Leonid Boytsov, Eric Nyberg

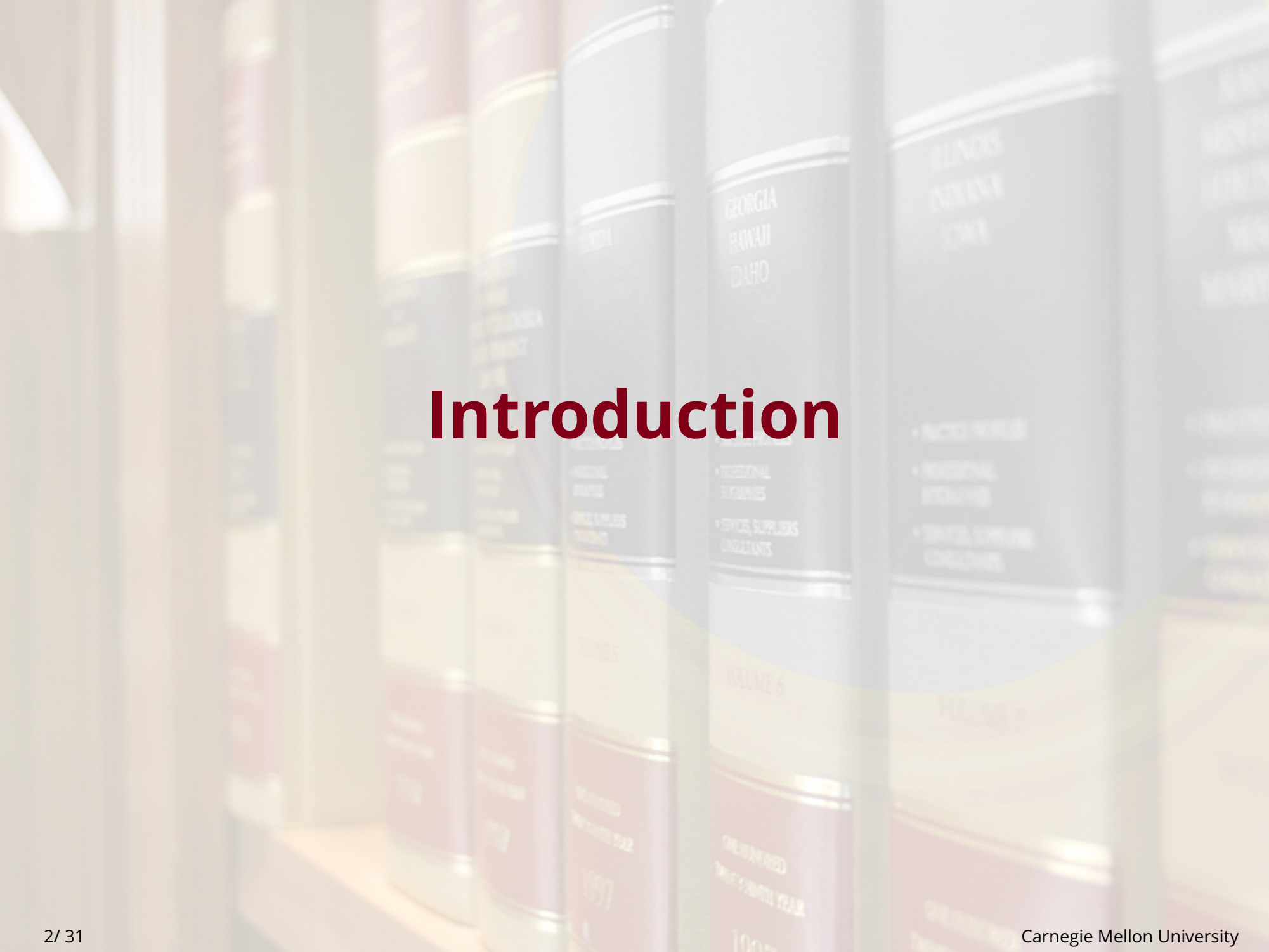
Carnegie Mellon University

1. Pruning Algorithms for Low-Dimensional Non-Metric k-NN Search: A Case Study

2. Accurate and Fast Retrieval for Complex Non-Metric Data via Neighborhood Graphs

Acknowledgements

- Part of NMSLIB project:
  - Bileg Naidan
  - Yury Malkov
  - David Novak
- NSF grant #1618159: “Matching & ranking via proximity graphs: applications to QA & beyond”



Introduction

Motivation I

- A well-known problem of *k*-NN search (IR, ML, NLP)
- Metric search is a well-researched problem
- Little attention to non-metric search

Motivation II

- We aim to fill this gap
- Focusing on
  - **Generic** methods
  - **Accurate** retrieval
  - **Challenging** distances

Terminology

- **Input:** We have a set of data points or objects
- **Query:** A query is a new object
- **Task:** Find *k* most similar objects
- **Distance:** Smaller for more similar objects

Roadmap

- Data sets & distances
- Evaluation of hardness
- Indexing methods for non-metric similarities
  - Tree-based
  - Neighborhood graphs

Data, Distances, Hardness

Data

Name	max. # of records	dimensionality	Source
RandHist-d	0.5×10^6	$d \in \{8, 32\}$	Random hist.
RCV-d	0.5×10^6	$d \in \{8, 128\}$	LDA histograms RCV1
Wiki-d	2×10^6	$d \in \{8, 128\}$	LDA histograms Wikipedia
Manner	1.46×10^5	1.23×10^5	Yahoo Answers

Distances

Euclidean (L_2)	$\|\|x-y\|\|_2 = \left[\sum_i (x_i - y_i)^2\right]^{1/2}$
L_p ($p > 0$)	$\left[\sum_{i=1}^{m} (x_i - y_i)^p\right]^{1/p}$
Squared L_2	$\|\|x-y\|\|_2^2 = \sum_i (x_i - y^i)^2$
Cosine	$1 - \left(\sum_i x_i y_i\right) \left(\|\|x\|\|_2 \|\|y\|\|_2\right)^{-1}$
KL-divergence	$\sum_{i=1}^{m} x_i \log \frac{x_i}{y_i}$
Itakura-Saito dist.	$\sum_{i=1}^{m} \left[\frac{x_i}{y_i} - \log \frac{x_i}{y_i} - 1\right]$
Rényi divergence	$\frac{1}{\alpha-1} \log \left[\sum_{i=1}^{m} x_i^{\alpha} y_i^{1-\alpha}\right], \alpha > 0, \alpha \neq 1$
BM25	$-\sum_{x_i=y_i} \mathrm{TF}_q(x_i) \cdot \mathrm{TF}_d(y_i) \cdot \mathrm{IDF}(y_i)$

VP-tree Indexing

Select results on Wiki-128:

	Recall	Speed-up over BF.
$L_p(p = 0.125)$	**0.07**	14845
Cosine dist.	0.73	55
Rényi div. ($\alpha = 2$)	0.71	55
KL-div.	0.56	41
Itakura-Saito	**0.14**	384

Distance Approximations

- Average-based symmetrization: $\frac{d(x,y)+d(y,x)}{2}$
- Min-based symmetrization: $\min\left(d(x,y), d(y,x)\right)$
- Argument-reversed distance: $d(y,x)$
- The Euclidean distance
- A learned metric and/or *non*-metric distance

Filter-and-Refine via Approximation

Select results for Rényi div. $\alpha = 2$:

Data set	Symmetrization		Distance learning (best result)	
	k_c (cand. k)	Recall reached	k_c (cand. k)	Recall reached
RCV-8	20	99	640	100
Wiki-8	20	99	640	99
RandHist-8	160	100	320	99
RCV-128	80	99	20480	66
Wiki-128	80	99	20480	87
RandHist-32	2560	99	20480	100

Takeaways

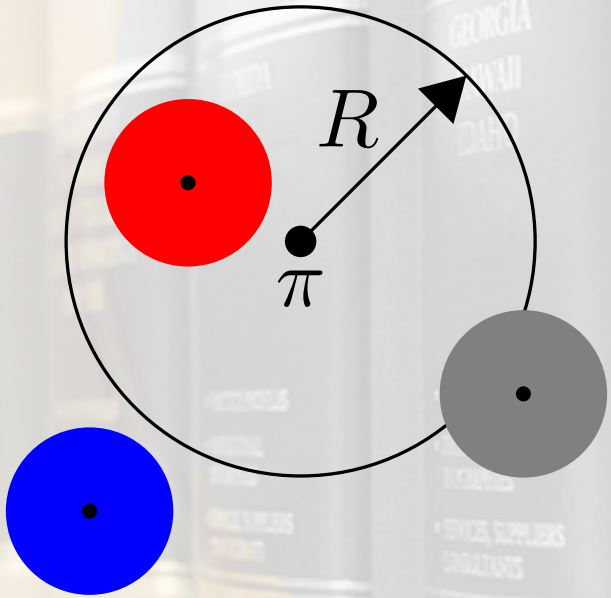
- Non-metric data cannot be handled using metric search methods directly
- Filter-and-refine entails either efficiency or accuracy penalty
- Symmetrization may work in some cases
- Metrization results are quite poor

Adapating Tree Based Methods

Hierarchical Space Decomposition (VP-tree)

Hierarchical Space Decomposition (VP-tree)

Three Types of the Query Ball

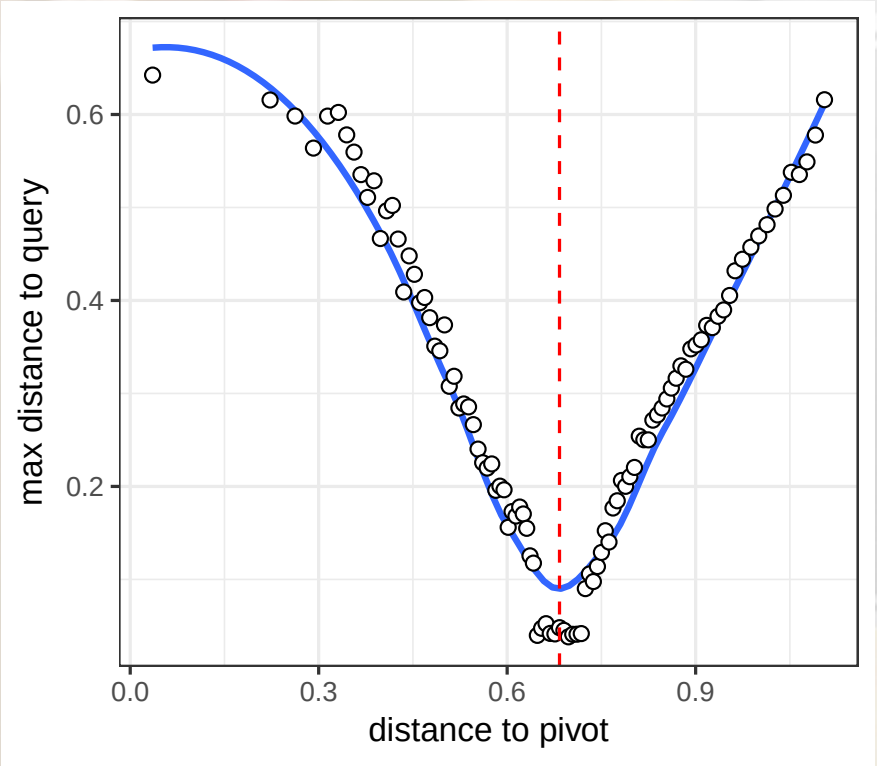


Gray ball case:

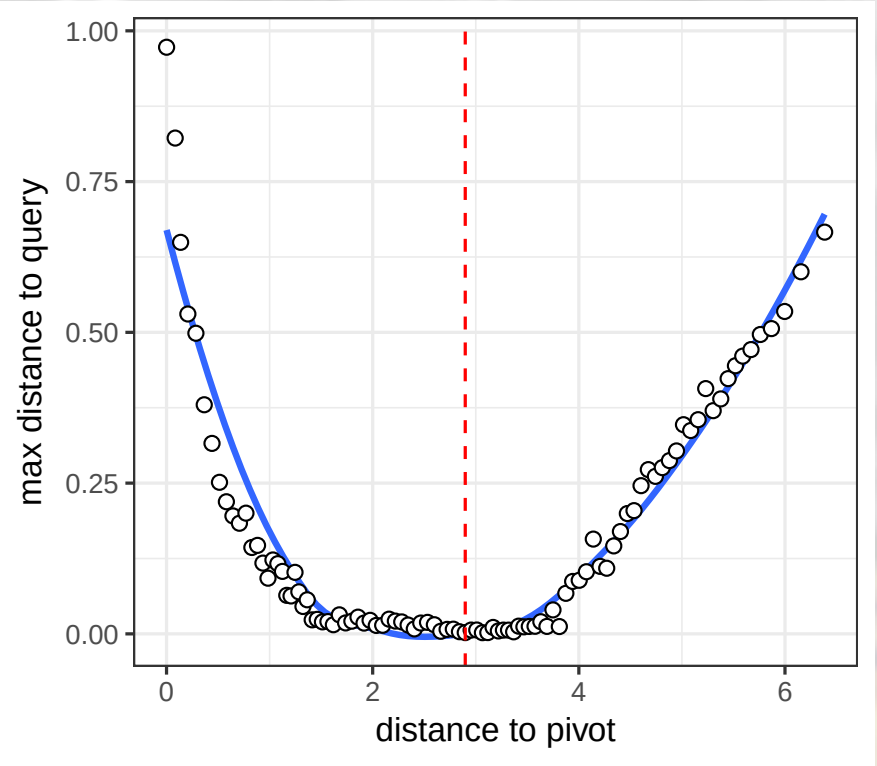
$$r \geq D_{\pi,R}(d(\pi, q)) = |R - d(\pi, q)|$$

Generic Decision Function $D_{\pi,R}$

Euclidean



KL-divergence



Gray ball case:

$$\text{max. dist. to query} \geq D_{\pi,R}(\text{dist. to pivot})$$

Adapting Metric Methods to non-Metric Spaces

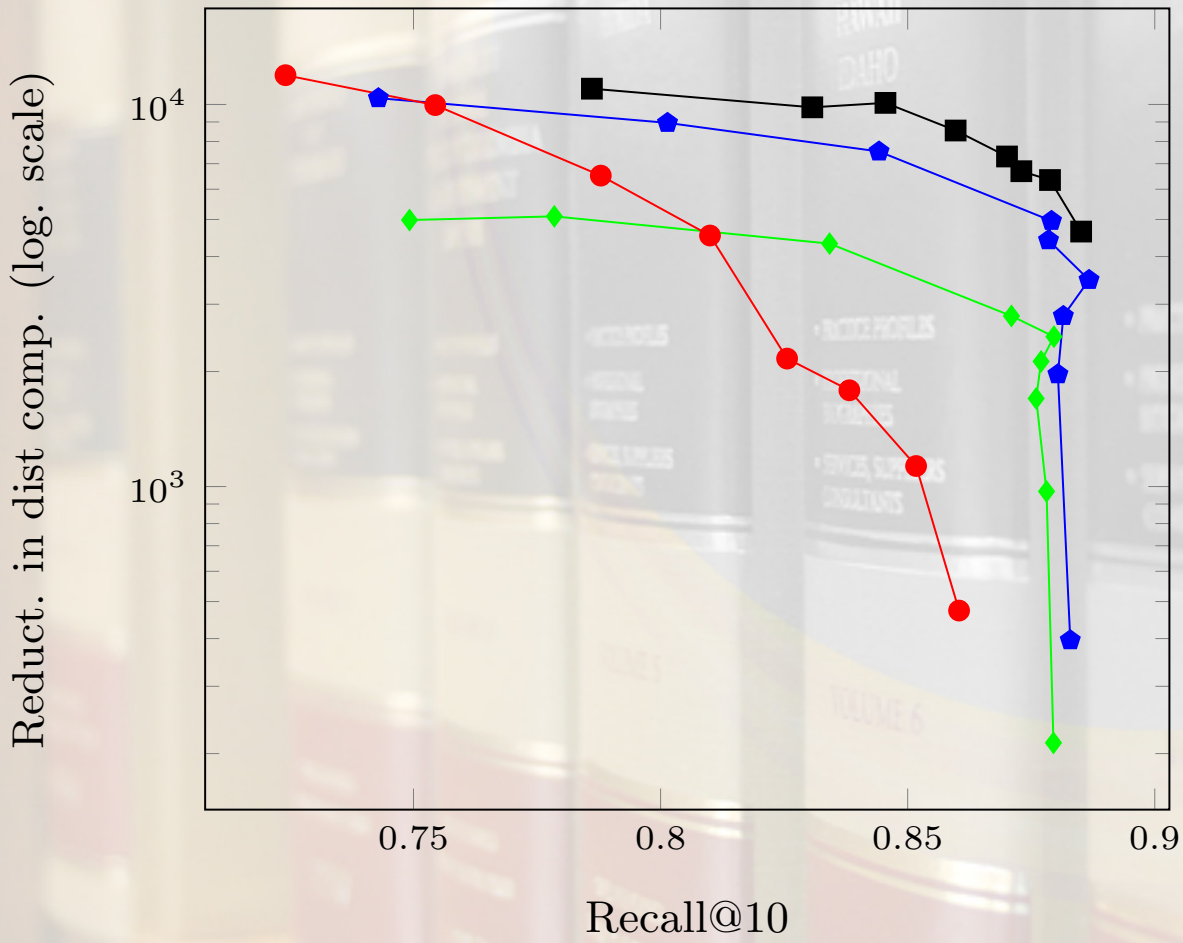
- Learning a piece-wise linear decision-making function
- Stretching the distance using concave mapping (TriGen)

Two Simple Symmetrization Approaches

- **TriGen0**
  - Prune using the symmetrized version
  - Update neighbors using the original distance
- **TriGen1**
  - Compute symmetrized distance only in internal nodes
  - Update query radius using an upper bound

Case in Point: TriGen1 vs TriGen0

TriGen1 (blue) performs fewer distance computations than TriGen0 (green):



Rényi divergence $\alpha = 0.25$, Wikipedia-8

Key Results

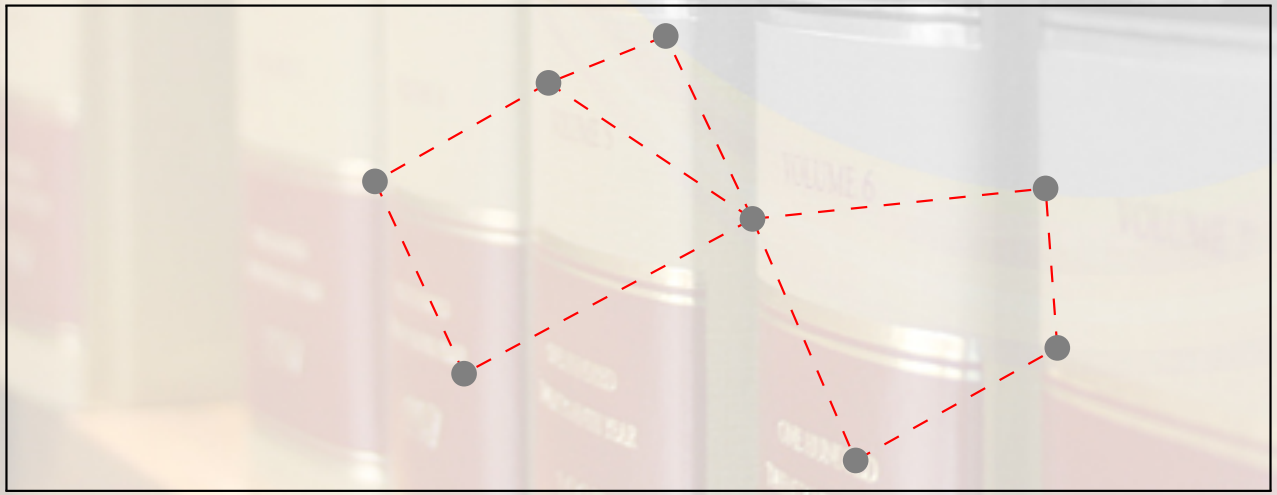
- Symmetrization approach matters
- Original TriGen distance mapping is not very efficient and it hurts performance a lot
- The best approach may be a hybrid between piece-wise linear pruner and TriGen

Experiments with Neighborhood Graphs

Proximity/Neighborhood Graphs

- Data points are nodes
- Sufficiently close points are connected
- A search procedure is a semi-greedy traversal of the graph

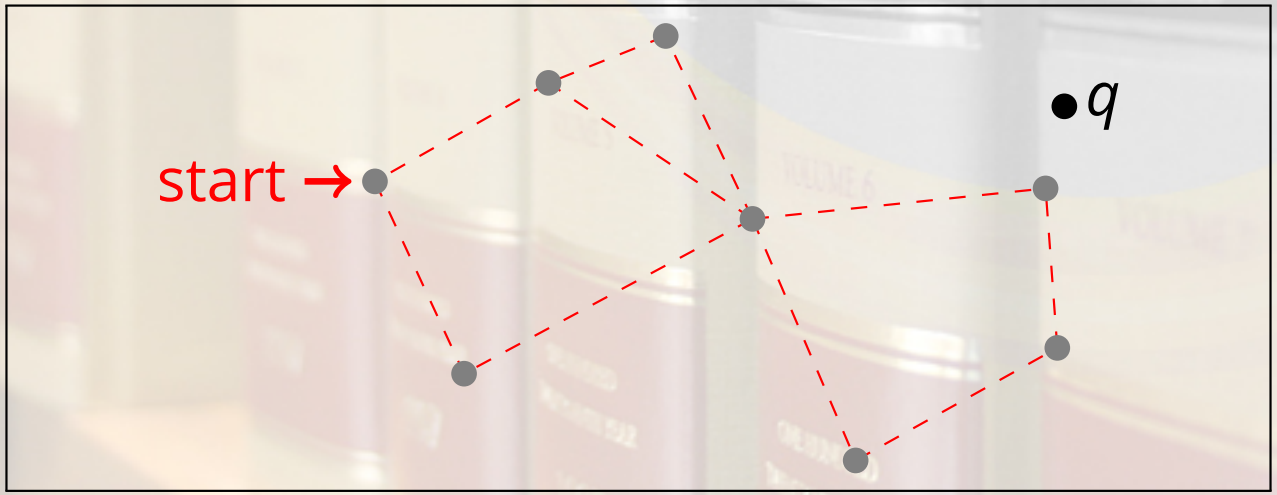
Example two-nearest neighbor graph:



Proximity/Neighborhood Graphs

- Data points are nodes
- Sufficiently close points are connected
- A search procedure is a semi-greedy traversal of the graph

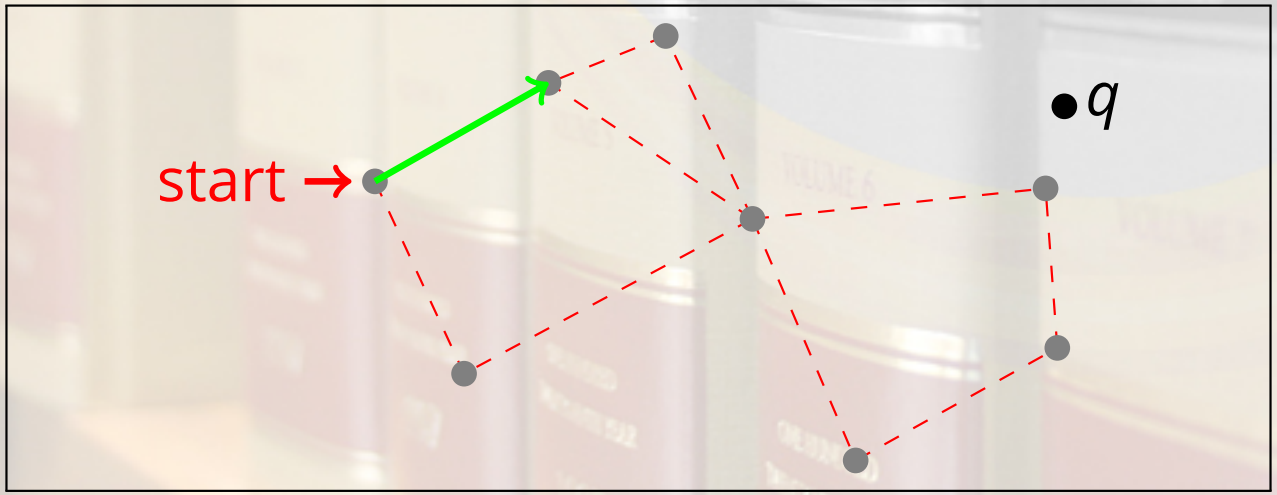
Example two-nearest neighbor graph:



Proximity/Neighborhood Graphs

- Data points are nodes
- Sufficiently close points are connected
- A search procedure is a semi-greedy traversal of the graph

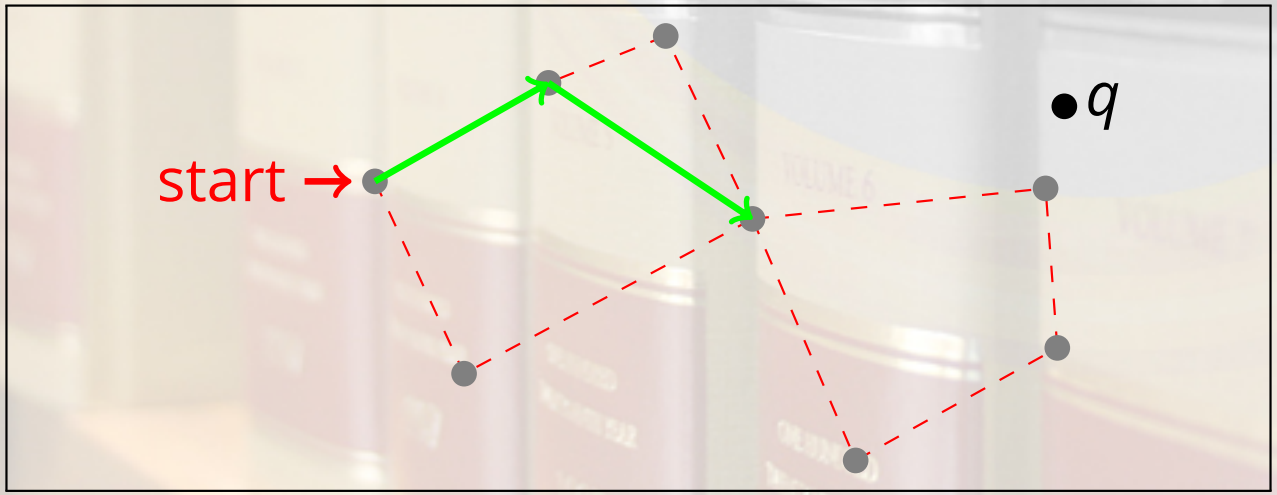
Example two-nearest neighbor graph:



Proximity/Neighborhood Graphs

- Data points are nodes
- Sufficiently close points are connected
- A search procedure is a semi-greedy traversal of the graph

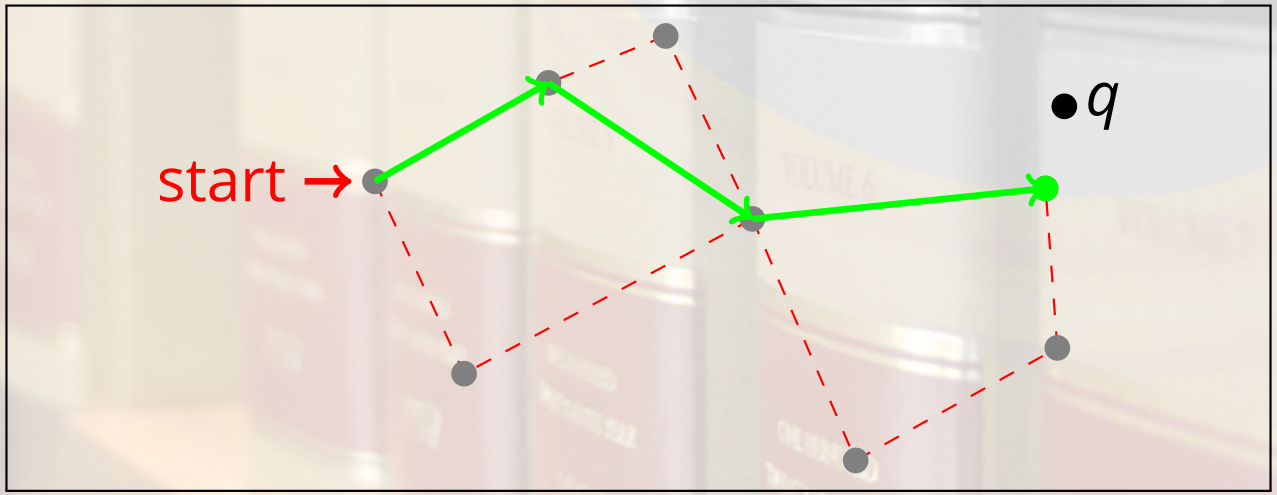
Example two-nearest neighbor graph:



Proximity/Neighborhood Graphs

- Data points are nodes
- Sufficiently close points are connected
- A search procedure is a semi-greedy traversal of the graph

Example two-nearest neighbor graph:



Proxy Functions for Indexing

Build a neighborhood graph using:

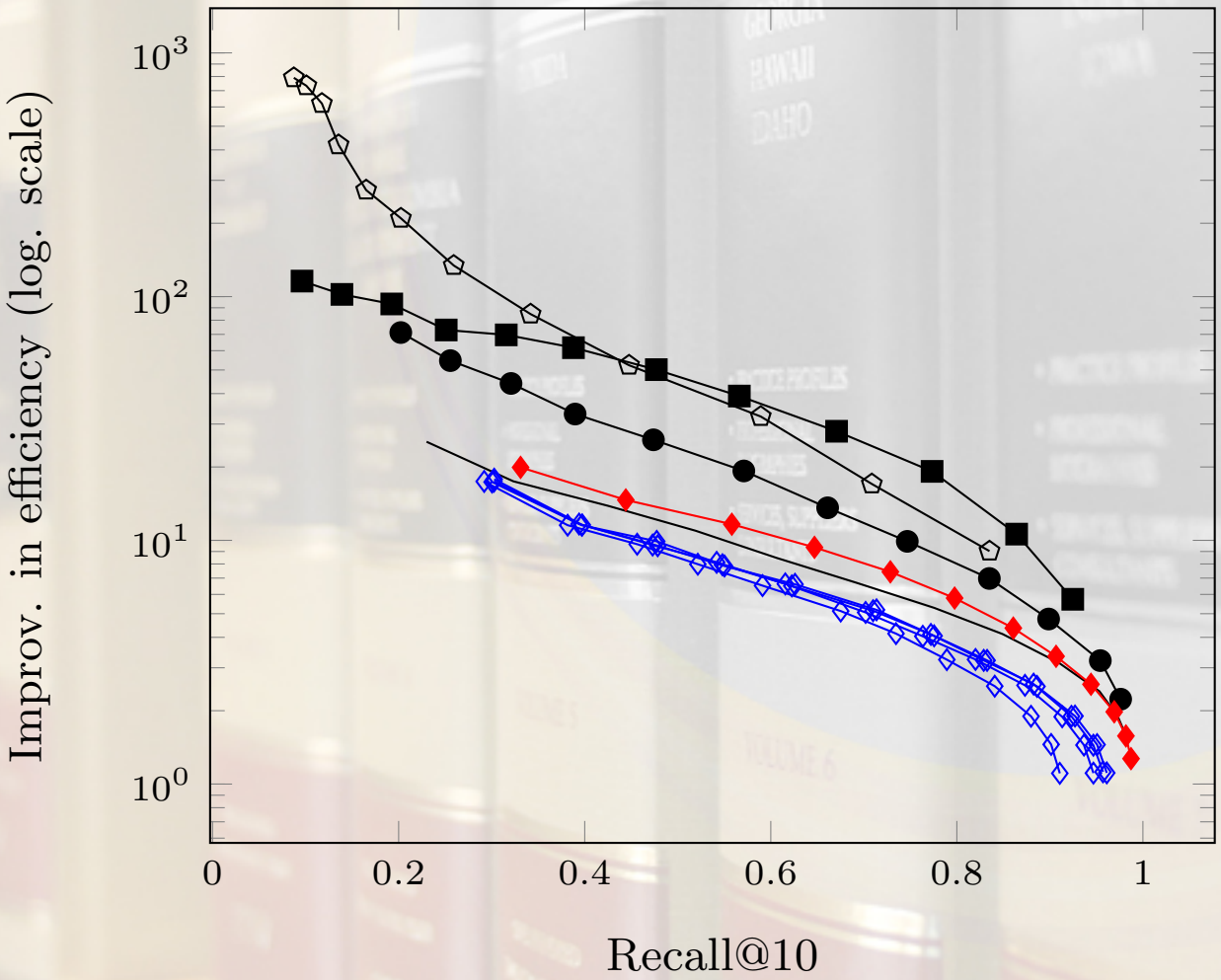
- min-based symmetrization
- average-based symmetrization
- reversed-argument distance
- Euclidean distance

Legend & Experiment Summary

- **Red**: no symmetrization
- **Blue**: full filter-and-refine symmetrization
- **Solid black**: index-time proxying
- **Solid black circle**: averaged-based distance index-time
- **Solid black box**: reversed-argument distance index-time

Case 1

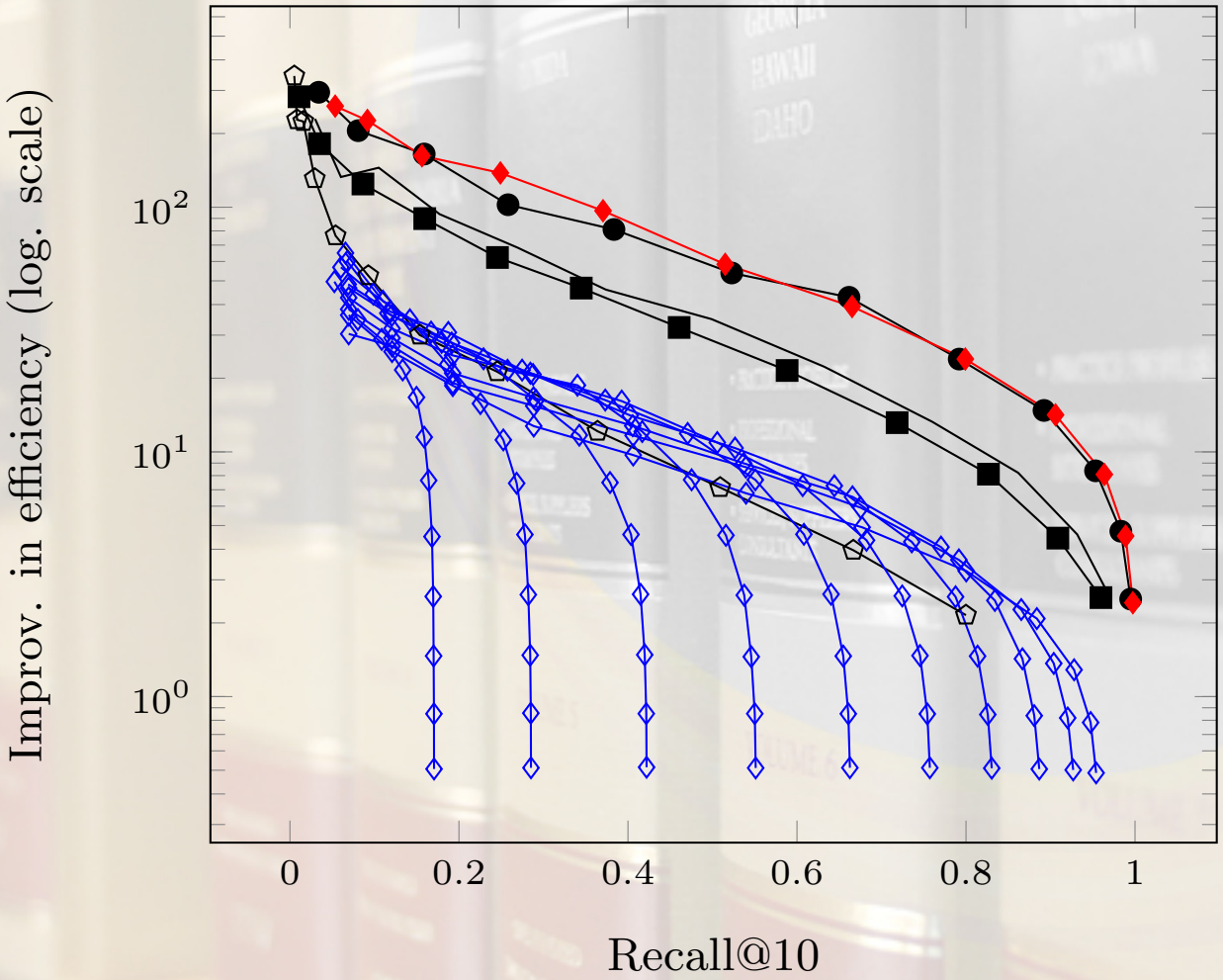
10x speed-up with index-time proxy:



Itakura-Saito dist. RCV-128

Case 2

Reversal of arguments **can** hurt:



Itakura-Saito dist. RandHist-32

Takeaways

- The neighborhood graph worked well for substantially non-symmetric distance
- Index-time proxying, e.g., symmetrization, can be beneficial
- Yet, full filter-and-refine symmetrization is likely suboptimal

Conclusion

Concluding Remarks

- Non-Metric *k*-NN search can be challenginge
- We have working solution (NMSLIB):
  - Non-metric adaptations of metric methods
  - Neighborhood graphs
  - Neighborhood approximation index

Thank you!