Learning to Route in Similarity Graphs

Dmitry Baranchuk

joint work with Dmitry Persiyanov, Anton Sinitsin and Artem Babenko

Yandex

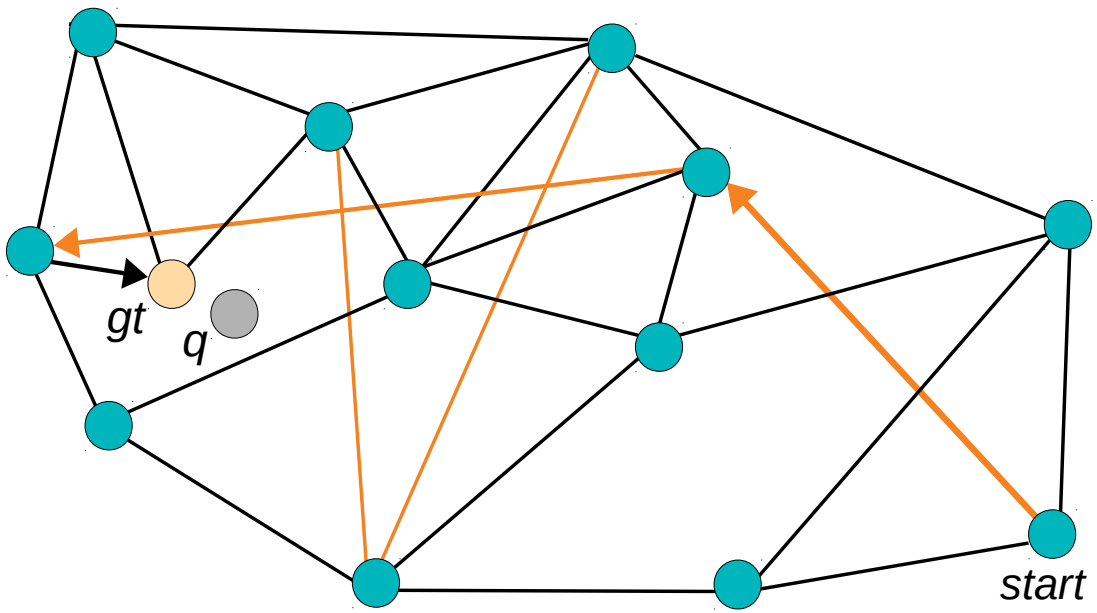
Overview

The Budgeted Nearest Neighbor Search Problem

- $\{x_1, ..., x_N\} \subset \mathbb{R}^D$ — search database
- $q \in \mathbb{R}^D$ — query
- DCS — maximal number of distance computations
- $Recall@1$ — a rate of queries for which the actual nearest neighbor is successfully found

Similarity Graphs

- **Vertices** correspond to the database items
- **Edges** connect (mostly) nearest neighbors



- Several state-of-the-art methods exist e.g. HNSW[1], NSG[2]

[1] Malkov, Y., Yashunin, D. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. TPAMI 2018

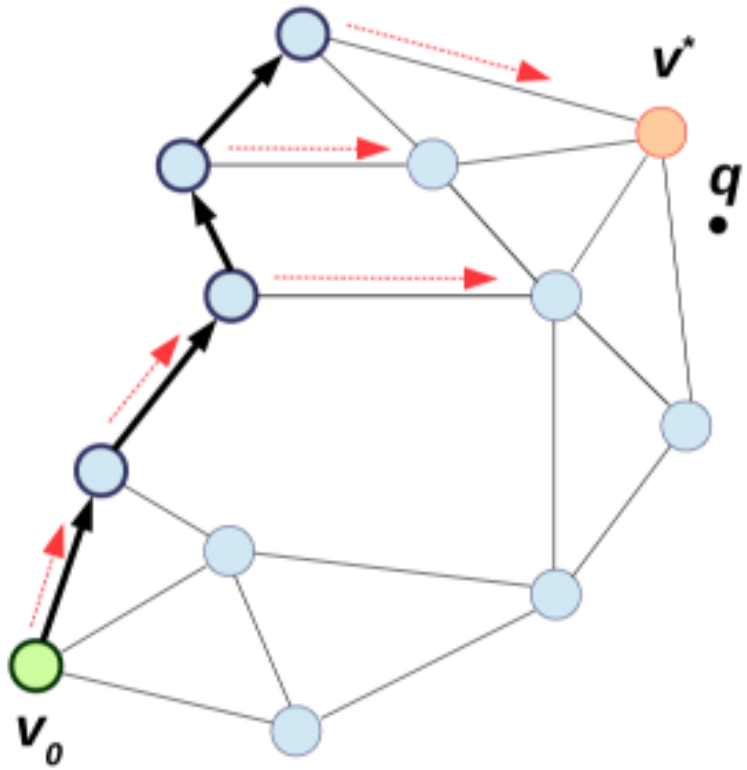
[2] Cong Fu, Chao Xiang, Changxu Wang, and Deng Cai. Fast approximate nearest neighbor search with the navigating spreading-out graph. PVLDB 2019

Routing Algorithms

- **Greedy routing**: Pick the best neighbor of the current vertex
- **Beam search**: Expand the most promising vertex in the candidate pool
- **Our method**: Learn a routing algorithm directly from data

Learning to Route in Similarity Graphs

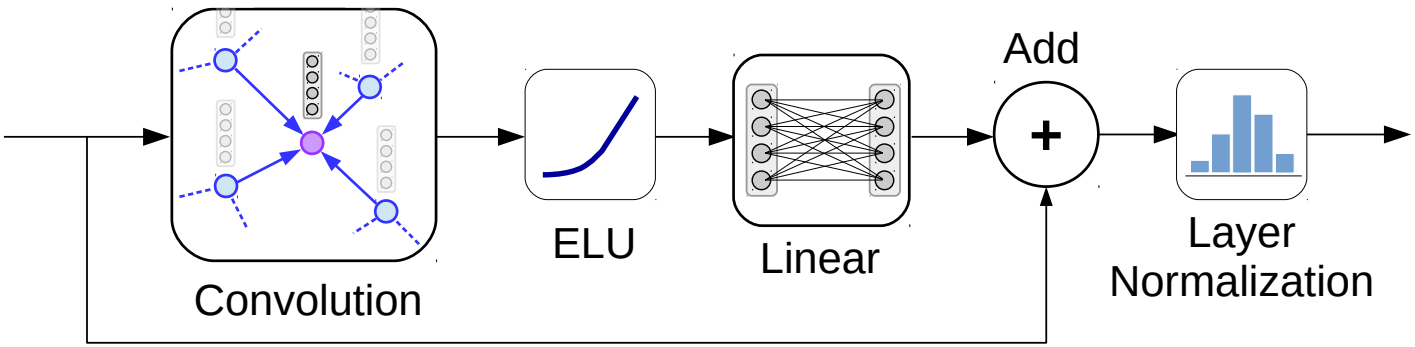
1. **Imitation Learning**: Train the agent to imitate expert decisions
2. **Agent** is a beam search based on learned vertex representations
3. **Expert** encourages the agent to follow a shortest path to the actual nearest neighbor v^*



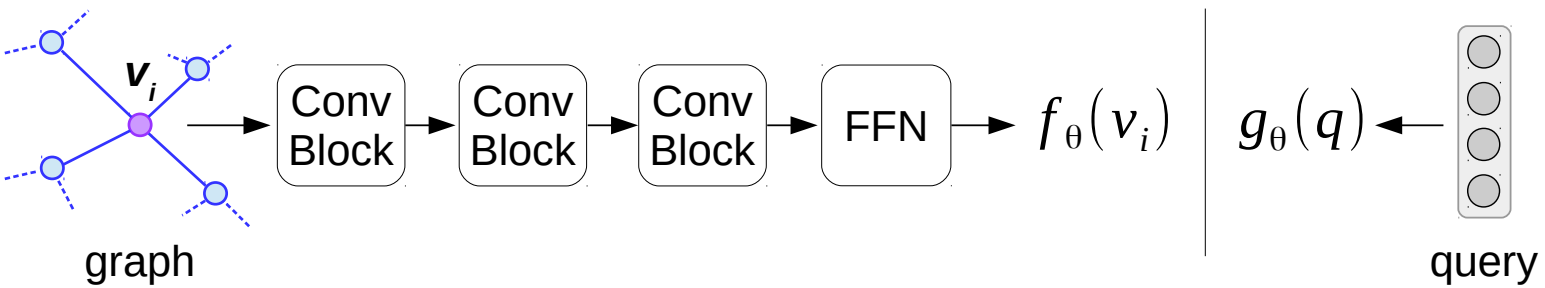
Ross, S., Gordon, G. J., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. AISTATS 2011

Model Architecture

Graph Convolutional Network learns representations for vertices that account for the underlying structure of the similarity graph



Graph Convolutional Block



Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. ICLR 2017

Evaluation

- Datasets with 10^5 points
- No additional cost in run-time
- PyTorch implementation[3]

DCS budget	Vertex Representations	SIFT100K *Recall*@1	DEEP100K *Recall*@1	GloVe100K *Recall*@1
128	Original	0.239	0.386	0.198
	Learned	**0.371**	**0.474**	**0.305**
256	Original	0.672	0.795	0.400
	Learned	**0.799**	**0.811**	**0.526**
512	Original	0.936	0.940	0.582
	Learned	**0.949**	**0.945**	**0.676**

Search performance *Recall*@1 for distance computation (DCS) budgets

[3] https://github.com/dbaranchuk/learning-to-route