Representation Power and Theoretical Foundations of Modern Node Embeddings

Cameron Musco.
University of Massachusetts Amherst
SIAM Mathematics of Data Science 2022

Our Driving Questions

Are there inherent limitations on the power of low-dimensional node embeddings in representing real world graphs?

- *An Interpretable Graph Generative Model with Heterophily*. Sudhanshu Chanpuriya, Ryan Rossi, Anup Rao, Tung Mai, Nedim Lipka, Zhao Song, and Cameron Musco. ArXiv.
- *On the Power of Edge Independent Graph Models*. Sudhanshu Chanpuriya, Cameron Musco, Konstantinos Sotiropoulos, and Charalampos Tsourakakis. NeurIPS 2021.
- *Node Embeddings and Exact Low-Rank Representations of Complex Networks*. Sudhanshu Chanpuriya, Cameron Musco, Konstantinos Sotiropoulos, and Charalampos Tsourakakis. NeurIPS 2020.

Why exactly do 'modern' node embeddings outperform classic methods, like spectral embeddings?

- *InfiniteWalk: Deep Network Embeddings as Laplacian Embeddings with a Nonlinearity*. Sudhanshu Chanpuriya and Cameron Musco. KDD 2020.

Our Driving Questions

Are there inherent limitations on the power of low-dimensional node embeddings in representing real world graphs?

- *An Interpretable Graph Generative Model with Heterophily*. Sudhanshu Chanpuriya, Ryan Rossi, Anup Rao, Tung Mai, Nedim Lipka, Zhao Song, and Cameron Musco. ArXiv.
- *On the Power of Edge Independent Graph Models*. Sudhanshu Chanpuriya, Cameron Musco, Konstantinos Sotiropoulos, and Charalampos Tsourakakis. NeurIPS 2021.
- *Node Embeddings and Exact Low-Rank Representations of Complex Networks*. Sudhanshu Chanpuriya, Cameron Musco, Konstantinos Sotiropoulos, and Charalampos Tsourakakis. NeurIPS 2020.

Why exactly do 'modern' node embeddings outperform classic methods, like spectral embeddings?

- *InfiniteWalk: Deep Network Embeddings as Laplacian Embeddings with a Nonlinearity*. Sudhanshu Chanpuriya and Cameron Musco. KDD 2020.

Our Driving Questions

Are there inherent limitations on the power of low-dimensional node embeddings in representing real world graphs?

- *An Interpretable Graph Generative Model with Heterophily*. Sudhanshu Chanpuriya, Ryan Rossi, Anup Rao, Tung Mai, Nedim Lipka, Zhao Song, and Cameron Musco. ArXiv.
- *On the Power of Edge Independent Graph Models*. Sudhanshu Chanpuriya, Cameron Musco, Konstantinos Sotiropoulos, and Charalampos Tsourakakis. NeurIPS 2021.
- *Node Embeddings and Exact Low-Rank Representations of Complex Networks*. Sudhanshu Chanpuriya, Cameron Musco, Konstantinos Sotiropoulos, and Charalampos Tsourakakis. NeurIPS 2020.

Why exactly do 'modern' node embeddings outperform classic methods, like spectral embeddings?

- *InfiniteWalk: Deep Network Embeddings as Laplacian Embeddings with a Nonlinearity*. Sudhanshu Chanpuriya and Cameron Musco. KDD 2020.

Summary of our Results

- **Starting Point:** Seshadhri, Sharma, Stolman, and Goel (PNAS 2020) argue that low-dimensional node embeddings cannot represent sparse, triangle-dense graphs. E.g., most social networks.

Summary of our Results

- **Starting Point:** Seshadhri, Sharma, Stolman, and Goel (PNAS 2020) argue that low-dimensional node embeddings cannot represent sparse, triangle-dense graphs. E.g., most social networks.
- **Positive Result:** We show that in fact, low-dimensional node embeddings can exactly represent any bounded-degree or bounded arboricity graph, including triangle-dense graphs. *As long as each node is assigned two embeddings* (as in DeepWalk, node2vec, LINE, CELL, etc.)
- **Negative Result:** We show that, regardless of dimension, edge-independent graph generative models (including NetGAN, variational graph autoencoders, CELL, Graphite, etc.) cannot generate triangle-dense graphs, unless they essentially memorize a single graph.

Embedding Model

Consider a simple model for generating a random unweighted, undirected graph $G = (V, E)$ with n nodes.

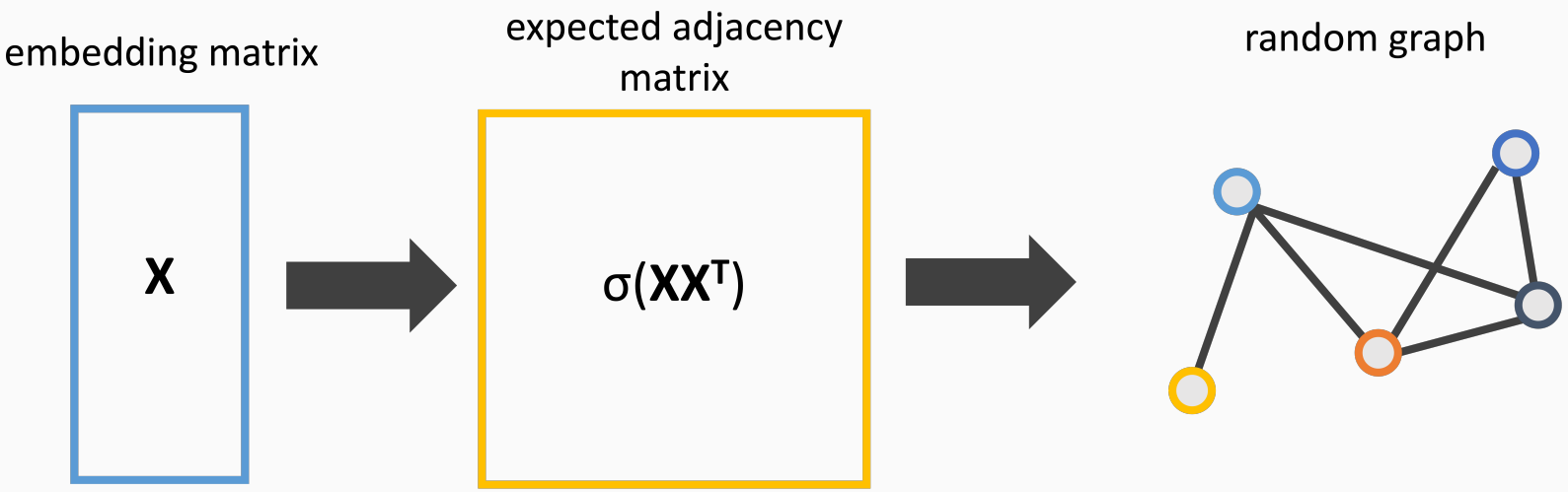
- For each node, we compute an embedding $x_i \in \mathbb{R}^k$.
- We let $p_{ij} = \sigma(\langle x_i, x_j \rangle)$ be the probability of edge (i, j) appearing in the graph. $\sigma : \mathbb{R} \to [0, 1]$ is a non-linearity that outputs probabilities. E.g., $\sigma(z) = \frac{1}{1+e^{-z}}$ or $\sigma(z) = \max(0, \min(z, 1))$.
- We generate G by including each edge independently with probability p_{ij}.
- Letting $\mathbf{X} \in \mathbb{R}^{n \times k}$ have rows equal to the embeddings, the expected adjacency matrix of G is given by $\mathbb{E}[\mathbf{A}] = \sigma(\mathbf{X}\mathbf{X}^T)$.

Embedding Model

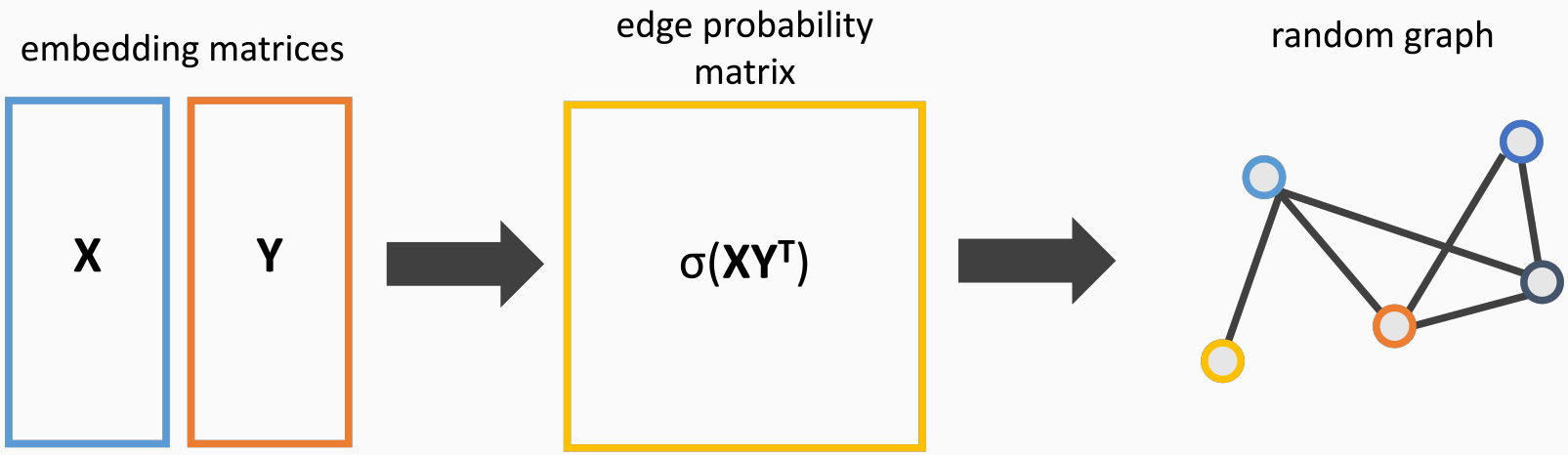
Consider a simple model for generating a random unweighted, undirected graph $G = (V, E)$ with n nodes.

- For each node, we compute an embedding $x_i \in \mathbb{R}^k$.
- We let $p_{ij} = \sigma(\langle x_i, x_j \rangle)$ be the probability of edge (i, j) appearing in the graph. $\sigma : \mathbb{R} \to [0, 1]$ is a non-linearity that outputs probabilities. E.g., $\sigma(z) = \frac{1}{1+e^{-z}}$ or $\sigma(z) = \max(0, \min(z, 1))$.
- We generate G by including each edge independently with probability p_{ij}.
- Letting $\mathbf{X} \in \mathbb{R}^{n \times k}$ have rows equal to the embeddings, the expected adjacency matrix of G is given by $\mathbb{E}[\mathbf{A}] = \sigma(\mathbf{X}\mathbf{X}^T)$.
- Will also consider a related model where each node has two embeddings x_i, y_i and $p_{ij} = \sigma(\langle x_i, y_j \rangle)$. E.g., DeepWalk, LINE, CELL.

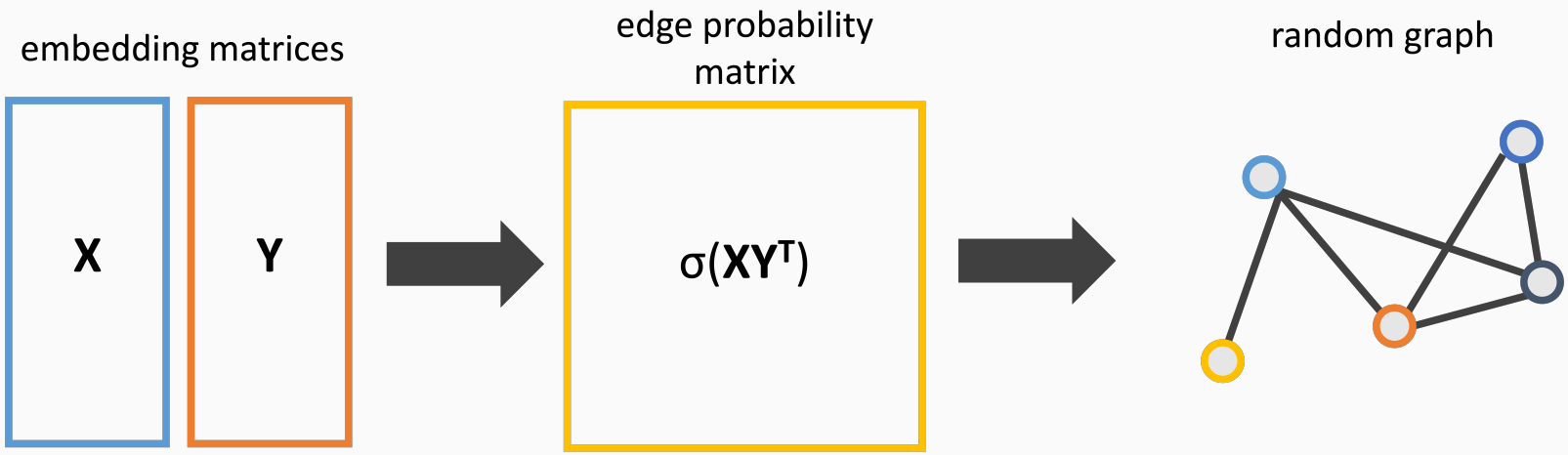
Embedding Model



Embedding Model



Embedding Model



- The embeddings are often trained using an input graph with adjacency matrix **A**, with the goal of having $\mathbf{A} \approx \sigma(\mathbf{X}\mathbf{X}^T)$. I.e., a sort of low-rank approximation of the adjacency matrix.
- Edge probabilities can be used for link prediction.
- Embeddings can be used directly for tasks like node-classification, clustering, etc.

Impossibility Result for Triangle-Dense Networks

Empirical Observation: Many real-world social networks are both sparse and triangle-dense. I.e., they have $O(n)$ edges (equivalently, $O(1)$ average degree) but also $\Omega(n)$ triangles.

Impossibility Result for Triangle-Dense Networks

Empirical Observation: Many real-world social networks are both sparse and triangle-dense. I.e., they have $O(n)$ edges (equivalently, $O(1)$ average degree) but also $\Omega(n)$ triangles.

Impossibility Theorem: [Seshadhri et al. 2020] For any embedding matrix $\mathbf{X} \in \mathbb{R}^{n \times k}$ and any constants c_1, c_2, if the graph generated by $\sigma(\mathbf{X}\mathbf{X}^T)$ has $c_1 n$ expected triangles incident to vertices of expected degree c_2, then $k \geq c_3 n / \log^2 n$.

Impossibility Result for Triangle-Dense Networks

Empirical Observation: Many real-world social networks are both sparse and triangle-dense. I.e., they have $O(n)$ edges (equivalently, $O(1)$ average degree) but also $\Omega(n)$ triangles.

Impossibility Theorem: [Seshadhri et al. 2020] For any embedding matrix $\mathbf{X} \in \mathbb{R}^{n \times k}$ and any constants c_1, c_2, if the graph generated by $\sigma(\mathbf{X}\mathbf{X}^T)$ has $c_1 n$ expected triangles incident to vertices of expected degree c_2, then $k \geq c_3 n / \log^2 n$.

I.e., any node-embedding that represents sparse, triangle-dense graphs must have dimension scaling nearly linearly in n.

This is a significant and very surprising limitation on a very broad class of embedding methods! How does it reconcile with the popularity of these methods for modeling graphs in practice?

Impossibility Result Proof

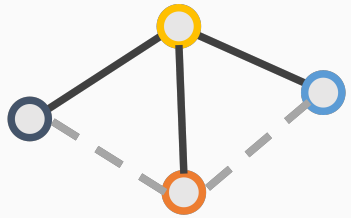
Extremely Rough Proof Idea:

- Assume for simplicity that all embeddings $x_1, \ldots, x_n \in \mathbb{R}^k$ have the same Euclidean norm.

Impossibility Result Proof

Extremely Rough Proof Idea:

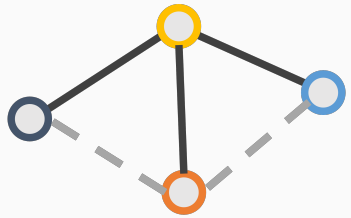
- Assume for simplicity that all embeddings $x_1, \ldots, x_n \in \mathbb{R}^k$ have the same Euclidean norm.
- In a sparse, triangle-dense graph, each wedge must be closed with $\Theta(1)$ probability, since there are $\Theta(n)$ wedges and $\Theta(n)$ triangles.



Impossibility Result Proof

Extremely Rough Proof Idea:

- Assume for simplicity that all embeddings $x_1, \ldots, x_n \in \mathbb{R}^k$ have the same Euclidean norm.
- In a sparse, triangle-dense graph, each wedge must be closed with $\Theta(1)$ probability, since there are $\Theta(n)$ wedges and $\Theta(n)$ triangles.

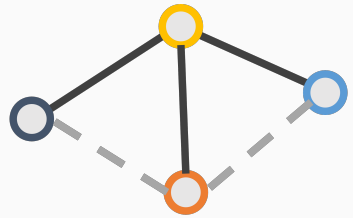


- To have such high probability edges, we must have $\langle x_i, x_j \rangle = \Theta(1)$, and thus $\|x_i\|_2, \|x_j\|_2 = \Theta(1)$ by Cauchy-Schwarz.

Impossibility Result Proof

Extremely Rough Proof Idea:

- Assume for simplicity that all embeddings $x_1, \ldots, x_n \in \mathbb{R}^k$ have the same Euclidean norm.
- In a sparse, triangle-dense graph, each wedge must be closed with $\Theta(1)$ probability, since there are $\Theta(n)$ wedges and $\Theta(n)$ triangles.

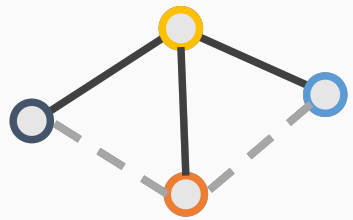


- To have such high probability edges, we must have $\langle x_i, x_j \rangle = \Theta(1)$, and thus $\|x_i\|_2, \|x_j\|_2 = \Theta(1)$ by Cauchy-Schwarz. Thus, $\mathbf{tr}(\mathbf{X}\mathbf{X}^T) = \sum_i \|x_i\|_2^2 = \Theta(n)$.

Impossibility Result Proof

Extremely Rough Proof Idea:

- Assume for simplicity that all embeddings $x_1, \ldots, x_n \in \mathbb{R}^k$ have the same Euclidean norm.
- In a sparse, triangle-dense graph, each wedge must be closed with $\Theta(1)$ probability, since there are $\Theta(n)$ wedges and $\Theta(n)$ triangles.



- To have such high probability edges, we must have $\langle x_i, x_j \rangle = \Theta(1)$, and thus $\|x_i\|_2, \|x_j\|_2 = \Theta(1)$ by Cauchy-Schwarz. Thus, $\mathbf{tr}(\mathbf{X}\mathbf{X}^T) = \sum_i \|x_i\|_2^2 = \Theta(n)$.
- At the same time, since the expected graph is sparse, $\|\mathbf{X}\mathbf{X}^T\|_F^2 = \sum_{i,j} \langle x_i, x_j \rangle^2 \lesssim \sum_{i,j} \langle x_i, x_j \rangle = O(n)$.

Impossibility Result Proof

So Far: If $\mathbf{X}\mathbf{X}^T$ yields a sparse triangle-dense network, we must have:

$$\mathrm{tr}(\mathbf{X}\mathbf{X}^T) = \Theta(n) \quad \text{and} \quad \|\mathbf{X}\mathbf{X}^T\|_F^2 = O(n).$$

Impossibility Result Proof

So Far: If XX^T yields a sparse triangle-dense network, we must have:

$$\mathrm{tr}(XX^T) = \Theta(n) \quad \text{and} \quad \|XX^T\|_F^2 = O(n).$$

This directly gives a lower bound on the embedding dimension:

$$k = \mathrm{rank}(XX^T) \geq \frac{\mathrm{tr}(XX^T)^2}{\|XX^T\|_F^2} = \frac{\Theta(n^2)}{\Theta(n)} = \Theta(n).$$

Breaking the Impossibility Result

Many popular node embedding methods use two embeddings per node, and thus produce an edge probability matrix $\sigma(\mathbf{X}\mathbf{Y}^T)$.

$\mathbf{tr}(\mathbf{X}\mathbf{Y}^T)$ can be very small, even if the rows of $\mathbf{X}$ and $\mathbf{Y}$ have relatively large norms! Thus the rank lower bound fails.

Breaking the Impossibility Result

Many popular node embedding methods use two embeddings per node, and thus produce an edge probability matrix $\sigma(\mathbf{X}\mathbf{Y}^T)$.

$\mathbf{tr}(\mathbf{X}\mathbf{Y}^T)$ can be very small, even if the rows of $\mathbf{X}$ and $\mathbf{Y}$ have relatively large norms! Thus the rank lower bound fails.

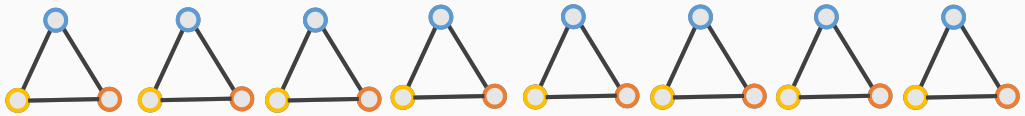
Exact Embedding Theorem: [Chanpuriya et al. 2020] Let $\mathbf{A} \in \{0,1\}^{n \times n}$ be the adjacency matrix of a graph with maximum degree D. Then for $k = O(D)$ there exist $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times k}$ such that $\mathbf{A} = \sigma(\mathbf{X}\mathbf{Y}^T)$.

I.e., we can exactly embed any graph with dimension proportional to its maximum degree. This includes very sparse, triangle-dense graphs.

This shows that the results of Seshhadri et al. critically hinge on the use of a symmetric embedding model.

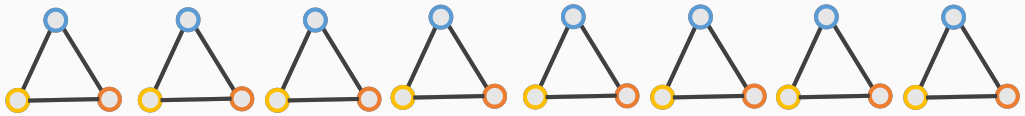
An Extreme Example

Consider a union of $n/3$ disjoint triangles. This graph has $\Theta(n)$ edges and $\Theta(n)$ triangles.



An Extreme Example

Consider a union of $n/3$ disjoint triangles. This graph has $\Theta(n)$ edges and $\Theta(n)$ triangles.



Symmetric Embeddings: Require $\Theta(n/\log^2 n)$ dimensions to represent such a graph by the Seshhadri et al. result.

Asymmetric Embeddings: Require just $O(1)$ dimensions to represent this graph exactly!

Proof Sketch

- Suffices to exhibit $\mathbf{X}, \mathbf{Y}$ such that $(\mathbf{X}\mathbf{Y}^T)_{ij} < 0$ whenever $\mathbf{A}_{ij} = 0$ and $(\mathbf{X}\mathbf{Y}^T)_{ij} > 0$ whenever $\mathbf{A}_{ij} = 1$. Then, for some large enough scaling factor c, we will have $\sigma(c \cdot \mathbf{X}\mathbf{Y}^T) = \mathbf{A}$.

$\mathbf{XY^T}$

-1	-2	4	-3	4
-2	-3	5	2	-3
1	3	-1	-1	-2
-1	4	-4	-1	2
6	-1	-5	2	-2

$\mathbf{A}$

0	0	1	0	1
0	0	1	1	0
1	1	0	0	0
0	1	0	0	1
1	0	0	1	0

Proof Sketch

- Suffices to exhibit $\mathbf{X}, \mathbf{Y}$ such that $(\mathbf{XY}^T)_{ij} < 0$ whenever $\mathbf{A}_{ij} = 0$ and $(\mathbf{XY}^T)_{ij} > 0$ whenever $\mathbf{A}_{ij} = 1$. Then, for some large enough scaling factor c, we will have $\sigma(c \cdot \mathbf{XY}^T) = \mathbf{A}$.

$\mathbf{XY}^\mathbf{T}$

-1	-2	4	-3	4
-2	-3	5	2	-3
1	3	-1	-1	-2
-1	4	-4	-1	2
6	-1	-5	2	-2

A

0	0	1	0	1
0	0	1	1	0
1	1	0	0	0
0	1	0	0	1
1	0	0	1	0

- I.e., it suffices to bound the sign-rank of $\mathbf{A}$, a well-studied property in the communication complexity literature.
- This can be done via polynomial interpolation techniques.

Proof Sketch

Let $\mathbf{X}$ be a Vandermonde matrix with $\mathbf{X}_{ab} = a^{b-1}$ and $\mathbf{Y}^T$ be a coefficient matrix.

Vandermonde matrix $\mathbf{X}$

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \end{matrix}$$

$\vdots$

$\vdots$

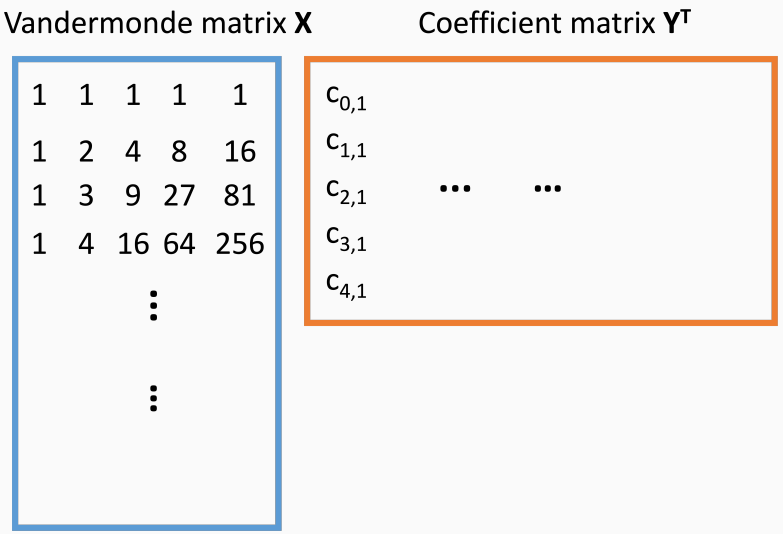
Coefficient matrix $\mathbf{Y}^T$

$$\begin{matrix} c_{0,1} & & \\ c_{1,1} & & \\ c_{2,1} & \cdots & \cdots \\ c_{3,1} & & \\ c_{4,1} & & \end{matrix}$$

- Each column of $\mathbf{XY}^T$ is a degree $k-1$ polynomial evaluated at the integers $1, \ldots, n$.

Proof Sketch

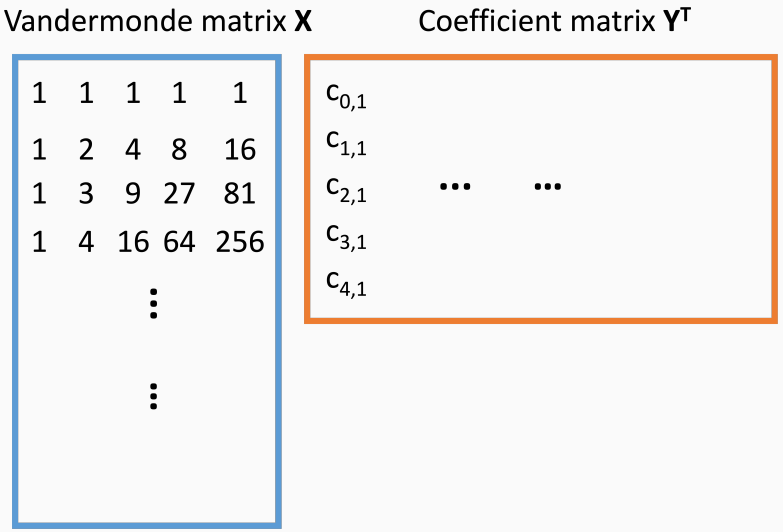
Let $\mathbf{X}$ be a Vandermonde matrix with $\mathbf{X}_{ab} = a^{b-1}$ and $\mathbf{Y}^T$ be a coefficient matrix.



- Each column of $\mathbf{XY}^T$ is a degree $k-1$ polynomial evaluated at the integers $1, \ldots, n$.
- We need this polynomial to be positive at the locations where $\mathbf{A}_{ij} = 1$. There are at most D such locations, so this can be done using a degree $2D$ polynomial.

Proof Sketch

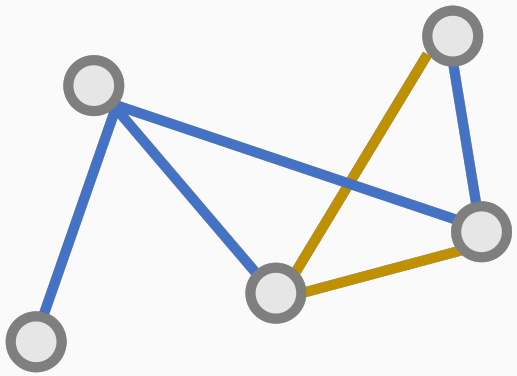
Let $\mathbf{X}$ be a Vandermonde matrix with $\mathbf{X}_{ab} = a^{b-1}$ and $\mathbf{Y}^T$ be a coefficient matrix.



- Each column of $\mathbf{XY}^T$ is a degree $k-1$ polynomial evaluated at the integers $1, \ldots, n$.
- We need this polynomial to be positive at the locations where $\mathbf{A}_{ij} = 1$. There are at most D such locations, so this can be done using a degree $2D$ polynomial.
- Thus, $k = 2D + 1$ suffices to match the sign of $\mathbf{A}$ at all entries.

Extension to Bounded Arboricity Graphs

We can extend our bound to show that when a graph has arboricity α, it can be exactly embedded using $k = O(\alpha^2)$ dimensions.



By the Nash-Williams theorem, the arboricity is the maximum average degree of any induced subgraph, and is typically small for sparse real-word networks, even when the maximum degree is large.

$$\alpha = \left\lceil \max_{S \subseteq V} \frac{E(S)}{V(S) - 1} \right\rceil .$$

Exact Embeddings of Real-World Networks

In practice, we can find very low-rank exact embeddings (sometimes substantially lower than our theoretical bounds) via a simple logistic PCA model (i.e., σ is the logistic function) trained with L-BFGS.

Dataset	# Nodes	Mean Degree	**Exact Factorization Dimension**
Pubmed	19 581	4.48	48
ca-HepPh	11 204	21.0	32
BlogCatalog	10 312	64.8	128
Citeseer	3 327	2.74	16
Cora	2 708	3.90	16

Taking a Step Back

Our exact embedding results demonstrate that low-dimensional, asymmetric embeddings can in fact be very effective in representing real-world networks. But how useful are these exact embeddings?

Taking a Step Back

Our exact embedding results demonstrate that low-dimensional, asymmetric embeddings can in fact be very effective in representing real-world networks. But how useful are these exact embeddings?

- Exact embeddings don't yield interesting generative models: if we generate a graph from the probability matrix $\sigma(\mathbf{X}\mathbf{Y}^T) = \mathbf{A}$, where $\mathbf{A} \in \{0,1\}^{n \times n}$ is the adjacency matrix of the input graph, it equals the input graph with probability 1.
- $\sigma(\mathbf{X}\mathbf{Y}^T)$ will also be useless e.g., in link prediction tasks. No 'generalization'. We find that the embeddings are also not particularly useful in node classification, clustering, etc.
- Is there an inherent trade-off here? Does achieving sparsity + high triangle density necessitate simply memorizing the input graph?

Limits on Edge Independent Graph Models

Consider a probability matrix $\mathbf{P} \in [0,1]^{n \times n}$ (e.g., $\mathbf{P} = \sigma(\mathbf{X}\mathbf{Y}^T)$) and the distribution $\mathcal{G}(\mathbf{P})$ on graphs where each edge is added independently with probability $\mathbf{P}_{ij}$.

Let $Ov(\mathbf{P}) = \mathbb{E}_{G_1, G_2 \sim \mathcal{G}(\mathbf{P})} |E(G_1) \cap E(G_2)|$ be the expected number of shared edges in two graphs drawn from $\mathcal{G}(\mathbf{P})$. The 'overlap'.

Let $\Delta(G)$ be the number of triangles in G.

Limits on Edge Independent Graph Models

Consider a probability matrix $\mathbf{P} \in [0,1]^{n \times n}$ (e.g., $\mathbf{P} = \sigma(\mathbf{X}\mathbf{Y}^T)$) and the distribution $\mathcal{G}(\mathbf{P})$ on graphs where each edge is added independently with probability $\mathbf{P}_{ij}$.

Let $Ov(\mathbf{P}) = \mathbb{E}_{G_1, G_2 \sim \mathcal{G}(\mathbf{P})} |E(G_1) \cap E(G_2)|$ be the expected number of shared edges in two graphs drawn from $\mathcal{G}(\mathbf{P})$. The 'overlap'.

Let $\Delta(G)$ be the number of triangles in G.

Theorem: [Chanpuriya et al. 2021]

$$\mathbb{E}_{G \sim \mathcal{G}(\mathbf{P})} |\Delta(G)| \lesssim Ov(\mathbf{P})^{3/2}.$$

E.g., if our model generates graphs with $\Theta(n)$ edges but just $O(\sqrt{n})$ overlap, these graphs have $O(n^{3/4})$ triangles in expectation.

Limits on Edge Independent Graph Models

Consider a probability matrix $\mathbf{P} \in [0,1]^{n \times n}$ (e.g., $\mathbf{P} = \sigma(\mathbf{X}\mathbf{Y}^T)$) and the distribution $\mathcal{G}(\mathbf{P})$ on graphs where each edge is added independently with probability $\mathbf{P}_{ij}$.

Let $Ov(\mathbf{P}) = \mathbb{E}_{G_1, G_2 \sim \mathcal{G}(\mathbf{P})} |E(G_1) \cap E(G_2)|$ be the expected number of shared edges in two graphs drawn from $\mathcal{G}(\mathbf{P})$. The 'overlap'.

Let $\Delta(G)$ be the number of triangles in G.

Theorem: [Chanpuriya et al. 2021]

$$\mathbb{E}_{G \sim \mathcal{G}(\mathbf{P})} |\Delta(G)| \lesssim Ov(\mathbf{P})^{3/2}.$$

E.g., if our model generates graphs with $\Theta(n)$ edges but just $O(\sqrt{n})$ overlap, these graphs have $O(n^{3/4})$ triangles in expectation.

An analog of the Seshadhri et al. result but without any requirement that $\mathbf{P}$ is generated from low-dimensional embeddings!

Conclusions

- We have demonstrated that asymmetric node embeddings can be much more powerful than symmetric embeddings in representing sparse, triangle-dense graphs.
- At the same time, we show that if these embeddings are used to form an edge-independent generative model, they are still limited, regardless of their dimensionality.
- They face a trade-off between memorization and representation.
- This indicates a possible reason for preferring generative models that incorporate dependencies between edges (e.g., graphRNNs) over edge-independent models (e.g., NetGAN, CELL, variational graph autoencoders, Graphite, MolGAN)

Open Questions

- A key feature of the models we study is that $\mathbf{A}$ is approximated by $\sigma(\mathbf{X}\mathbf{Y}^T)$ – a low-rank approximation with a non-linearity applied. Classic spectral embeddings do not use a nonlinearity.
- In our work on InfiniteWalk, we show that in a natural limit, DeepWalk reduces to a classic spectral embedding method with a simple non-linearity. Can we understand exactly why this nonlinearity is so useful?
- Are there any applications of our exact embedding results outside understanding representational power?
- Can we improve our bounds on edge-independent models and extend them to simple classes of edge-dependent models?