Cosine Approximate Nearest Neighbors

David C. Anastasiu

Department of Computer Engineering

San José State University

Near Duplicate Detection

SpaceX rocket fails to land on barge

Company never expected to nail this landing, says SpaceX chief Elon Musk

The Associated Press Posted: Mar 04, 2016 3:00 PM ET | Last Updated: Mar 04, 2016 8:46 PM ET

SpaceX has already succeeded in landing a Falcon rocket at an on-shore site near the Cape Canaveral pad where it launched, but it has failed in previous attempts to guide rockets back to ocean platforms. (SpaceX)

Related Stories

- SpaceX rocket launches satellite but botches ocean landing
- Why competition is good for the space race: Bob McDonald
- SpaceX rocket explosion debris likely found by UK Coast Guard

SpaceX has another launch under its belt, but not another rocket landing.

The leftover first-stage booster hit the floating platform hard Friday, said SpaceX chief Elon Musk. The company never expected to nail this landing, he said, because of the faster speed of the booster that was required to deliver the satellite to an extra-high orbit.

- **SpaceX pushes satellite launch, rocket landing to Friday**

SpaceX scored a rocket landing on the ground at Cape Canaveral in December, but has yet to nail a trickier barge landing at sea.

The good news, though, is that the unmanned Falcon 9 rocket successfully hoisted the broadcasting satellite for Luxembourg-based company SES.

It was the fifth launch attempt over the past 1½ weeks; Sunday's try ended with an engine shutdown a split second before liftoff. Friday's sunset launch provided a stunning treat along the coast.

SPACEX LAUNCHES SATELLITE, BUT FAILS TO LAND ROCKET ON BARGE

AP

Space-X's Falcon 9 rocket with the Jason-3 satellite aboard, stands ready for flight at Vandenberg Air Force Base, Calif. on Saturday, Jan. 16, 2016. (Matt Hartman)

Share G+1 Tweet

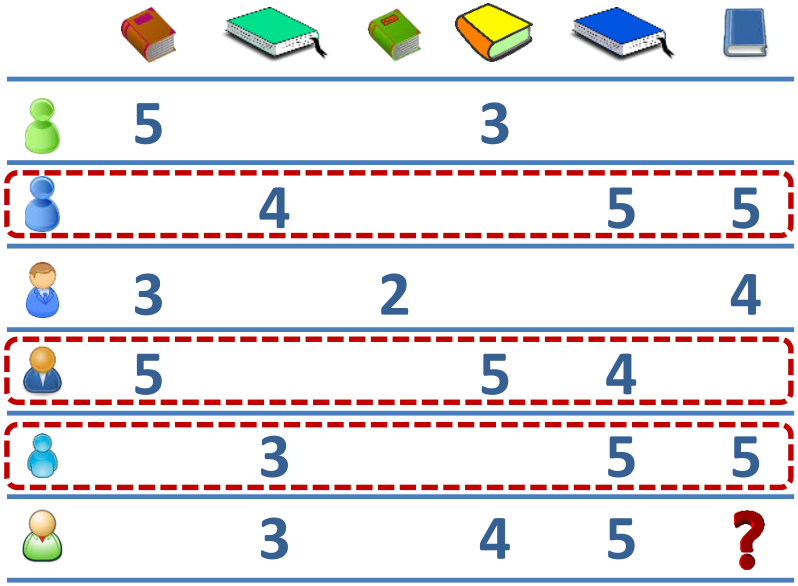
AP

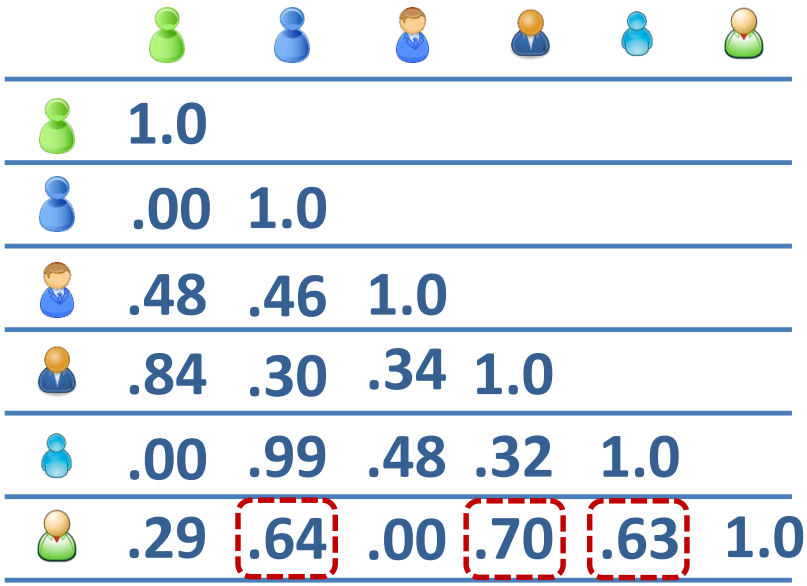
Saturday, March 05, 2016 02:24PM

CAPE CANAVERAL, Fla. -- SpaceX has another launch under its belt, but not another rocket landing.

The leftover first-stage booster hit the floating platform hard Friday, said SpaceX chief Elon Musk. The company never expected to nail this landing, he said, because of the faster speed of the booster that was required to deliver the satellite to an extra-high orbit.

Collaborative Filtering





★★★★★ Loved it
★★★★☆ Liked it
★★★☆☆ It was ok
★★☆☆☆ Disliked it
★☆☆☆☆ Hated it

? = 5

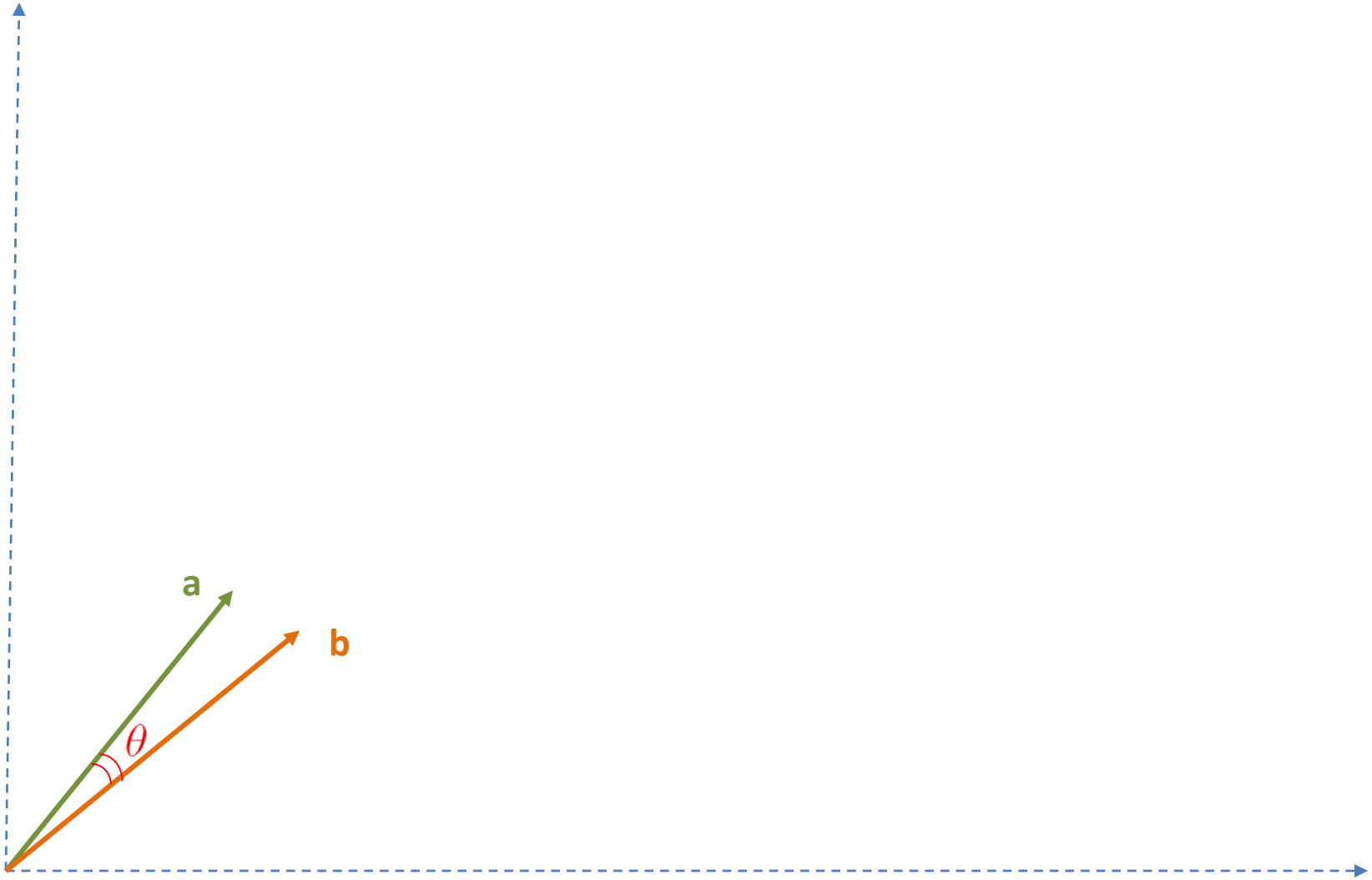
The problem

- For each object d_i from a set D,

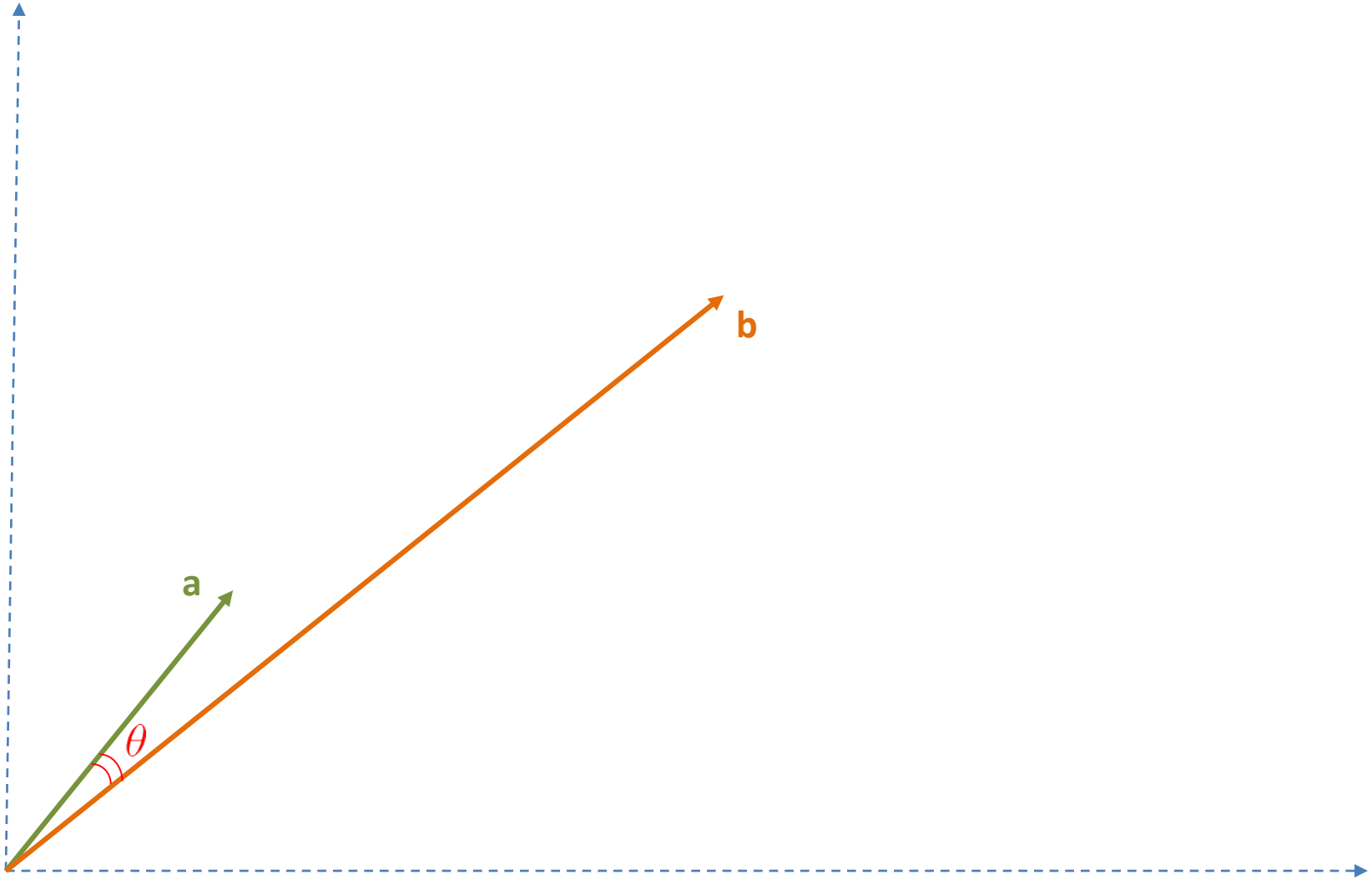
  find all neighbors d_j with $\mathrm{C}(d_i, d_j) \geq \epsilon$.

$$\mathrm{C}(d_i, d_j) = \frac{\langle \mathbf{d}_i, \mathbf{d}_j \rangle}{||\mathbf{d}_i||_2 \times ||\mathbf{d}_j||_2}$$

Cosine invariant to vector length

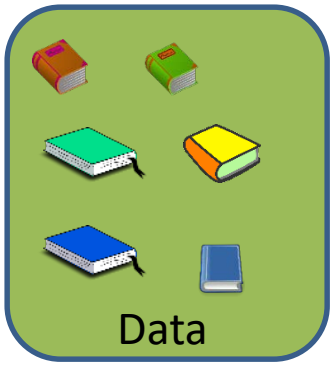


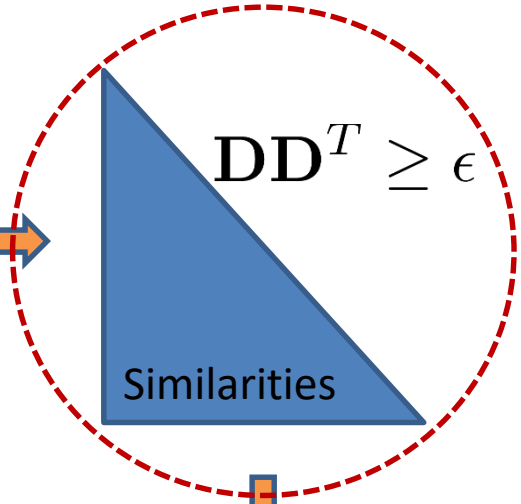
Cosine invariant to vector length



- Normalize all object vectors s.t. $\|\boldsymbol{d_i}\| = 1$

$$\mathrm{C}(d_i, d_j) = \frac{\langle \mathbf{d}_i, \mathbf{d}_j \rangle}{||\mathbf{d}_i|| \times ||\mathbf{d}_j||}$$





Data Mining

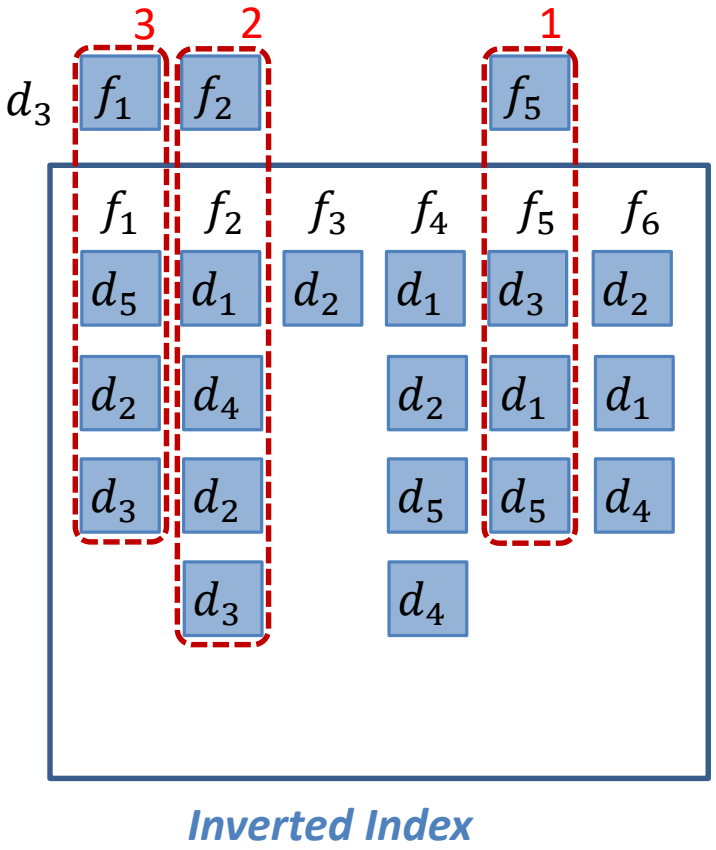
- Threat/Outlier detection
- Duplicate detection
- Recommendation
- Clustering
- Online advertising
- etc…

CANN: An Approximate Solution

- $\mathbf{D}\mathbf{D}^T$ dense in most cases
  - Does not scale for large n

- Approximate solution, in 2 steps:
  1. Construct approximate min-ϵ k-NN graph $\mathcal{G}$
  2. Use $\mathcal{G}$ to construct final min-ϵ NN graph
  - Heuristically choose objects that area likely neighbors:
    - Step 1: objects with high weights in common
    - Step 2: close neighbors of my closest neighbors

Step 1: Approximate min-ϵ k-NN Graph



Inverted Index

For d_3, let $f_5 \geq f_2 \geq f_1$.

$C_{\mu=3} = [d_1, d_5, d_4]$

- Prioritize objects that have high weight features in common with the *query*
  - Create inverted index
  - Sort index lists in decreasing weight order
  - Sort vectors in decreasing weight order
  - Choose $\mu \geq k$ candidates by following lists in order
  - Keep top k neighbors

Step 1: Approximate min-ϵ k-NN Graph

- Improvements:
  - Prefix filtering
  - Bounded similarity computation with pruning
- Prefix filtering:
  - If vectors have no features in common in prefix, their similarity will be $< \epsilon$

$$\langle \mathbf{d}_q, \mathbf{d}_c \rangle = \underbrace{\langle \mathbf{d}_q^{\leq p}, \mathbf{d}_c^{\leq p} \rangle}_{\text{prefix}} + \underbrace{\langle \mathbf{d}_q^{>p}, \mathbf{d}_c^{>p} \rangle}_{\text{suffix}}$$

  - Choose prefix s.t. $\|\mathbf{d}_i\| < \epsilon$

$$\langle \mathbf{d}_q^{>p}, \mathbf{d}_c^{>p} \rangle \leq ||\mathbf{d}_q^{>p}|| \times ||\mathbf{d}_c^{>p}||$$

(Cauchy-Schwarz inequality)

Step 1: Approximate min-ϵ k-NN Graph

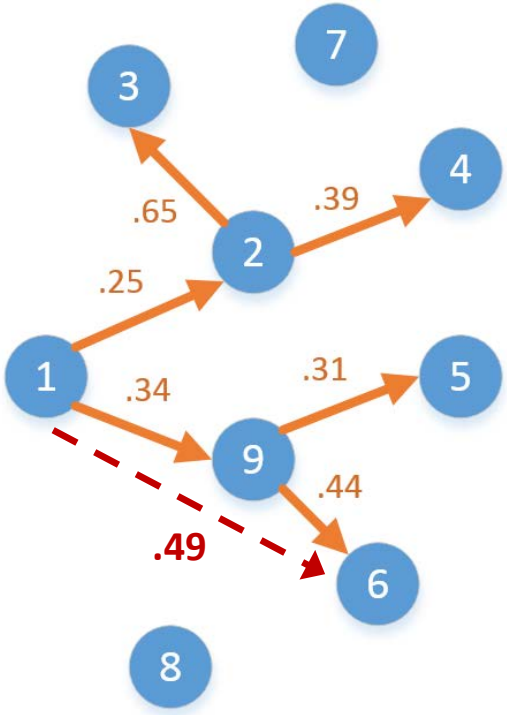
- Bounded similarity computation with pruning

```
1: function BOUNDEDSIM(d_q, d_c, ε)
2:     s ← 0
3:     for each j = 1, ..., m s.t. d_{c,j} > 0 do
4:         if d_{q,j} > 0 then
5:             s ← s + d_{q,j} × d_{c,j}
6:             if s + ||d_q^{>j}|| × ||d_c^{>j}|| < ε then
7:                 return -1
8: return s
```

$$\langle \mathbf{d}_q, \mathbf{d}_c \rangle = \underbrace{\langle \mathbf{d}_q^{\leq p}, \mathbf{d}_c^{\leq p} \rangle}_{\text{compute}} + \underbrace{\langle \mathbf{d}_q^{>p}, \mathbf{d}_c^{>p} \rangle}_{\text{estimate}}$$

Step 2: Approximate min-ϵ NN Graph

- For each object, find other min-ϵ neighbors
- Keep up to n neighbors in a max-heap
  - Initialize with k-NN neighbors & reverse-neighbors
  - Choose next neighbor's neighbor candidate in decreasing order of similarity
  - Choose at most μ candidates
  - Output all min-ϵ neighbors



Experimental evaluation: datasets

- RCV1: text of newswire stories
- WW500k, WW100k: EN Wikipedia documents
- Twitter: follow relationships on Twitter
- Wiki: page links among EN Wikipedia articles
- Orkut: friendship relationships on Orkut

Dataset	n	m	nnz
RCV1	804,414	43,001	61M
WW500k	494,244	343,622	197M
WW100k	100,528	339,944	79M
Twitter	146,170	143,469	200M
Wiki	1,815,914	1,648,879	44M
Orkut	3,072,626	3,072,441	223M

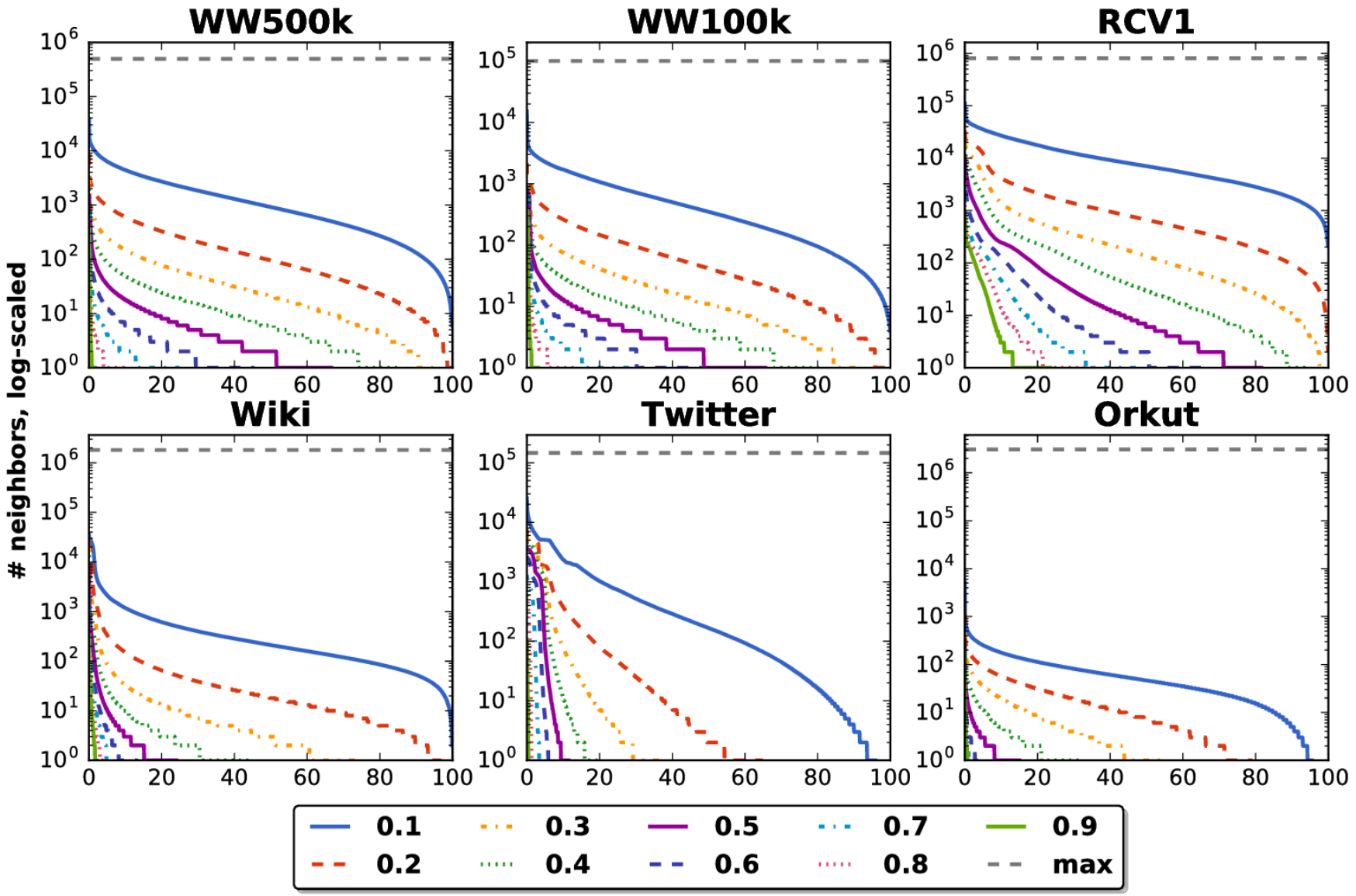
Neighborhood Graph Statistics

ϵ	μ	ρ	μ	ρ	μ	ρ
	WW500k		**RCV1**		**Orkut**	
0.1	1,749	3.5e-03	10,986	1.4e-02	76	2.5e-05
0.2	233	4.7e-04	2,011	2.5e-03	21	6.9e-06
0.3	64	1.3e-04	821	1.0e-03	7.2	2.4e-06
0.4	25	5.1e-05	355	4.4e-04	2.3	7.6e-07
0.5	10	2.2e-05	146	1.8e-04	0.69	2.3e-07
0.6	4.7	9.5e-06	57	7.2e-05	0.22	7.2e-08
0.7	2.1	4.2e-06	25	3.2e-05	0.09	3.1e-08
0.8	0.93	1.9e-06	14	1.8e-05	0.07	2.1e-08
0.9	0.28	5.7e-07	8.1	1.0e-05	0.06	2.0e-08

μ: Average neighborhood size

ρ: Output graph density

Neighborhood Graph Statistics



Experimental evaluation: results

ϵ	**cand**	**dps**	**cand**	**dps**	**cand**	**dps**
	WW100k		WW500k		RCV1	
0.3	0.2908	0.0380	0.1400	0.0152	0.4040	0.1058
0.4	0.1335	0.0176	0.0488	0.0060	0.2014	0.0521
0.5	0.0931	0.0094	0.0268	0.0022	0.1408	0.0271
0.6	0.0650	0.0045	0.0216	0.0010	0.1165	0.0134
0.7	0.1546	0.0057	0.0209	0.0004	0.0963	0.0058
0.8	0.3505	0.0042	0.0710	0.0002	0.1117	0.0040
0.9	0.3480	0.0012	0.1403	0.0001	0.0864	0.0019

The table shows the percent of potential object comparisons (***cand***) and computed dot-products (***dps***) executed by our method as opposed to those of a naive approach, when tuned to achieve 0.9 recall, for the test datasets and ϵ ranging from 0.3 to 0.9.

Experimental evaluation: results

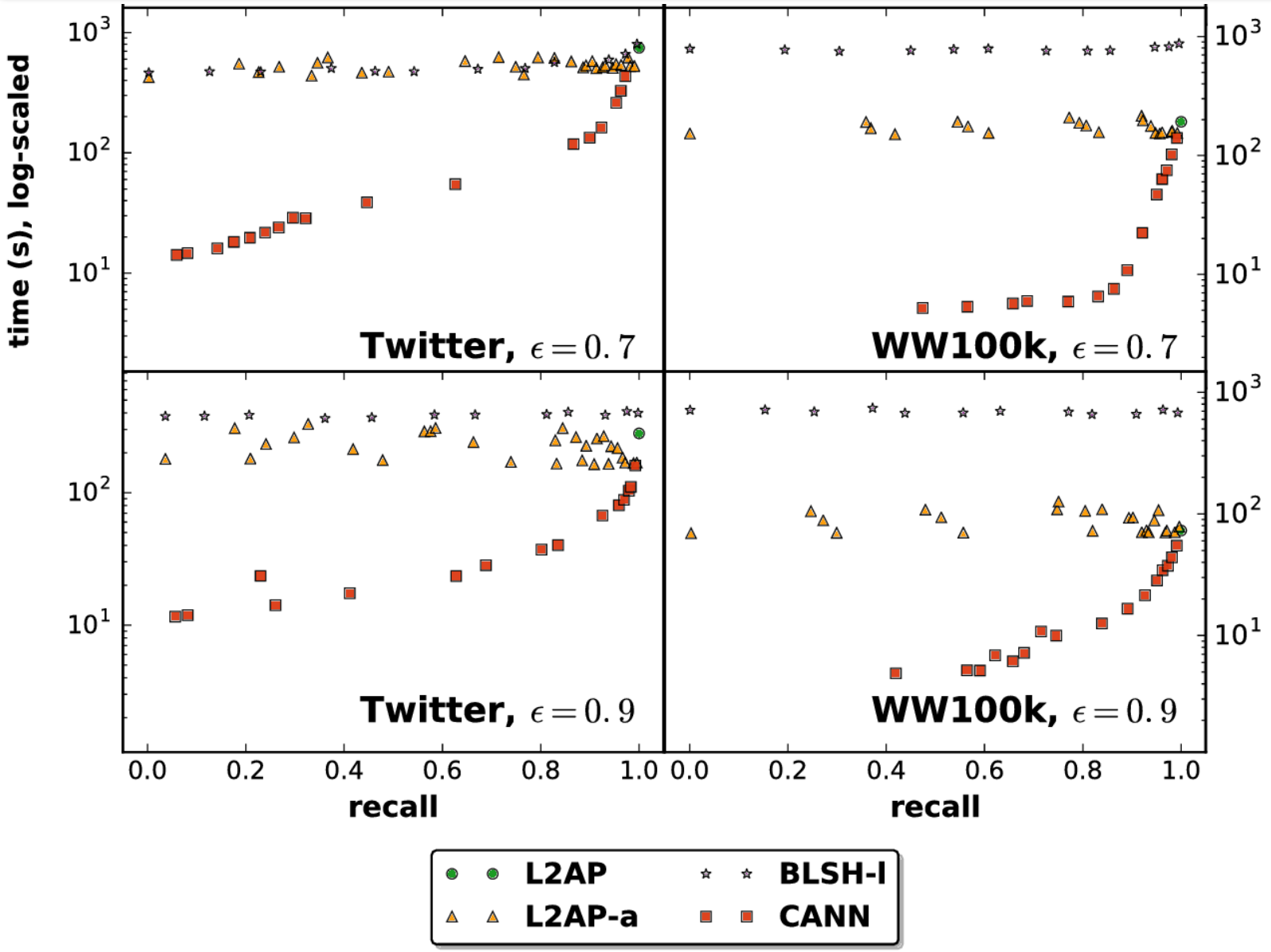
ϵ	cand	dps	cand	dps	cand	dps
	Twitter		Orkut		Wiki	
0.3	1.2240	0.2905	0.0063	0.0044	0.0194	0.0075
0.4	0.8944	0.1990	0.0045	0.0029	0.0100	0.0031
0.5	0.8007	0.1501	0.0029	0.0018	0.0087	0.0020
0.6	0.5810	0.0852	0.0018	0.0010	0.0055	0.0010
0.7	0.5374	0.0419	0.0009	0.0005	0.0042	0.0006
0.8	0.4131	0.0164	0.0003	0.0002	0.0025	0.0003
0.9	0.4736	0.0070	0.0003	0.0001	0.0020	0.0001

The table shows the percent of potential object comparisons (***cand***) and computed dot-products (***dps***) executed by our method as opposed to those of a naive approach, when tuned to achieve 0.9 recall, for the test datasets and ϵ ranging from 0.3 to 0.9.

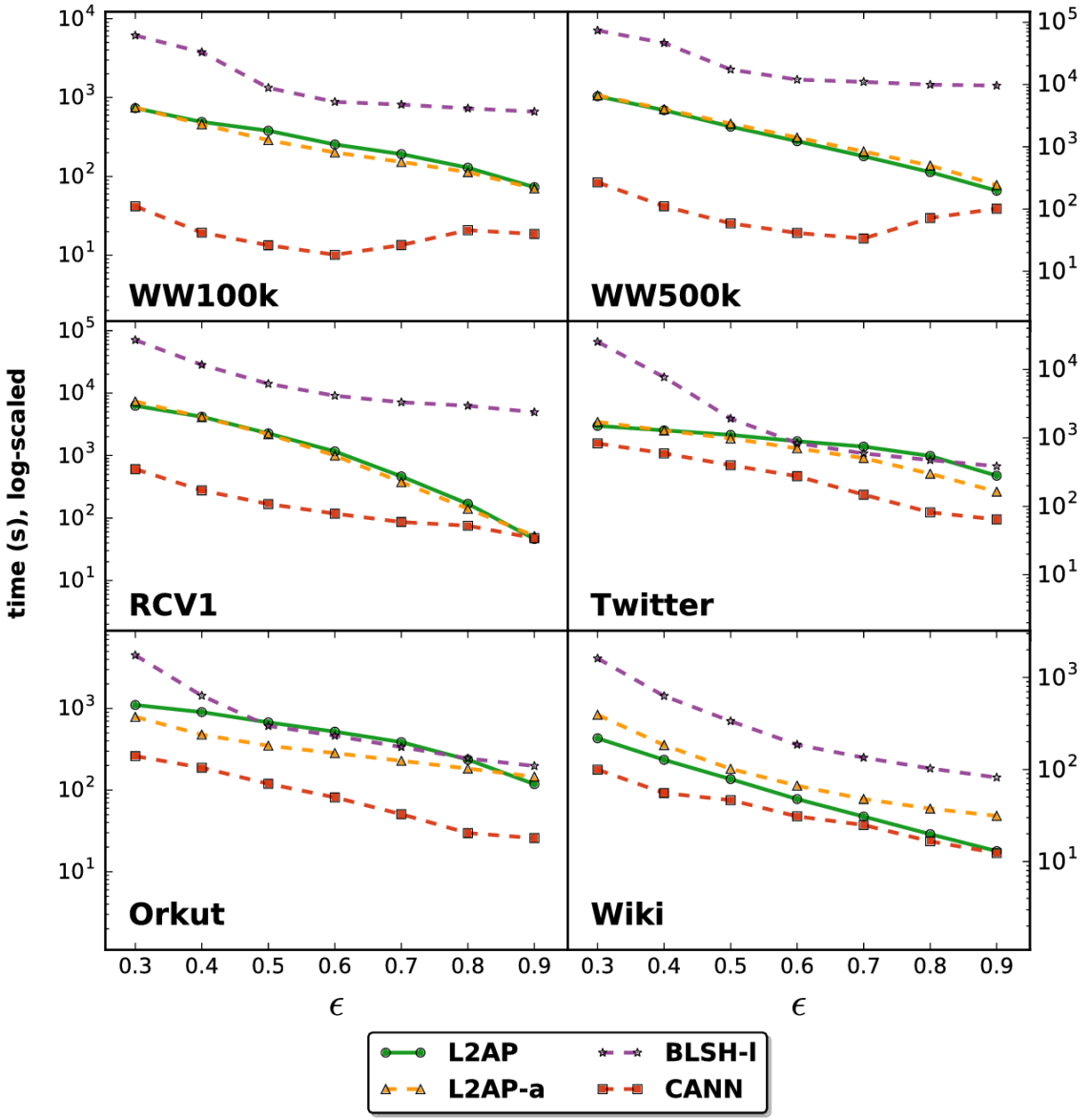
Experimental evaluation: methods

- L2AP (L2-Norm All-Pairs) [1]
  - *Exact* Cosine min-ϵ NN method
  - Uses several upper bound similarity estimates to prune the majority of false-positive candidates
- BayesLSH-Light (*BLSH-l*) [2]
  - Uses similar (weaker) candidate selection as L2AP
  - Filters candidates through Bayesian inference based on LSH bucket counts
- L2AP-Approx (*L2AP-a*) [1]
  - L2AP candidate selection and most filtering + Bayesian inference based filtering

Experimental evaluation: results



Experimental evaluation: results



Recall = 0.9

Questions?

References

[1] David C. Anastasiu and George Karypis. L2AP: Fast Cosine Similarity Search With Prefix L-2 Norm Bounds. Proceedings of the 30th IEEE International Conference on Data Engineering (ICDE 2014).

[2] V. Satuluri and S. Parthasarathy, “Bayesian locality sensitive hashing for fast similarity search,” Proc. VLDB Endow., vol. 5, no. 5, pp. 430–441, Jan. 2012.