PieClam: Inclusive Exclusive Cluster Affiliation Model With Prior

Daniel Zilberg, Ron Levie

Background

Graph Representation Learning

Setting: An **undirected, featureless graph** $G = ([N], E)$ with an adjacency matrix $\mathbf{A} = \{a_{nm} \in \{0,1\}\}_{n,m=1}^{N}$. An edge between nodes n and m is denoted by $n \sim m$. A non-edge by $n \not\sim m$.

Goal: Build a **universal auoencoder**, which can represent any graph G of any size N with a fixed budget of parameters per node C.

BigClam [1]: Inclusive Community Affiliation

Inclusive communities: Common membership raises the probability to connect: $P(n \sim m | \mathbf{f}_n, \mathbf{f}_m) = 1 - e^{-\mathbf{f}_n^\top \mathbf{f}_m}$

The probability for the entire graph is

$$P(E|\mathbf{F}) = \sqrt{\prod_{n\in[N]} \prod_{m\in\mathcal{N}(n)} P(n \sim m|\mathbf{f}_n, \mathbf{f}_m) \prod_{m\notin\mathcal{N}(n)} P(n \not\sim m)|\mathbf{f}_n, \mathbf{f}_m)}$$

The log likelihood is

$$l(\mathbf{F}) = \frac{1}{2}\sum_{n\in[N]} \Big(\sum_{m\in\mathcal{N}(n)} \log(1 - e^{-\mathbf{f}_n^\top \mathbf{f}_m}) - \sum_{m\notin\mathcal{N}(n)} \mathbf{f}_n^\top \mathbf{f}_m \Big).$$

Optimize with gradient updates

$$\nabla_{\mathbf{f}_n} l = \sum_{m\in\mathcal{N}(n)} \mathbf{f}_m (1 - e^{-\mathbf{f}_n^\top \mathbf{f}_m})^{-1} - \sum_{n\in[N]} \mathbf{f}_m + \mathbf{f}_n.$$

Can be implemented as an **MPNN**.

Bipartite Blindness Of BigClam

Triangle inequality: if $n \sim k$ and $m \sim k$ then $\mathbf{f}_n^\top \mathbf{f}_k$ and $\mathbf{f}_m^\top \mathbf{f}_k$ are large and therefore $\mathbf{f}_n^\top \mathbf{f}_m$ is also large which implies $m \sim n$ with high probability.

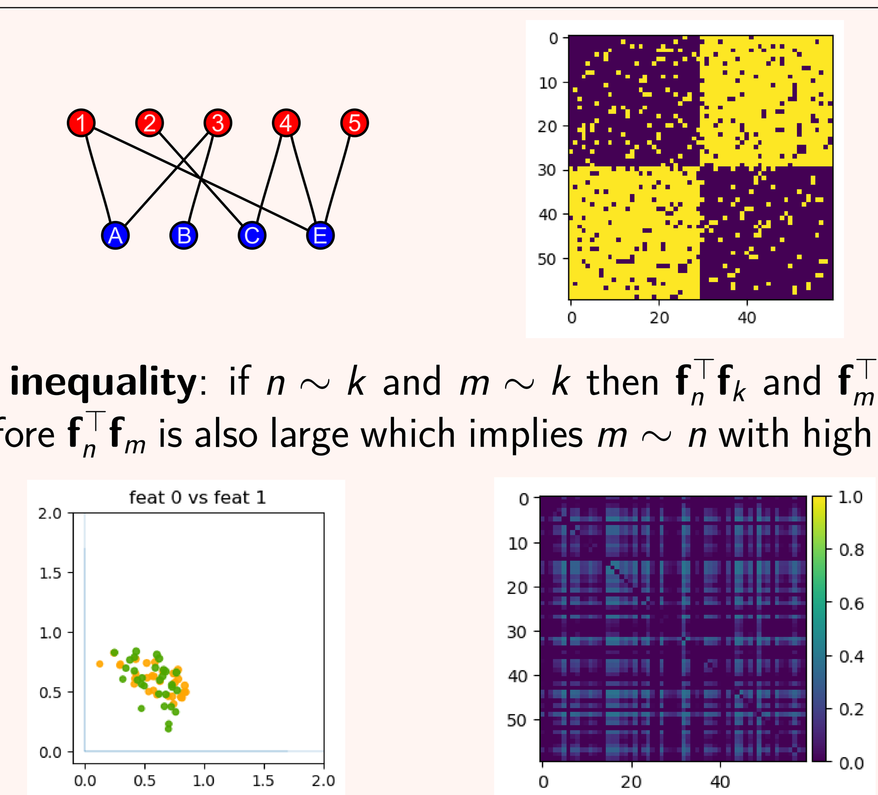

Figure 1. Bipartite autoencoding with BigClam

Inner product decoding is not universal!

Quick fix: **use node features!**, or...

Innovations

IeClam: Inclusive Exclusive Clustering

Exclusive communities: common membership *reduces* the probability to connect.

Representation: $\mathbf{f}_n = (\mathbf{t}_n, \mathbf{s}_n)$ where $\mathbf{t}_n$ are inclusive and $\mathbf{s}_n$ are exclusive communities.

L-Product: Instead of inner product use

$$\mathbf{f}_n^\top \mathbf{L} \mathbf{f}_m = \mathbf{t}_n^\top \mathbf{t}_m - \mathbf{s}_n^\top \mathbf{s}_m$$

where $\mathbf{L} = \mathrm{diag}(1, \ldots, 1, -1, \ldots, -1)$.

Edge probability:

$$P(n \sim m|\mathbf{f}_n, \mathbf{f}_m) = 1 - e^{-\mathbf{f}_n^\top \mathbf{L} \mathbf{f}_m}$$

Log likelihood:

$$l(\mathbf{F}) = \frac{1}{2}\sum_{n\in[N]} \Big(\sum_{m\in\mathcal{N}(n)} \log(1 - e^{-\mathbf{f}_n^\top \mathbf{L} \mathbf{f}_m}) - \sum_{m\notin\mathcal{N}(n)} \mathbf{f}_n^\top \mathbf{L} \mathbf{f}_m \Big)$$

Bipartite encoding: A bipartite graph can be encoded by embedding part 1 to (b, b) and part 2 to $(b, -b)$ where $b \in \mathbb{R}_+$.

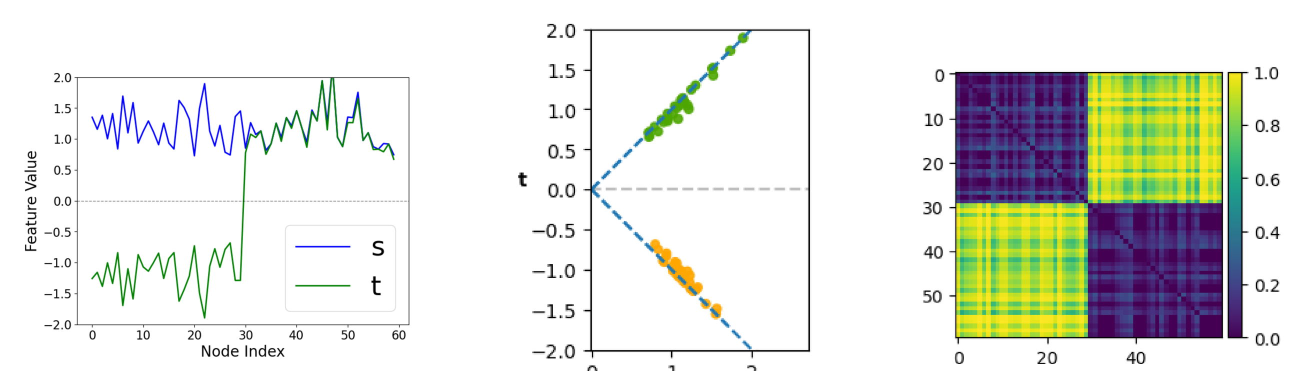

Figure 2. Bipartite encoding with IeClam. **Left**: community value per node. **Center**: embedding space with one **s** and one **t** community, **Right**: reconstucted adjacency.

PieClam

Extend BigClam and IeClam into **Generative models**: Learn a joint probability distribution

$$p(E \wedge \mathbf{F}) = P(E|\mathbf{F})p(\mathbf{F})$$

Log likelihood loss (assuming independent features):

$$l(\mathbf{F}) = \sum_{n\in[N]} \Bigg(\frac{1}{2} \Big(\sum_{m\in\mathcal{N}(n)} \log(e^{\mathbf{f}_n^\top \mathbf{L} \mathbf{f}_n} - 1) - \mathbf{f}_n^\top \mathbf{L} \sum_{n\in[N]} \mathbf{f}_m + \mathbf{f}_n^\top \mathbf{L} \mathbf{f}_n \Big) + \log\left(p(\mathbf{f}_n)\right) \Bigg)$$

Neural network prior: Model $p(\mathbf{F})$ as a **normalizing flow** [2].

Optimization: Alternating optimization between features and prior parameters.

PClam: Extend BigClam into a generative model by the same method.

Theory

Universality in Autoencoders

A family of code spaces $\{\mathbb{R}^C\}_{C\in\mathbb{N}}$ and corresponding decoders $\{\mathbf{D}_C : \mathbb{R}^{2C} \to [0,1]^{N\times N}\}_{C\in\mathbb{N}}$ is **universal** w.r.t. to the distance $d(.,.)$ if for every $\epsilon > 0$ there is $C \in \mathbb{N}$ (depending only on ϵ) such that for every $N \in \mathbb{N}$ and every graph with adjacency matrix $\mathbf{A} \in [0,1]^{N\times N}$ there are N points $\{z_n \in \mathbb{R}^C\}_{n=1}^N$ such that

$$d(\mathbf{D}_M(\mathbf{z}), \mathbf{A}) < \epsilon.$$

Log Cut Distance

Log Cut Metric between probabilistic graph models:

$$D_\square(\mathbf{P}||\mathbf{Q}) := \frac{1}{N^2} \sup_{\mathcal{U},\mathcal{V}\subset[N]} \left(\left| \log\Big(\prod_{n\in\mathcal{U}} \prod_{m\in\mathcal{V}} \frac{1-p_{n,m}}{1-q_{n,m}} \Big) \right| \right) = \| \log(1-\mathbf{P}) - \log(1-\mathbf{Q}) \|_\square$$

Interpretation: The biggest part of the probabilistic model P that can't be explained by the model Q.

For realizations (when one matrix has elements 1), regularization is added:

$$D_\square(\mathbf{P}||\mathbf{A}) := \inf_{0<d\leq 1} \left(d + \frac{1}{N^2} \sup_{\mathcal{U},\mathcal{V}\subset[N]} \left| \log\Big(\prod_{n\in\mathcal{U}} \prod_{m\in\mathcal{V}} \frac{1-p_{n,m}}{1-(1-d)a_{n,m}} \Big) \right| \right)$$

Our Experiments show that the log cut distance goes down when maximizing the log likelihood.

Universality Theorem for IeClam

Theorem: IeClam (and hence PieClam) is universal with respect to the log cut distance with $O(\epsilon^{-2})$ communities.

BigClam limitation: Not universal - embedding dimension must depend on number of nodes.

PieClam vs BigClam

The innovations of Clam methods are summarized in the table below.

Model	Generative	Universal
BigClam	X	X
PClam	✓	X
IeClam	X	✓
PieClam	✓	✓

Experiments

1. Prior Reconstruction

Sample nodes from synthetic priors (circles in $\mathcal{T}$, moons in $\mathbb{R}_+^2$) and connect with clam probability. Reset affiliation features and fit models to reconstruct shapes

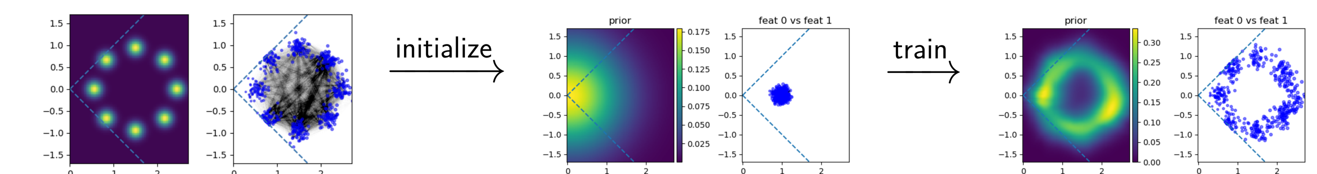

Figure 3. Node features sampled from synthetic priors in $\mathcal{T}$ and reconstructed with PieClam

2. SBM Reconstruction

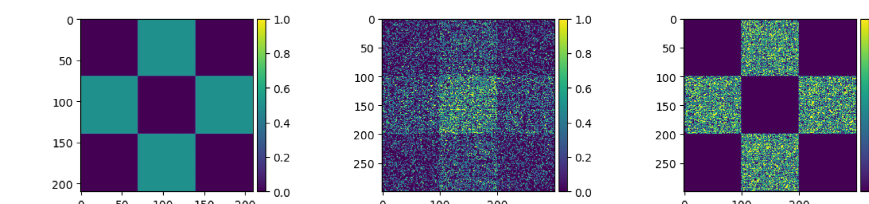
Figure 4. Left to right: Original SBM with 3 classes and 9 blocks; Adjacency matrix of the fitted BigClam graph, with six communities; Adjacency matrix of the fitted IeClam graph, with four communities.

3. Unsupervised Anomaly Detection Results

Method	Reddit	Elliptic	Photo
(S)- IeClam	**64.1**	43.6	57.7
(S) - PieClam	*64.0	43.5	<u>59.0</u>
(P) - PieClam	46.8	**63.2**	45.7
(PS) - PieClam	<u>64.0</u>	<u>53.8</u>	**59.0**
(S) - BigClam	63.7	43.4	*58.1
DOMINANT	51.1	29.6	51.4
AnomalyDAE	50.9	*49.6	50.7
OCGNN	52.5	25.8	53.1
AEGIS	53.5	45.5	55.2
GAAN	52.2	25.9	43.0
TAM	60.6	40.4	56.8

Table 1. Anomaly detection AUC scores. First place in **boldface**, second with <u>underline</u>, third with *star.

4. Link Prediction Results

Method	Squirrel	Photo	Texas	JH55
PieClam	**98.7**	**98.4**	**85.0**	95.5*
BigClam	<u>98.5</u>	97.4*	78.2*	94.9
VGAE	98.2	94.9	68.6	92.8
GAT	98.0	97.3	68.5	94.3
LINKX	98.1	97.0	75.8	93.4
AA	97.1	97.4	53.1	<u>96.1</u>
DisenLink	98.3*	<u>97.9</u>	<u>81.0</u>	**97.5**

Table 2. Link prediction AUC scores. First place in **boldface**, second with <u>underline</u>, third with *star.

References

[1] Jaewon Yang and Jure Leskovec.
Overlapping community detection at scale: a nonnegative matrix factorization approach.
In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596, 2013.

[2] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio.
Density estimation using real nvp.
arXiv preprint arXiv:1605.08803, 2016.