Diamond Sampling for Approximate Maximum All-pairs Dot-product (MAD) Search

Grey Ballard, Tamara G. Kolda, Ali Pinar (Sandia National Labs)
and C. Seshadhri (UC Santa Cruz)

International Conference on Data Mining
Atlantic City, NJ
November 15, 2015

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Problem: Maximum All-pairs Dot-product (MAD) Search

Sandia National Laboratories

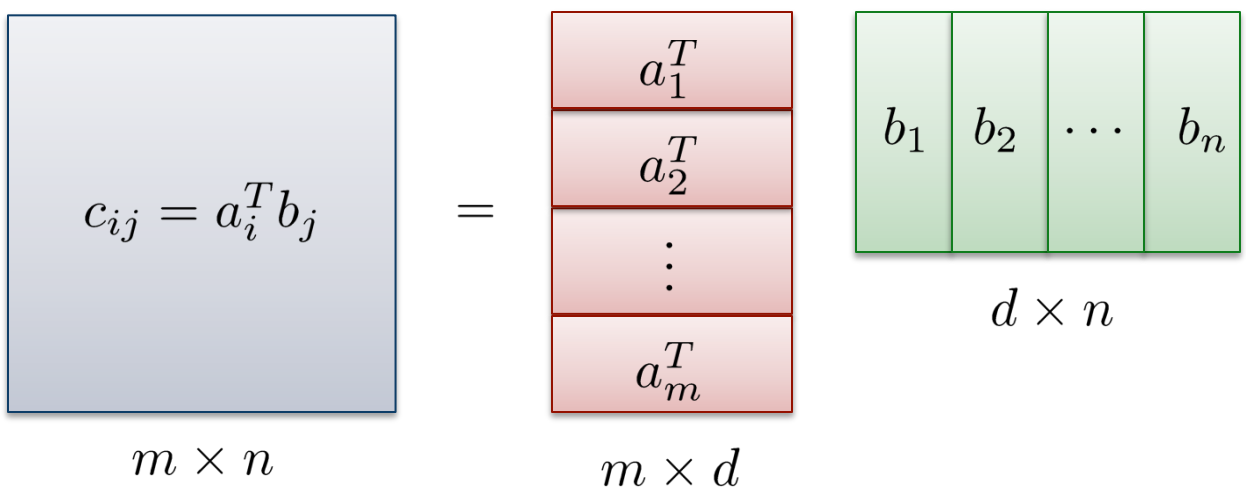
MAD Search

Given two sets of vectors in $\mathbb{R}^d$: $A = \{ a_1, \ldots, a_m \}$ and $B = \{ b_1, \ldots, b_n \}$

Find top-t pairs (i,j) that maximize: $| a_i \cdot b_j |$

Equivalent to finding the maximum-magnitude entries in the matrix product:

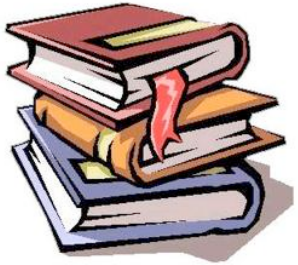
$$C = A^T B$$



Motivating Example: Free Samples!

Book seller wants to increase reviews on its site.

$$a_{ki} = \begin{cases} 1 & \text{if person } i \text{ reviewed book } k \\ 0 & \text{otherwise} \end{cases}$$



$$b_{kj} = \begin{cases} 1 & \text{if book } j \text{ similar to book } k \\ 0 & \text{otherwise} \end{cases}$$

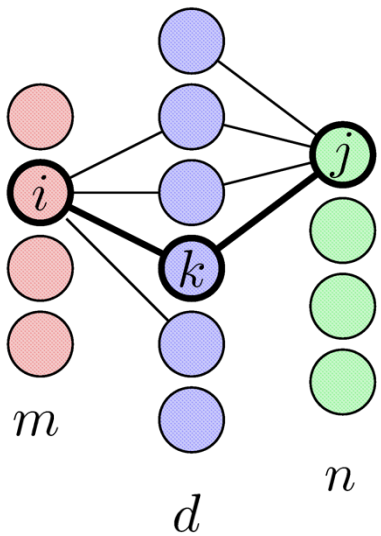
Send free books to users who have written reviews of similar books.

$$c_{ij} = \#\text{ books reviewed by person } i \text{ that are similar to book } j$$

MAD Search: Tripartite Graph Representation for Binary Inputs

Sandia National Laboratories

Tripartite Graph



Showing only edges corresponding to vector a_i in A and vector b_j in B.

Binary Inputs

$$A = [a_1 \cdots a_m] \text{ and } B = [b_1 \cdots b_n]$$

$$C = A^T B$$

c_{ij} = # wedges between nodes i and j:

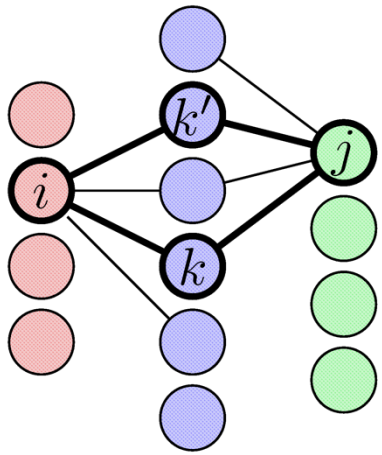
$$c_{ij} = a_i^T b_j = |\{ k \mid k \in \mathcal{N}_i^A \cap \mathcal{N}_j^B \}|$$

If there is a "wedge" (i,k,j), then $c_{ij} \geq 1$

$$\Pr(\text{ wedge } (i, *, j)) \propto c_{ij}$$

Wedge Sampling: Cohen & Lewis, 1997, 1999

Pair of Wedges = Diamond



Diamond = Two Intersecting Wedges

$c_{ij}{}^2$ = # diamonds between nodes i and j

Diamond: (k', i, k, j)

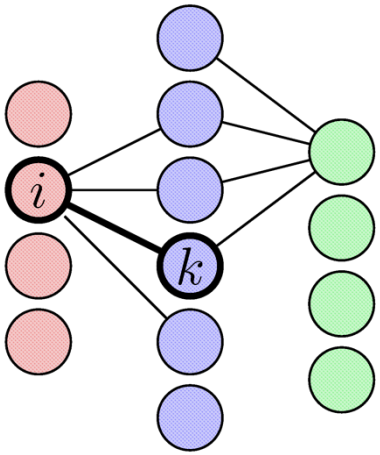
$$\Pr(\text{ diamond } (*, i, *, j)\,) \propto c_{ij}^2$$

<u>Key Idea:</u> Randomly select diamonds. The pairs (i,j) that are selected the most times will correspond to the highest entries of C with high probability.

Sampling Random Diamonds for General Inputs

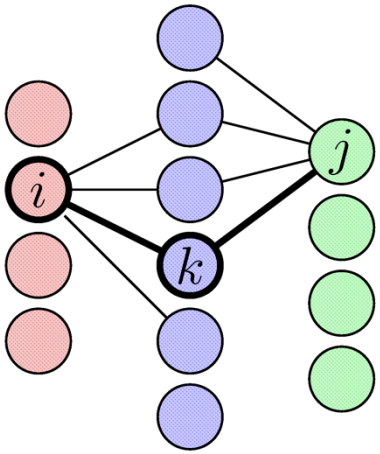
Enumerating all diamonds is too expensive.
Instead, we need an inexpensive way to sample.

Step 1: Choose edge $(i,k) \propto w_{ki}$



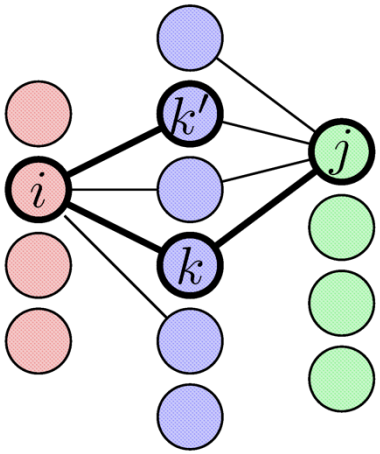
$$w_{ki} = |a_{ki}|\,\|a_{*i}\|_1 \|b_{k*}\|_1$$

Step 2: Choose neighbor of k



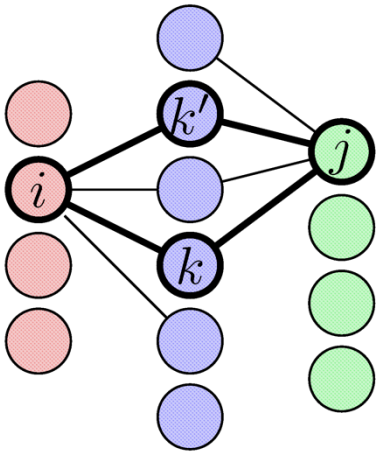
Sample j with probability $|b_{kj}|/\|b_{k*}\|_1$

Step 3: Choose neighbor of i



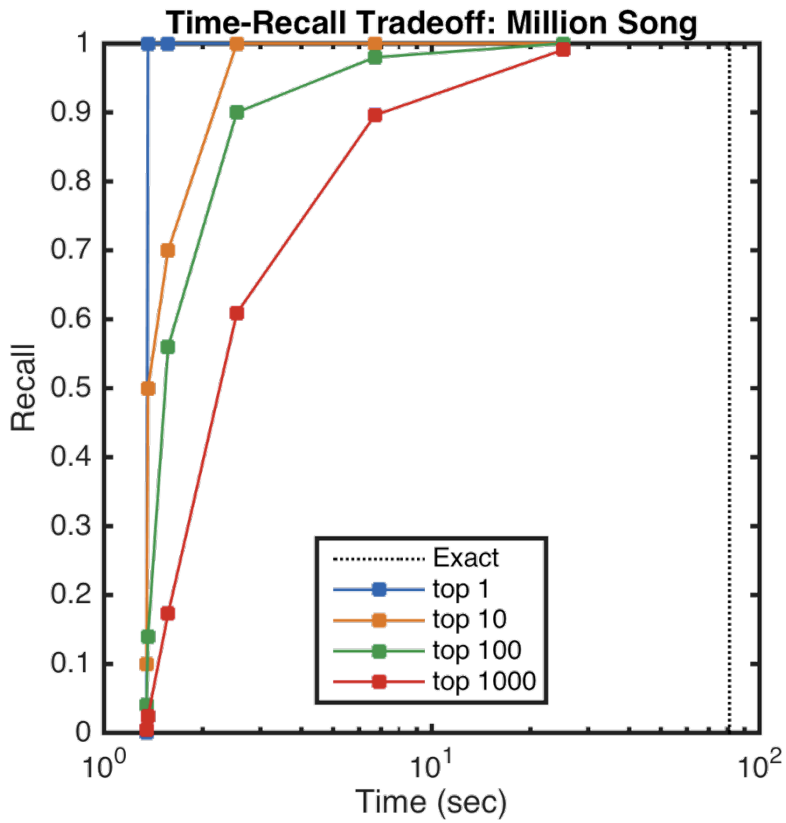
Sample k' with probability $|a_{k'i}|/\|a_{*i}\|_1$

Step 4: Check for edge (k',j) in B



$$x_{ij} \leftarrow x_{ij} + \operatorname{sgn}(a_{ki} b_{kj} a_{k'i})\, b_{k'j}$$

Quick Look: Up to 100X Speed-up Compared to Direct Computation



A = B = user-song listens
d ≈ 1M users, m = n ≈ 350K songs
nnz(A) ≈ 48M
C = A^TA = song similarity
max samples = 10^7

- Matrices
  - A = d x m, B = d x n
- Compute
  - C = A^TB = m x n matrix
  - Expensive in memory & time
- Memory workaround
  - Use a priority queue to hold the largest entries
- Use CSPARSE or BLAS
  - High-performance libraries

Expectation on Diamond Samples

Lemma: For diamond sampling, $\mathbb{E}[x_{ij}] = c_{ij}^2 \cdot s / \|W\|_1$

Picking entries proportional to c_{ij}^2 focuses on the largest-magnitude entries

X is output matrix from sampling
C is true output matrix
s is number of samples
$\|W\|_1$ is sum of weights of 3-paths

After sampling, we re-compute largest entries in *X* with exact dot products

Concentration Bounds for Diamond Sampling: How Many Samples?

Theorem: Fix some threshold τ and error probability $\delta \in (0,1)$. Assume all entries in A and B are nonnegative and at most K. Suppose $s \geq 12K\|W\|_1 \log(2mn/\delta)/\tau^2$. Then with probability at least $1-\delta$, the following holds for all indices i,j and i',j': if $c_{ij} > \tau$ and $c_{i'j'} < \tau/4$, then $x_{ij} > x_{i'j'}$.

For example, the number of samples needed to find the top entry is roughly

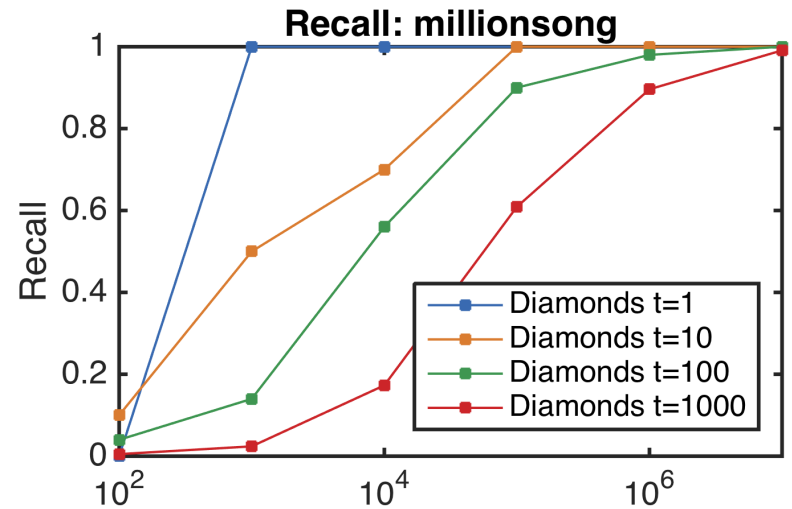
$$s = O(\|W\|_1 / \max c_{ij}^2)$$

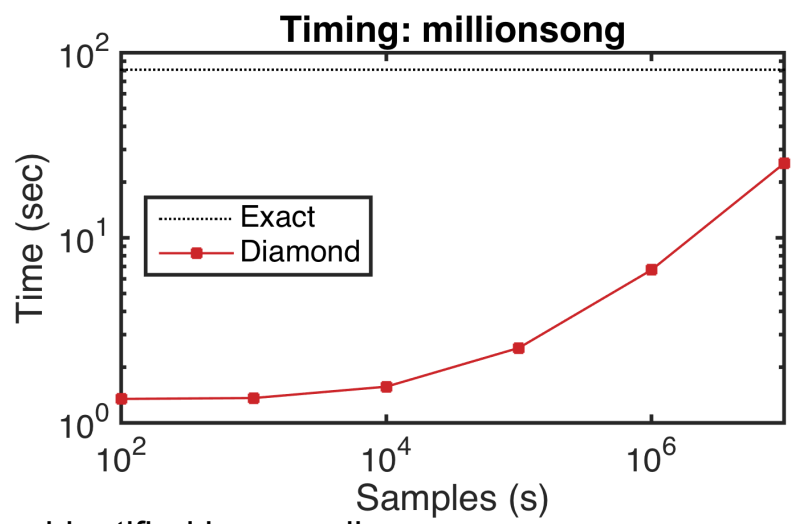
where $\|W\|_1$ is the sum of weights of all 3-paths

Results: Million Song

- Song plays by users
- A is sparse and $B = A$
  - $m \approx$ 400K songs
  - $n = d \approx$ 1M users
  - nnz(A) ≈ 48M
- Max entry in C: 6.1×10^6
- # Wedges = 1.4×10^{15}
- # Samples = $O(10^1)$
- Closure rate: 15%

- *Speed-up of 5-100X*
- *Top song pair: "Undo" by Bjork and "Revelry" by Kings of Leon*

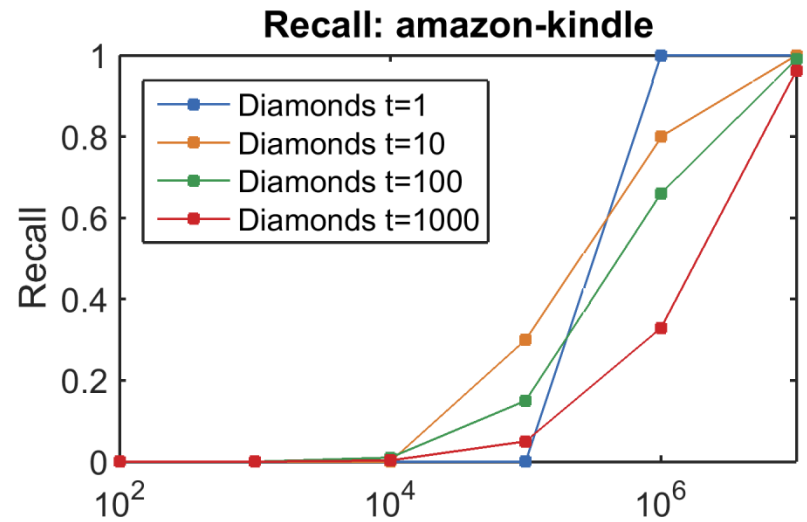


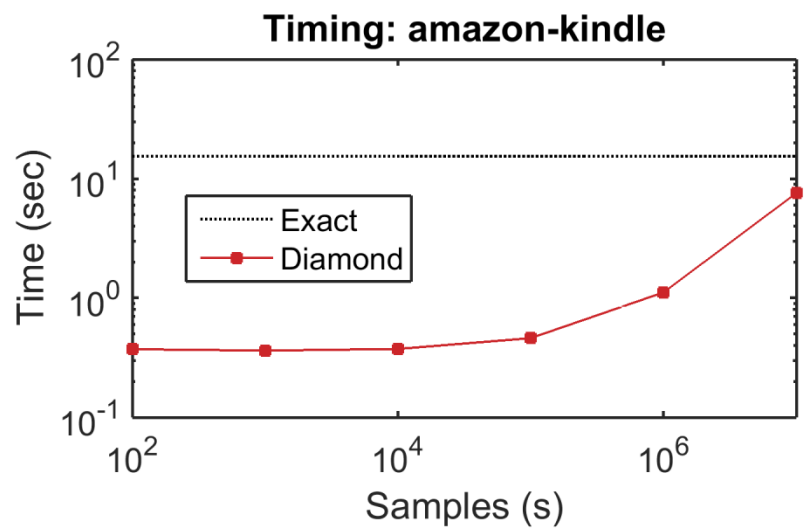


Recall is fraction of true top-t entries identified by sampling

Results: Amazon Kindle

- Product recommendations
- A is sparse reviewer-by-book rating matrix and B is sparse book similarity matrix
  - $m \approx$ 1.4M reviewers
  - $n = d \approx$ 430K books
  - nnz(A) ≈ 3.2M
  - nnz(B) ≈ 11M
- Max entry in C: 3.2×10^3
- # Wedges = 1.2×10^{12}
- # Samples = $O(10^{12})$
- Closure rate: 1.4%





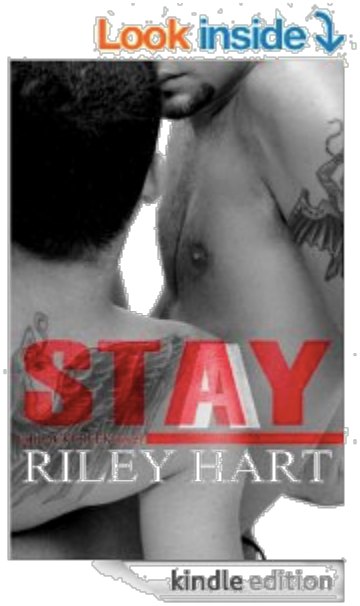
Recall is fraction of true top-t entries identified by sampling

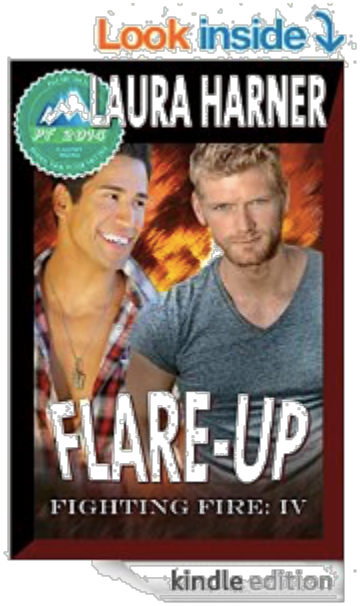
Example Reviewer-Book Recommendation

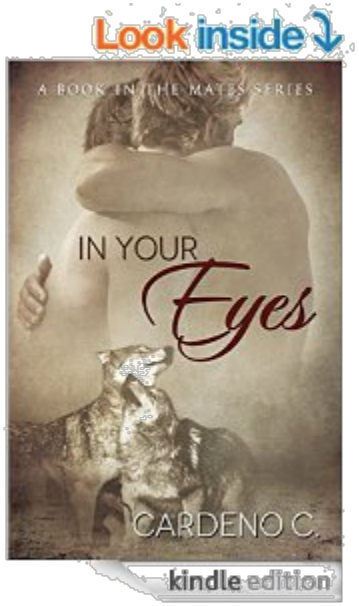
Sandia National Laboratories

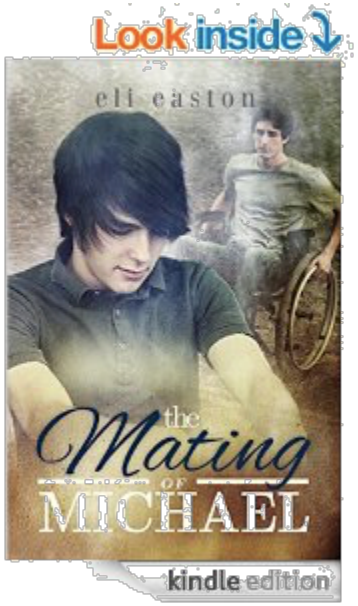
Recommended Book

Recommended User ID = 1317513
Previous Reviews Include…

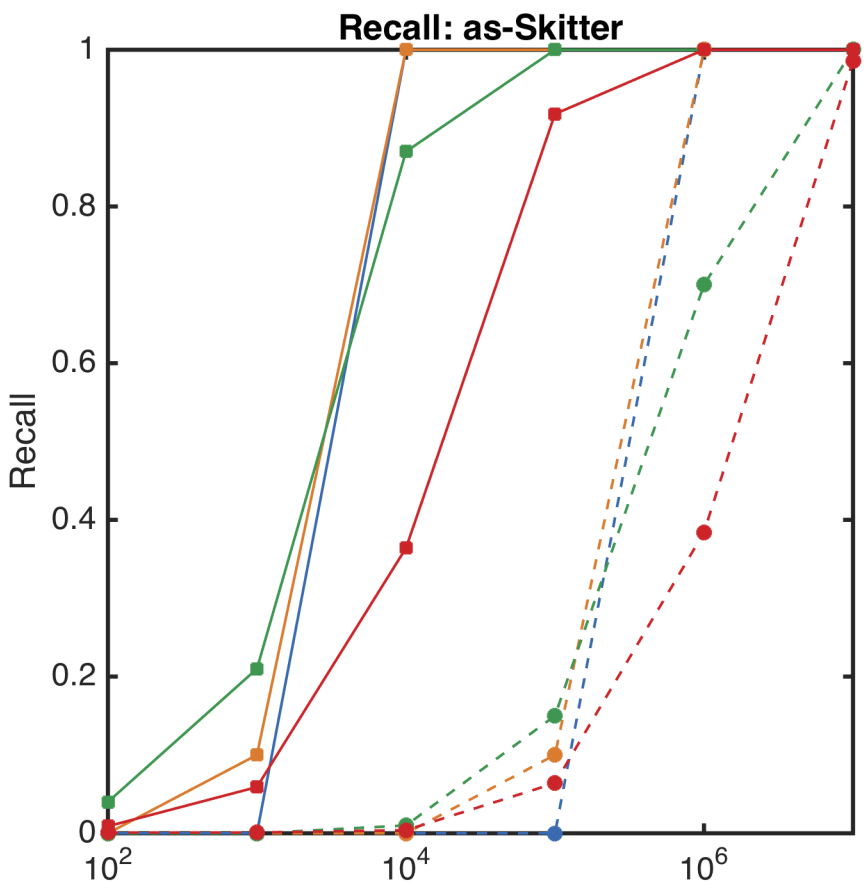


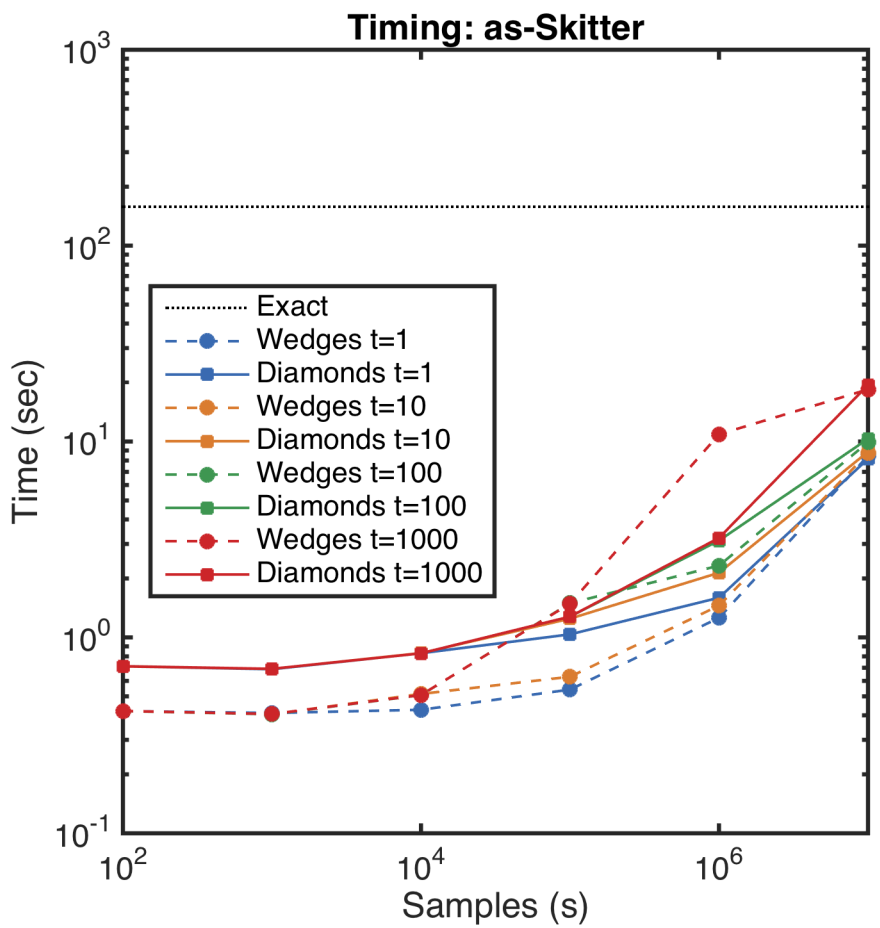






Comparison to Wedge Sampling

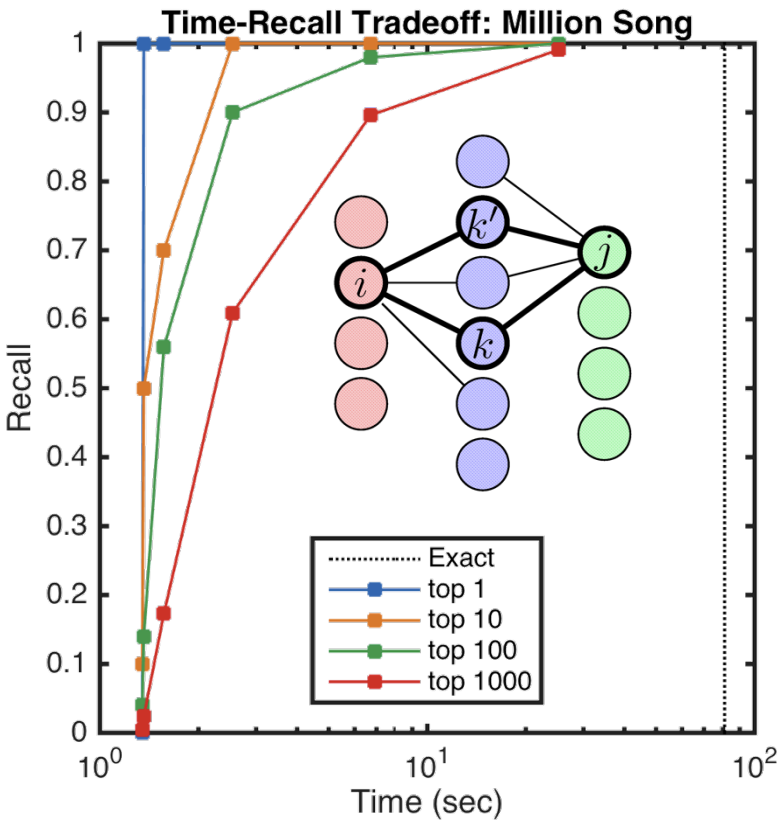




Wedge Sampling: Cohen & Lewis, 1997, 1999

Summary of Contributions

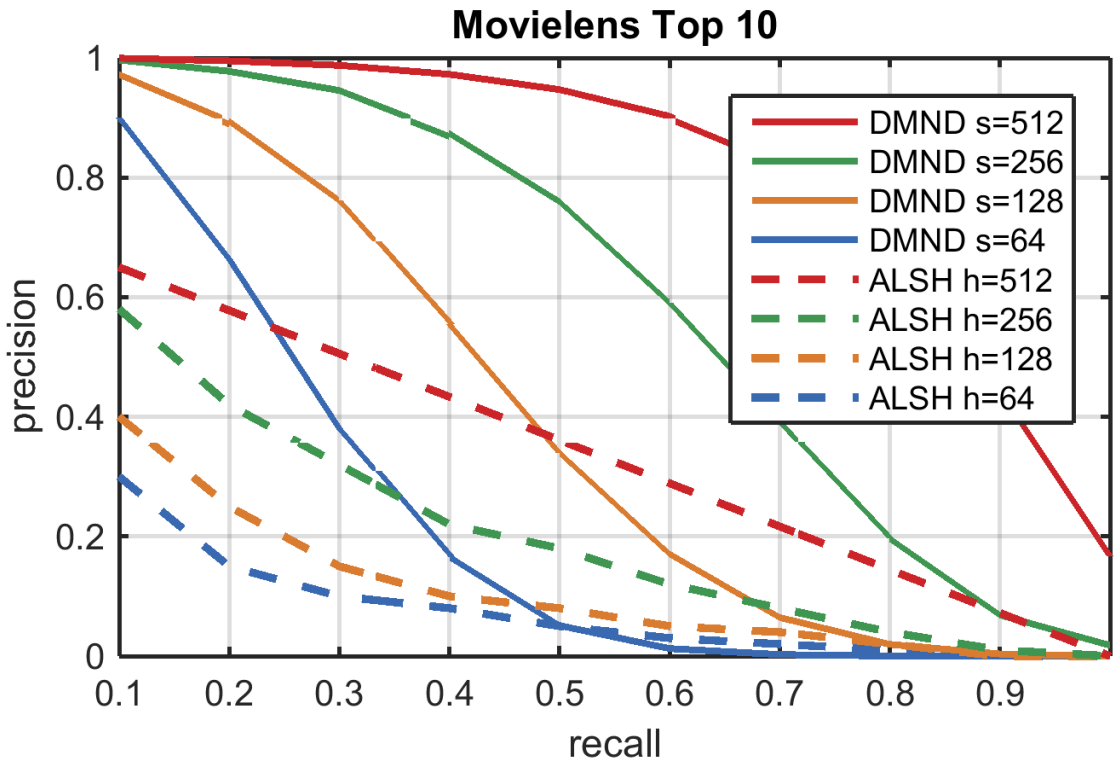
- Contributions
  - Diamond Sampling: new randomized approach for MAD search
    - Key step is to sample 3-paths
    - Selecting (i,j) proportional to $(a_i \cdot b_j)^2$
    - Works for binary, nonnegative, and real-valued
  - Theoretical analysis
    - Expectation shows how sampling is proportion to squared dot product
    - Concentration bounds indicate the number of samples required (nonnegative only)
  - Experimental results
    - Faster than direct computation
    - Fewer samples than wedge sampling
    - Compares well against recent LSH results



For more information:
Grey Ballard
gmballa@sandia.gov

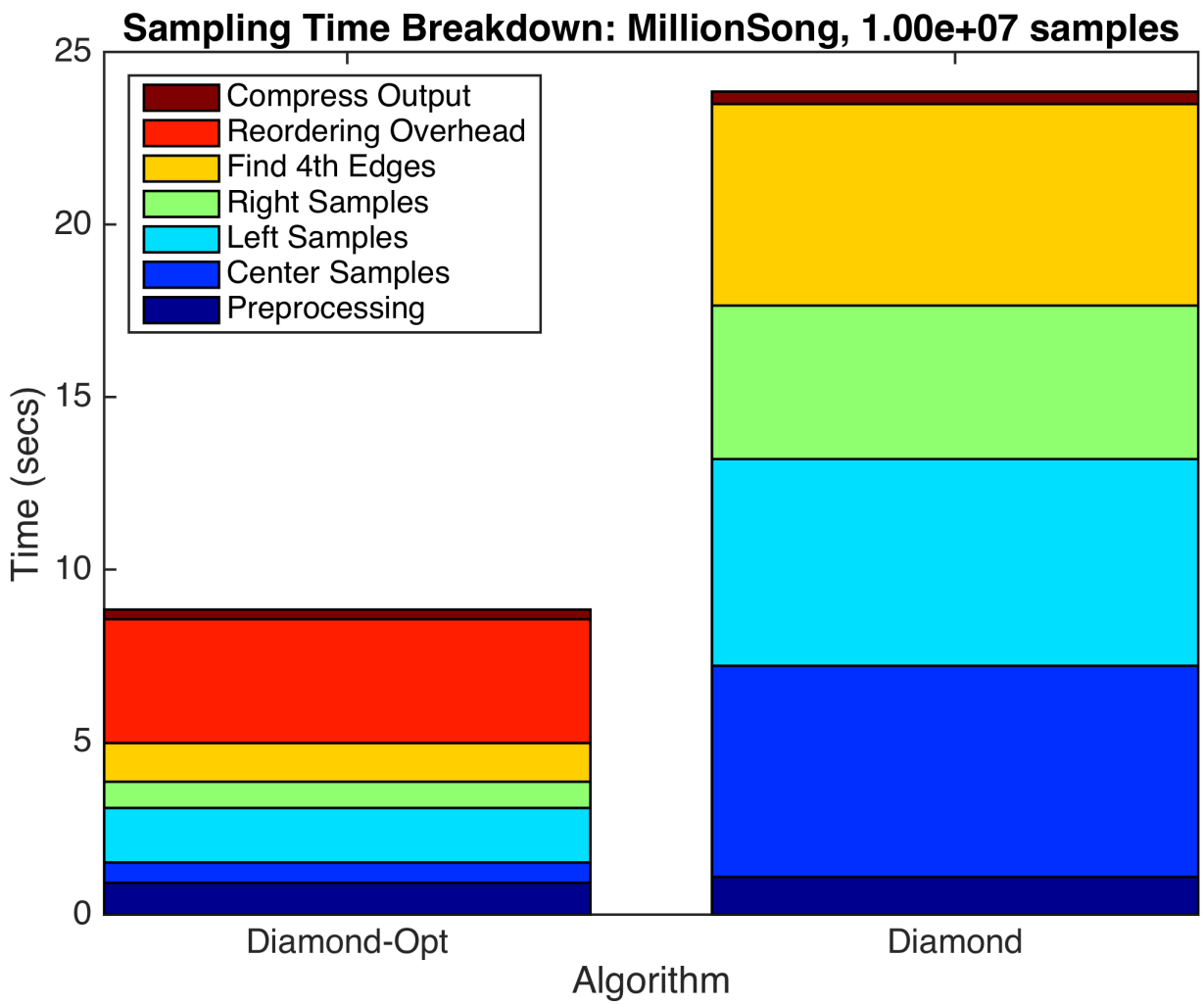
Comparison to Locality-Sensitive Hashing

Comparison of precision-recall for LSH versus Diamond Sampling



Asymmetric LSH: Shrivastava and Li, NIPS 2014

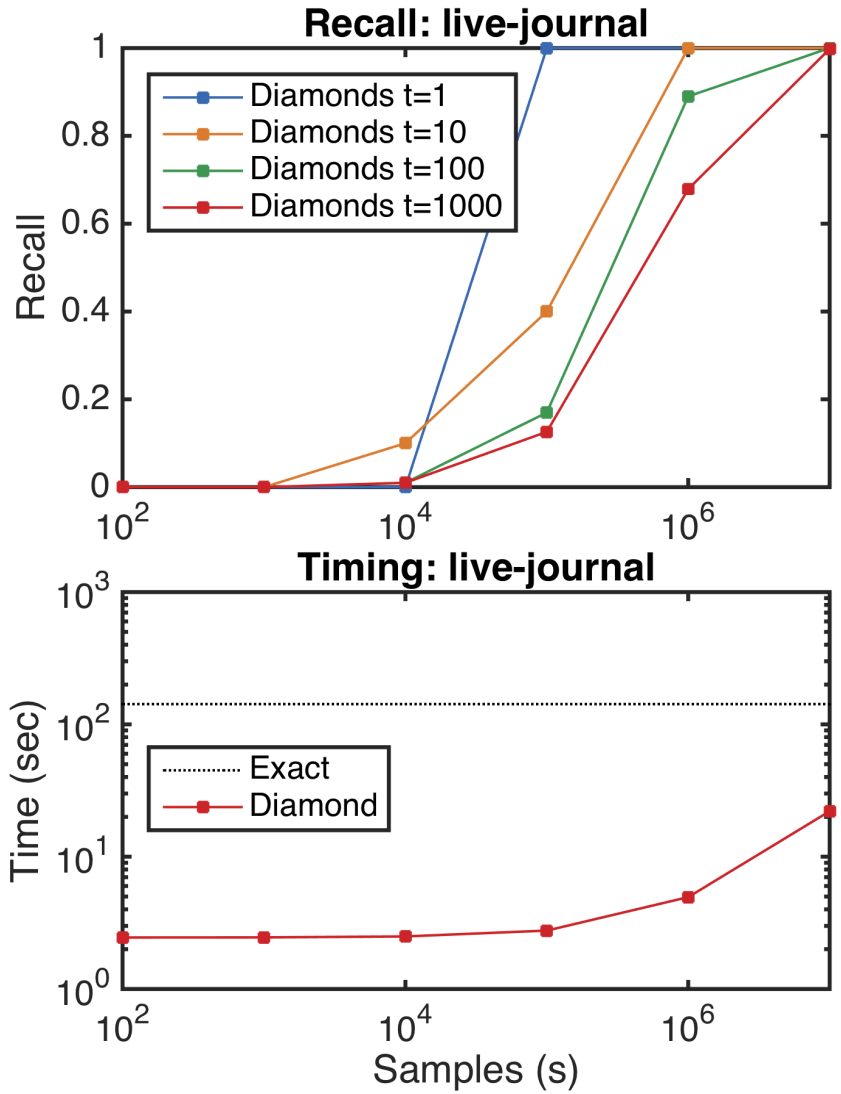
Cache locality optimizations



Results: Live Journal

- Online community
- A is sparse, symmetric, binary and $B = A$
  - $m = n = d$ = 4,847,571
  - nnz(A) = 68,993,773
- Max entry in C: 3.0×10^3
- # Wedges = 1.5×10^{12}
- # Samples = $O(10^5)$
- Closure rate: 13%

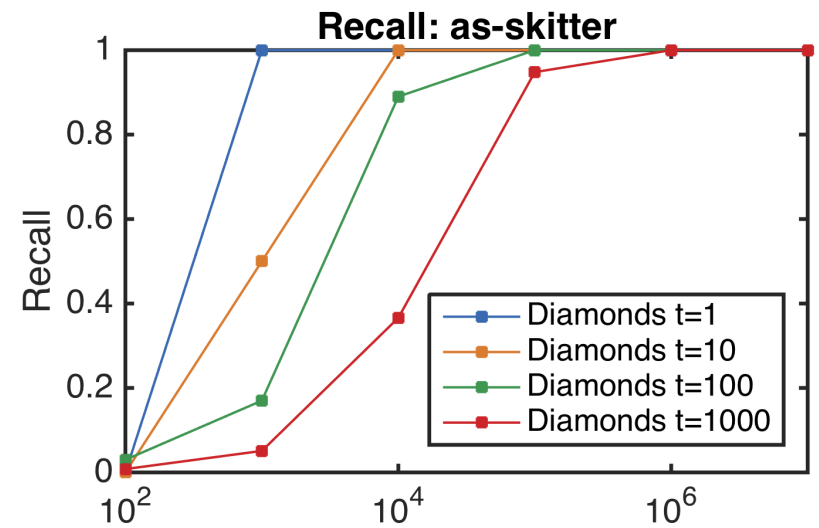
- *10-100X Speed-up*
- *Need 10000 samples for top entry, as predicted*

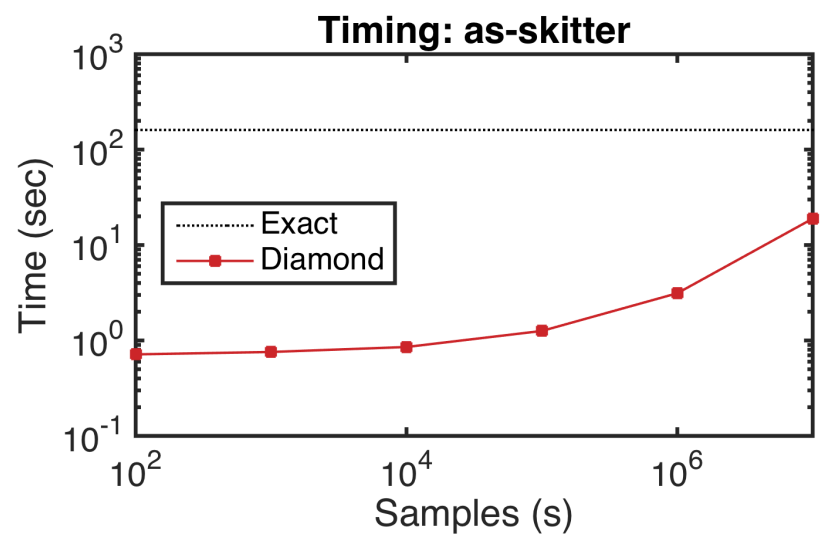


Results: as-Skitter

- Internet topology graph
- A is sparse, symmetric, binary and $B = A$
  - $m = n = d$ = 1,696,415
  - nnz(A) = 11,095,298
- Max entry in C: 3.0×10^5
- # Wedges = 2.9×10^{12}
- # Samples = $O(10^3)$
- Closure rate: 19%

- *10-100X Speed-up*
- *Need only 1000 samples for top entry, as predicted*

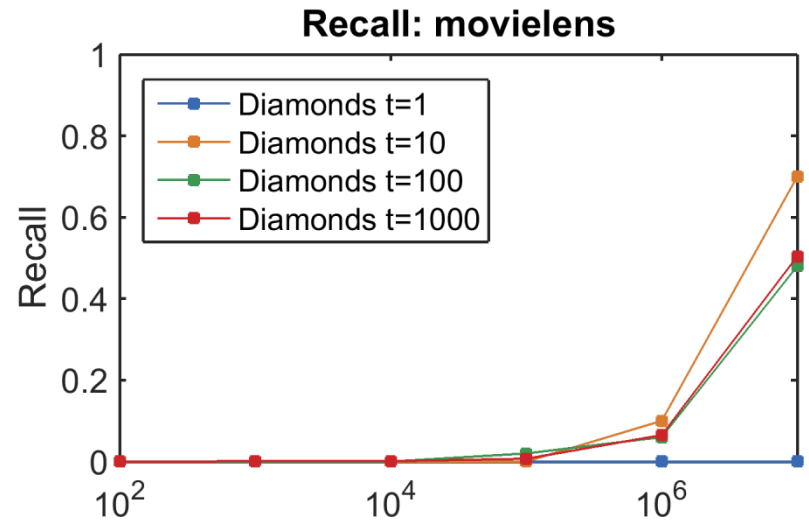


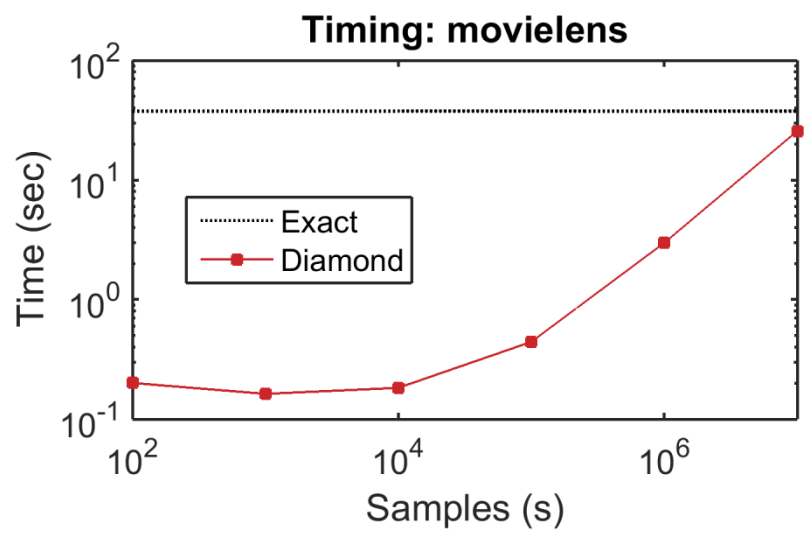


Results: Movielens

- Movielens-10M dataset
- Compute low-rank SVD/PCA of movie-rating matrix R
  - $R \approx A^T B$
- A and B are dense
  - m = 65,133 movies
  - n = 71,567 users
  - d = 150 components
- Max entry in C: 1.1×10^1
- # Wedges = 2.8×10^{10}
- # Samples = $O(10^8)$
- Closure rate: 100%

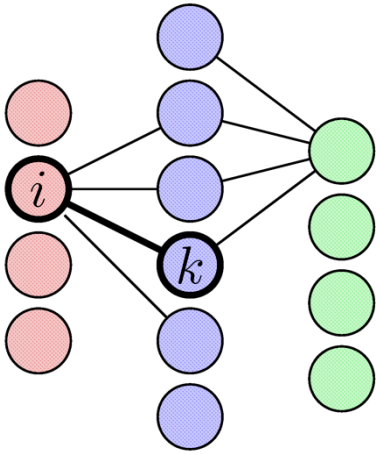
- *Difficult to find top entries due to relatively small range*





Sampling Random Diamonds for Binary Inputs

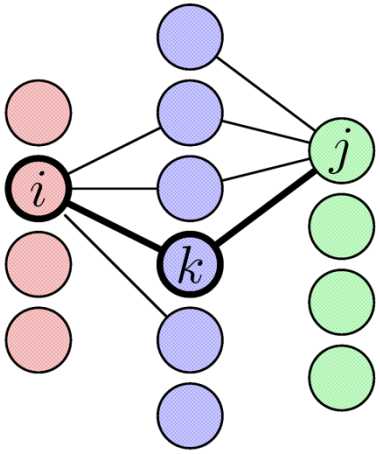
Step 1: Choose edge $(i,k) \propto w_{ik}$



Number of three-paths centered at edge (i,k):

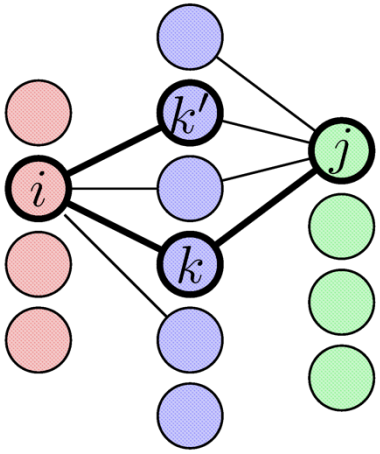
$$w_{ki} = \deg_i^A \deg_k^B$$

Step 2: Choose neighbor of k



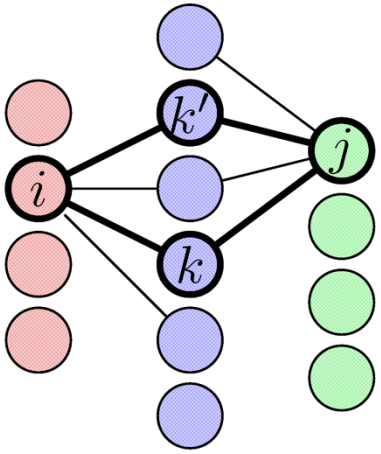
Sample j from $\mathcal{N}_k^B$

Step 3: Choose neighbor of i



Sample k' from $\mathcal{N}_i^A$

Step 4: Check for edge (k', j) in B



If edges exists, diamond complete:

$$x_{ij} \leftarrow x_{ij} + 1$$

Three-path sampling: Jha et al. WWW 2015

Chance of Picking Three-Path

Recall: $w_{ki} = |a_{ki}|\,\|a_{*i}\|_1 \|b_{k*}\|_1$

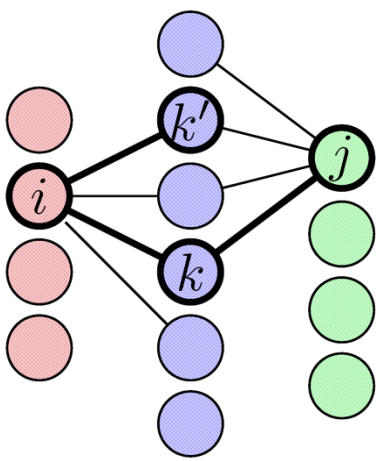
Lemma: $\Pr(\mathcal{E}_{k'ikj}) = |a_{ki} b_{kj} a_{k'i}| / \|W\|_1$ where $\|W\|_1 = \sum_{ki} |w_{ki}|$

Proof: $\Pr(\mathcal{E}_{k'ijk}) = \Pr(\text{ctr } (i,k)) \cdot \Pr(\text{endpts } j \text{ and } k' | \text{ctr } (i,k))$

$$= \frac{w_{ki}}{\|W\|_1} \cdot \frac{|b_{kj}|}{\|b_{k*}\|_1} \cdot \frac{|a_{k'i}|}{\|a_{*i}\|_1}$$

$$= \frac{|a_{ki}|\,\|a_{*i}\|_1 \|b_{k*}\|_1}{\|W\|_1} \cdot \frac{|b_{kj}|}{\|b_{k*}\|_1} \cdot \frac{|a_{k'i}|}{\|a_{*i}\|_1}$$

$$= \frac{|a_{ki} b_{kj} a_{k'i}|}{\|W\|_1}.$$



Expectation on Diamond Samples

Samples: $x_{ij} \leftarrow x_{ij} + \mathrm{sgn}(a_{ki} b_{kj} a_{k'i})\, b_{k'j}$

Lemma: For diamond sampling, $\mathbb{E}[x_{ij}] = c_{ij}^2 \cdot s / \|W\|_1$

Proof (for $s = 1$):

$$\begin{aligned}
\mathbb{E}[x_{ij}] &= \sum_k \sum_{k'} \Pr(\mathcal{E}_{k'ikj}) \cdot \mathrm{sgn}(a_{ki} b_{ki} a_{k'j})\, b_{k'j} \\
&= \sum_k \sum_{k'} \frac{|a_{ki} b_{kj} a_{k'i}|}{\|W\|_1} \cdot \mathrm{sgn}(a_{ki} b_{kj} a_{k'i})\, b_{k'j} \\
&= \frac{1}{\|W\|_1} \sum_k \sum_{k'} a_{ki} b_{kj} a_{k'i} b_{k'j} \\
&= \frac{1}{\|W\|_1} \Big(\sum_k a_{ki} b_{kj}\Big)\Big(\sum_{k'} a_{k'i} b_{k'j}\Big) \\
&= \frac{1}{\|W\|_1} \Big(\sum_k a_{ki} b_{kj}\Big)^2 = \frac{c_{ij}^2}{\|W\|_1}.
\end{aligned}$$

Picking entries proportional to c_{ij}^2 focuses on the largest-magnitude entries