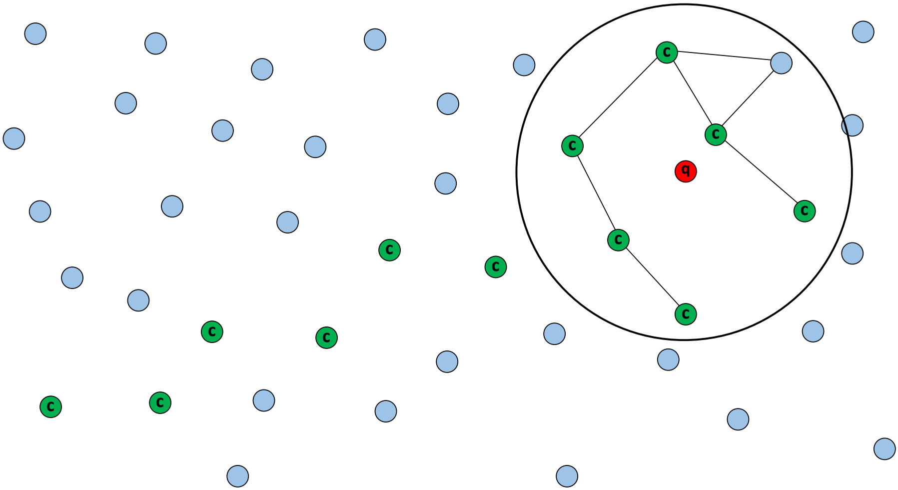
Beam search



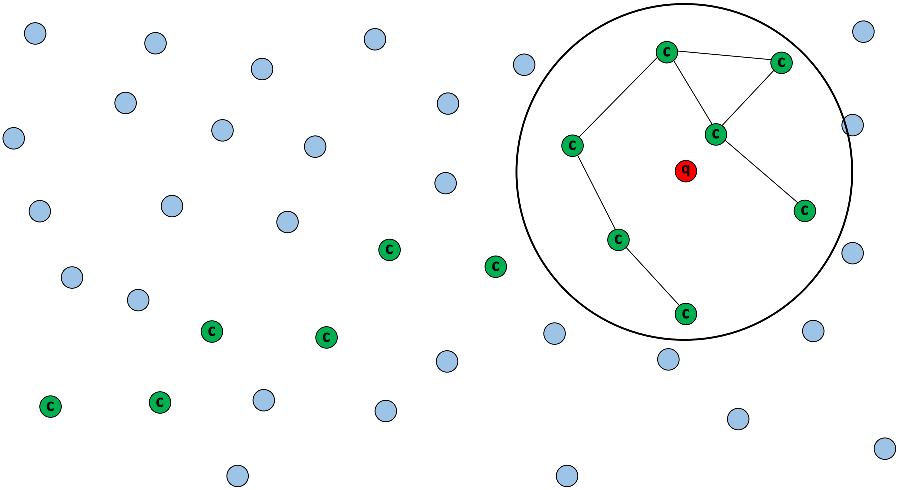
Beam search



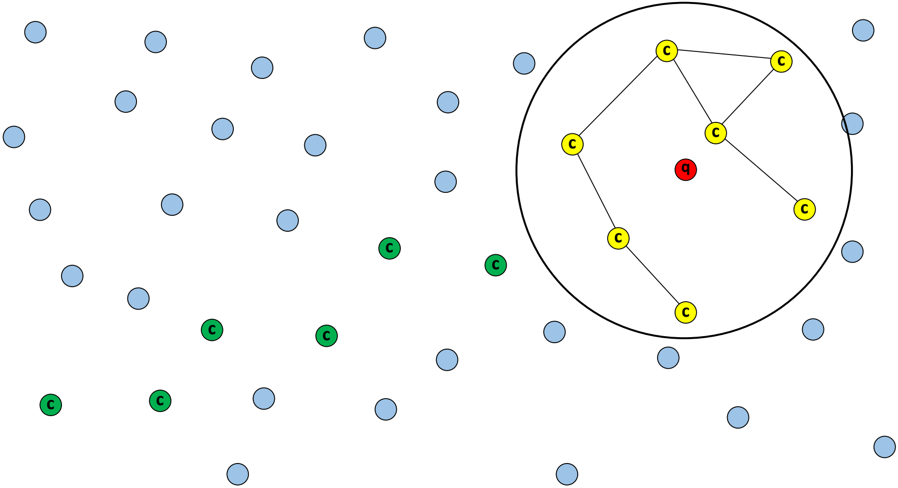
Beam search



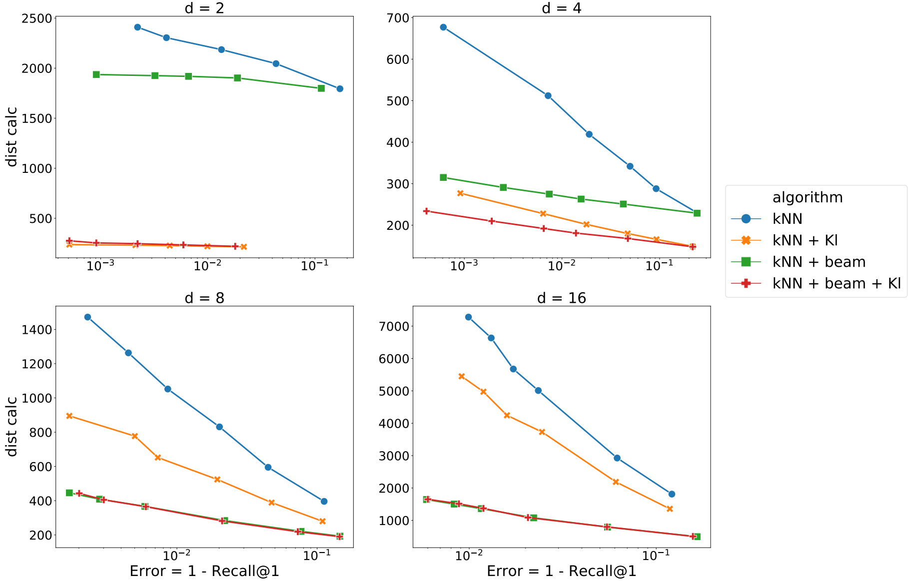
Beam search



Beam search



Synthetic uniform datasets



Uniformization and dimensionality reduction

- Our theoretical guarantees hold for uniform data
- For a general dataset, we can map it to a smaller dimension and make it more uniform while trying to preserve the neighborhoods[2]
- We perform beam search in the lower-dimensional space and then evaluate the candidates in the original space
- This allows to significantly improve the quality of plain NN graphs supplied with long edges
- See details in our paper

[2] Sablayrolles, A., Douze, M., Schmid, C., Jégou, H. "Spreading vectors for similarity search". ICLR 2019.

Thank you!