L2Knng: Fast Exact K-Nearest Neighbor Graph Construction with L2-Norm Pruning

David C. Anastasiu and George Karypis
Department of Computer Science and Engineering University of Minnesota

ACM/2015

Paulo Henrique da Silva

Prof. Dr. Wellington Santos Martins

{paulohsilva, wellington}@inf.ufg.br

Similarity Search

21 November, 2018

Seminários

Sumary

- Introduction
- Related Works
- Data and Methods
- Results
- Conclusions
- Future Works

Introduction

- Context
  - *K*-nearest neighbor search, also known as similarity search, involves finding the top *k* results (e.g., to 10 most similar) to a given query
  - The *k*-NNG is a directed graph where vertices correspond to the objects and edges connect their neighbors

Introduction

- Problem
  - Given a set *D* in a *r*-dimensional space and a query *q*, find the *k* points in *D* with the smallest distances *dist(q, p)*
  - High dimensional sparse datasets
  - Represented by weighted vectors
  - Cosine function to measure vector similarity

Introduction

- Approaches
  - Exact methods, which return the *k* most similar objects of a given object
  - Approximate methods, the *k* neighbors of each object do not necessarily correspond to the *k* most similar objects

Introduction

- Motivation
    - Relevance in many real-world applications
        - Information Retrieval
        - Clustering
        - Online advertising
        - Recomender systems

Introduction

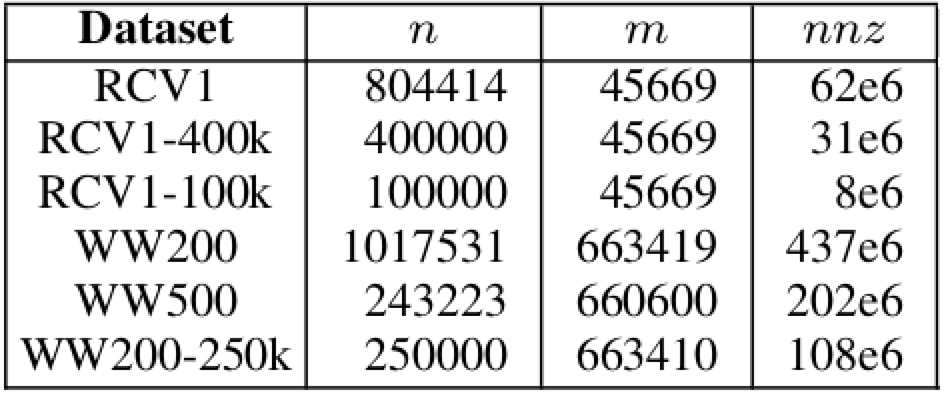
- Objectives
  - Introduce L2KnngApprox to obtain approximate initial solution for *k*-NNG
  - Introduce L2Knng to solve the exact cosine similarity *k*-NNG
  - New filtering methods to prune objects
  - Improve baselines

Related Works

- Bayardo et al. [2007] – Developed several strategies to prune the search space
- Dong et al. [2011] – Iterative improvements of an initial random k-NNG by considering neighbors' neighbors as potencial neighbors
- Park et al. [2014] – Approximate approach that focus on object pairs that have high-weight features in common
- Anastasiu and Karapis [2014] – Similarity search with new pruning strategies

Data and Methods

- Datasets
  - Textual datasets
  - Standard pre-processing tokenization and lemmatization
  - Represented as *tf-idf* weighted vectors



Dataset	n	m	nnz
RCV1	804414	45669	62e6
RCV1-400k	400000	45669	31e6
RCV1-100k	100000	45669	8e6
WW200	1017531	663419	437e6
WW500	243223	660600	202e6
WW200-250k	250000	663410	108e6

Data and Methods

- L2KnngApprox
  - Builds an approximate solution to the problem
  - Iteratively enhances the initial *k*-NNG by looking for new candidates in each neighbor's neighborhood

Data and Methods

- L2KnngApprox
  - Sorts the feature vectors in decreasing weight order
  - Define a minimum similarity between query and its *k* neighbors
  - Builds a set of *m (m > k)* initial neighbors based on high-weight features of vectors and prefix filtering
  - Compute the exact similariy of *m* candidates with query object and selects the initial *top*-*k* neighbors
  - Improves *k*-NNG with the similarity of its neighbors' neighbors

Data and Methods

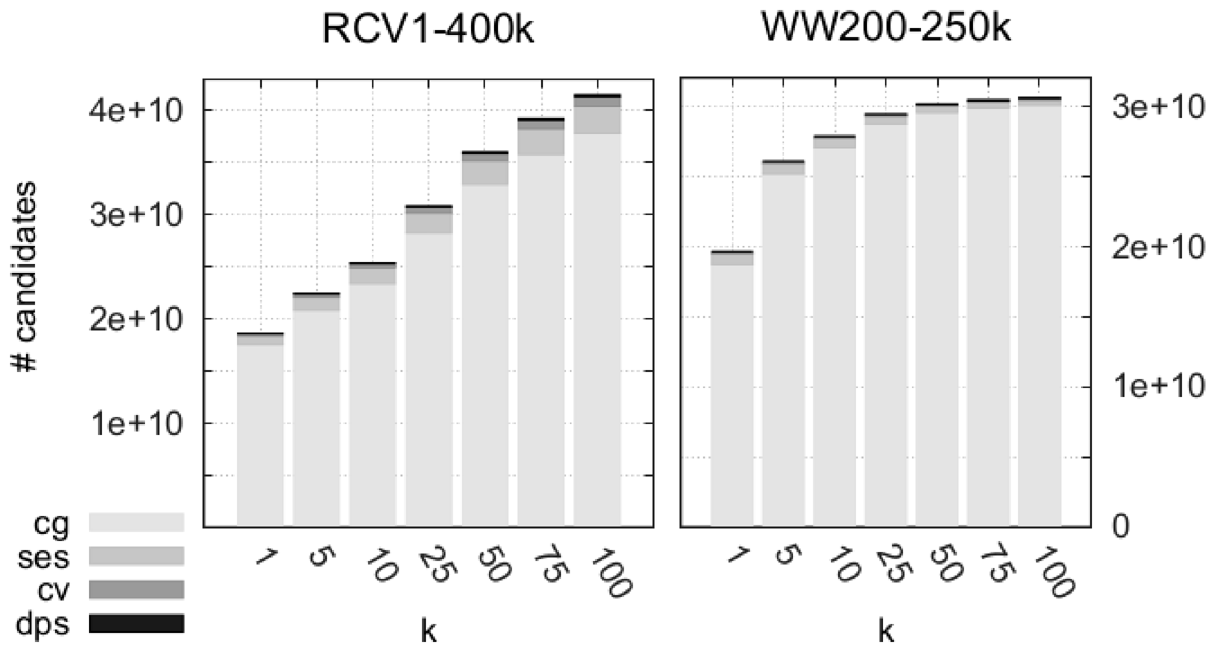
- L2Knng
  - Solves the exact cosine similarity *k*-NNG construction problem
  - Filtering candidates by the suffix filter to pruning not true neighbors
  - Reduce the full vector dot-products

Data and Methods

- L2Knng
  - Uses the approximate initial *k*-NNG
  - Computes the suffix similarity between the candidates and query and accumulate with the prefix similarity
  - Accumulate exact similarity with un-pruned candidates
  - Generate the final *k*-NNG with the top *k* neighbors

Results

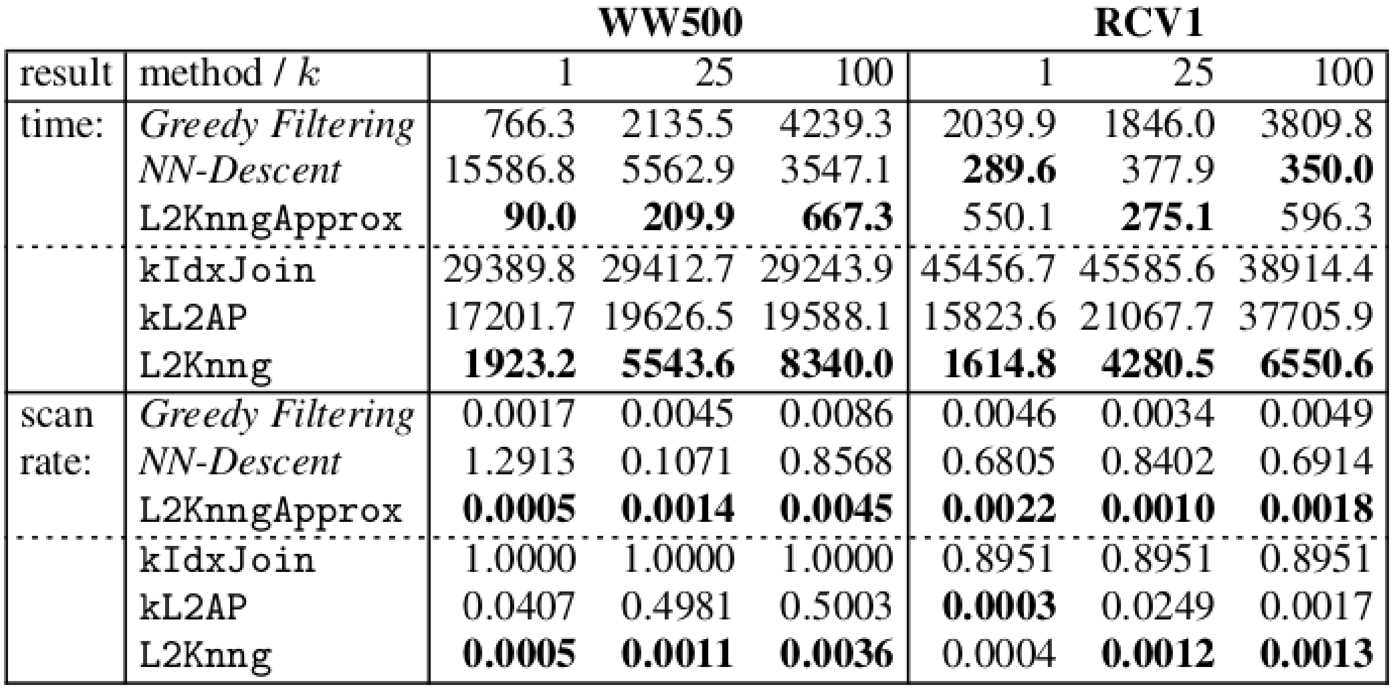
- Pruning effectiveness



cg (candidate generation), ses (suffix estimate score), cv (candidate verification), dps (full dot-product)

Results

- Execution time and scan rate for competing algorithms



		WW500			RCV1		
result	method / *k*	1	25	100	1	25	100
time:	*Greedy Filtering*	766.3	2135.5	4239.3	2039.9	1846.0	3809.8
	NN-Descent	15586.8	5562.9	3547.1	**289.6**	377.9	**350.0**
	L2KnngApprox	**90.0**	**209.9**	**667.3**	550.1	**275.1**	596.3
	kIdxJoin	29389.8	29412.7	29243.9	45456.7	45585.6	38914.4
	kL2AP	17201.7	19626.5	19588.1	15823.6	21067.7	37705.9
	L2Knng	**1923.2**	**5543.6**	**8340.0**	**1614.8**	**4280.5**	**6550.6**
scan rate:	*Greedy Filtering*	0.0017	0.0045	0.0086	0.0046	0.0034	0.0049
	NN-Descent	1.2913	0.1071	0.8568	0.6805	0.8402	0.6914
	L2KnngApprox	**0.0005**	**0.0014**	**0.0045**	**0.0022**	**0.0010**	**0.0018**
	kIdxJoin	1.0000	1.0000	1.0000	0.8951	0.8951	0.8951
	kL2AP	0.0407	0.4981	0.5003	**0.0003**	0.0249	0.0017
	L2Knng	**0.0005**	**0.0011**	**0.0036**	0.0004	**0.0012**	**0.0013**

Best results are emphasized in bold

Conclusions

- Introduction of new pruning bounds
- Estrategies to avoid full similarity computation for most object pairs
- Performance increased with the pruning of candidates
- L2Knng achieves improvement against exact baselines
- L2KnngApprox is faster than approximate baselines

Future Works

- Evaluate the efficiency of ℓ^2-norm with others similarity function (Dice and Tanimoto)
- Scaling up the number of threads and processors to solve the problem

L2Knng: Fast Exact K-Nearest Neighbor Graph Construction with L2-Norm Pruning

David C. Anastasiu and George Karypis
Department of Computer Science and Engineering University of Minnesota

ACM/2015

Paulo Henrique da Silva
Prof. Dr. Wellington Santos Martins
{paulohsilva, wellington}@inf.ufg.br

Similarity Search

21 November, 2018

Seminários