Compressive spectral embedding: sidestepping the SVD

Dinesh Ramasamy, Upamanyu Madhow
University of California, Santa Barbara

**FASTrEmbed:** Bypassing the SVD

**Implementation considerations**

**Results**

**Community detection**

- DBLP collaboration network
  - 31,078 authors, 1 million edges
  - ARPACK: Smallest 500 eigenvectors
  - 105 minutes
- Compressive embedding
  - 80-dimensional embedding of 500 eigenvectors
  - 1 minute (12 cores)

**Effect of cascading**

Pairwise cosine similarity (DBLP)

**Pairwise distances preserved**

- Community detection on Amazon co-purchasing network
- Payoff saturates with \( j \)
- Effect of cascading: Pairwise cosine similarity (DBLP)

**Summary**

- Succinctly captures pairwise \( i \), similarity metrics
- Dramatic dimensionality reduction (\( i \ll K \))
- Use more singular vectors for inference
- Speed
  - Parallel implementation
  - ~100 times faster than partial SVD
- Bottom-line: Better, faster inference

**Bottlenecks of scale**

- More data (Large \( n \)) \(
  \rightarrow\)
  Rich structure \(
  \rightarrow\)
  Many factors (Large \( K \))
- Partial SVD complexity \( O(n^2k) \)

**Spectral methods: Workflow**

- Singular Value Decomposition step prior for inference
- Recommender systems, clustering, community detection on networks, graph mining, ...

**Domain dependent TRANSFORM(input)**

- SVD (transformed input)
- \( M = N \times M \) \( \rightarrow\) factor

**Transformation**

- Adjacency matrix of a graph
- Inference (low-rank)

**FastEmbed**

**Reduce dimensionality**

- Kernel projections
- Preserves "similarity" via random projections

**Real world matrices are sparse**

- Matrix-vector products cheap
- Matrix-products can compute polynomial, etc.

**Polynomial approximation of \( f(A) \)**

\[ f(A) = \sum_{k=0}^{K-1} \alpha_k \phi_k(A) \]

**Dimensionality reduction**

\[ f(A) = \sum_{k=0}^{K-1} \phi_k(A) \]

**Polynomial approximation**

\[ f(A) = \sum_{k=0}^{K-1} \alpha_k \phi_k(A) \]

**FastEmbed**

**Construct matrix \( M \)

**Embedding of rows \( \phi(A) \)**

**Real world matrices are sparse**

- Matrix-vector products cheap
- Matrix-products can compute polynomial, etc.

**FastEmbed**

**Construct matrix \( M \)

**Embedding of rows \( \phi(A) \)**

**Real world matrices are sparse**

- Matrix-vector products cheap
- Matrix-products can compute polynomial, etc.

**FastEmbed**

**Construct matrix \( M \)

**Embedding of rows \( \phi(A) \)**

**Real world matrices are sparse**

- Matrix-vector products cheap
- Matrix-products can compute polynomial, etc.

**FastEmbed**

**Construct matrix \( M \)

**Embedding of rows \( \phi(A) \)**

**Real world matrices are sparse**

- Matrix-vector products cheap
- Matrix-products can compute polynomial, etc.