Stochastic Blockmodel Approximation of a Graphon: Theory and Consistent Estimation

Edoardo M. Airoldi[1], Thiago B. Costa[2,1], Stanley H. Chan[2,1]

[1]Department of Statistics, Harvard University [2]Harvard School of Engineering and Applied Sciences

Privacy Tools for Sharing Research Data
A National Science Foundation Secure and Trustworthy Cyberspace Project

Abstract

Non-parametric approaches for analyzing network data based on exchangeable graph models (ExGM) have recently gained interest. The key object that defines an ExGM is often referred to as a *graphon*. This non-parametric perspective on network modeling poses challenging questions on how to make inference on the graphon underlying observed data. In this paper, we propose a computationally efficient procedure to estimate a graphon from a set of observed networks generated from it. This procedure is based on a stochastic blockmodel approximation (SBA) of the graphon. We show that, by approximating the graphon with a stochastic block model, the graphon can be consistently estimated, that is, the estimation error vanishes as the size of the graph approaches infinity.

Problem

Graphons can be seen as kernel functions for random network models. To construct an n-vertex random graph $\mathcal{G}(n,w)$ for a given w, we first assign a random label $u_i \sim \text{Uniform}[0,1]$ to each vertex $i \in \{1,\ldots,n\}$, and connect any two vertices i and j with probability $w(u_i,u_j)$, *i.e.*,

$$\Pr\left(G[i,j]=1 \mid u_i,u_j\right) = w(u_i,u_j), \qquad i,j=1,\ldots,n,$$

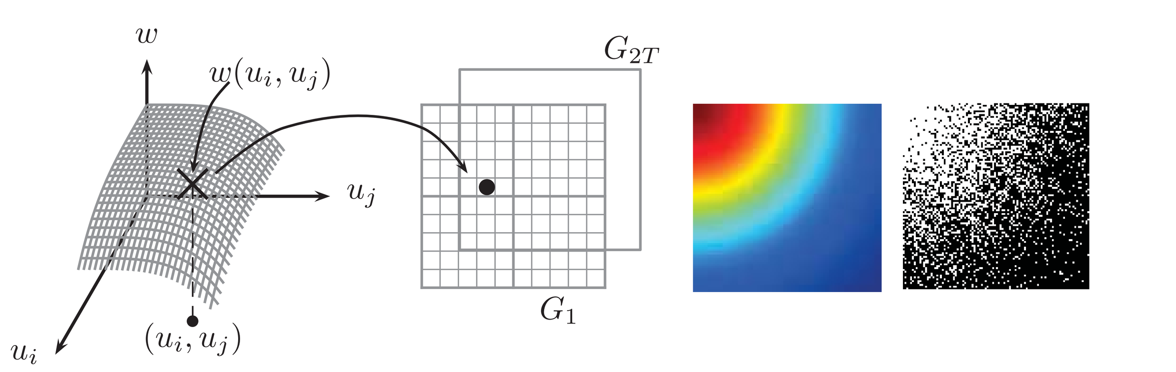

Figure 1: [Left] We draw i.i.d. samples u_i, u_j from Uniform[0,1] and assign $G_t[i,j]=1$ with probability $w(u_i,u_j)$, for $t=1,\ldots,2T$. [Middle] Heat map of a graphon w. [Right] A random graph generated by the graphon shown in the middle.

The problem of interest is defined as follows: Given a sequence of $2T$ observed *directed* graphs $G_1,\ldots,G_{2T}$, can we make an estimate $\widehat{w}$ of w, such that $\widehat{w} \to w$ with high probability as $n \to \infty$? (In this problem we assume that the observed graphs share the same set of vertices, in a way that the i-th vertex have the same position u_i in all graphs)

Similarity of graphon slices

To measure the similarity between two labels using the graphon slices, we define the following distance

$$\begin{aligned} d_{ij} &= \frac{1}{2}\left(\int_0^1 \left[w(x,u_i)-w(x,u_j)\right]^2 dx + \int_0^1 \left[w(u_i,y)-w(u_j,y)\right]^2 dy\right) \\ &= \frac{1}{2}\left[(c_{ii}-c_{ij}-c_{ji}+c_{jj}) + (r_{ii}-r_{ij}-r_{ji}+r_{jj})\right] \end{aligned}$$

where

$$c_{ij} = \int_0^1 w(x,u_i)w(x,u_j)dx \quad \text{and} \quad r_{ij} = \int_0^1 w(u_i,y)w(u_j,y)dy.$$

We consider the following estimators for c_{ij} and r_{ij}:

$$\hat{c}^k_{ij} = \frac{1}{T^2}\left(\sum_{1\le t_1\le T} G_{t_1}[k,i]\right)\left(\sum_{T<t_2\le 2T} G_{t_2}[k,j]\right),$$

$$\hat{r}^k_{ij} = \frac{1}{T^2}\left(\sum_{1\le t_1\le T} G_{t_1}[i,k]\right)\left(\sum_{T<t_2\le 2T} G_{t_2}[j,k]\right).$$

Summing all possible k's yields an estimator $\hat{d}_{ij}$ that looks similar to d_{ij}:

$$\hat{d}_{ij} = \frac{1}{2}\left[\frac{1}{S}\sum_{k\in\mathcal{S}}\left\{\left(\hat{r}^k_{ii}-\hat{r}^k_{ij}-\hat{r}^k_{ji}+\hat{r}^k_{jj}\right)+\left(\hat{c}^k_{ii}-\hat{c}^k_{ij}-\hat{c}^k_{ji}+\hat{c}^k_{jj}\right)\right\}\right],$$

where $\mathcal{S}=\{1,\ldots,n\}\backslash\{i,j\}$ is the set of summation indices.

Theorem 1 *The estimator $\hat{d}_{ij}$ for d_{ij} is unbiased and satisfies*

$$\mathbb{P}(|d_{ij}-\hat{d}_{ij}|>\epsilon) \le 8e^{-\frac{S\epsilon^2}{32/T+8\epsilon/3}},$$

for any $\epsilon>0$.

Algorithm (SBA)

To cluster the unknown labels $\{u_1,\ldots,u_n\}$ we propose a greedy approach as shown in Algorithm 1. Starting with $\Omega=\{u_1,\ldots,u_n\}$, we randomly pick a node i_p and call it the *pivot*. Then for all other vertices $i_v \in \Omega\backslash\{i_p\}$, we compute the distance $\widehat{d}_{i_p,i_v}$ and check whether $\widehat{d}_{i_p,i_v} < \Delta^2$ for some precision parameter $\Delta>0$. If $\widehat{d}_{i_p,i_v} < \Delta^2$, then we assign i_v to the same block as i_p. Therefore, after scanning through Ω once, a block $\widehat{B}=\{i_p,i_{v_1},i_{v_2},\ldots\}$ will be defined. By updating Ω as $\Omega \leftarrow \Omega\backslash\widehat{B}$, the process repeats until $\Omega=\emptyset$.

Algorithm 1: Clustering the vertices

```
Input: Observed graphs G_1,...,G_2T and precision parameter Δ
Output: Estimated stochastic blocks B̂_1,...,B̂_K
Initialize: Ω = {1,...,n}, and k = 1;
while Ω ≠ ∅ do
    Randomly choose a vertex i_p from Ω and assign it as the pivot for B̂_k:
    B̂_k ← i_p;
    for i_v ∈ Ω\{i_p} do
        Compute the distance estimate d̂_{i_p,i_v};
        if d̂_{i_p,i_v} ≤ Δ² then
            assign i_v as a member of B̂_k: B̂_k ← i_v;
        end
    end
    Update Ω ← Ω\B̂_k;
    Update k ← k + 1;
end
```

Once the blocks $\widehat{B}_1,\ldots,\widehat{B}_K$ are defined, we can then determine $\widehat{w}(u_i,u_j)$ by computing the empirical frequency of edges that are present across blocks $\widehat{B}_i$ and $\widehat{B}_j$:

$$\widehat{w}(u_i,u_j) = \frac{1}{|\widehat{B}_i||\widehat{B}_j|}\sum_{i_x\in\widehat{B}_i}\sum_{j_y\in\widehat{B}_j}\frac{1}{2T}\left(G_1[i_x,j_y]+G_2[i_x,j_y]+\ldots+G_{2T}[i_x,j_y]\right)$$

where $\widehat{B}_i$ is the block containing u_i.

Consistency

The performance of the Algorithm 1 depends on the number of blocks it defines. On the one hand, it is desirable to have more blocks so that the graphon can be finely approximated. But on the other hand, if the number of blocks is too large then each block will contain only few vertices, what might be a problem because in order to estimate the probabilities of connection, a sufficient number of vertices in each block is required. The trade-off between these two cases is controlled by the precision parameter Δ: a large Δ generates few large clusters, while small Δ generates many small clusters. The following theorems shows how to balance the choice of Δ in order to achieve consistency.

Theorem 2 *Let Δ be the accuracy parameter and K be the number of blocks estimated by Algorithm 1, then*

$$\Pr\left[K > \frac{QL\sqrt{2}}{\Delta}\right] \le 8n^2 e^{-\frac{S\Delta^4}{128/T+16\Delta^2/3}},$$

where L is the Lipschitz constant and Q is the number of Lipschitz blocks in w.

Theorem 3 *If $S\in\Theta(n)$ and $\Delta\in\omega\left(\left(\frac{\log(n)}{n}\right)^{\frac{1}{4}}\right)\cap o(1)$, then*

$$\lim_{n\to\infty}\mathbb{E}[\text{MAE}(\widehat{w})]=0 \text{ and } \lim_{n\to\infty}\mathbb{E}[\text{MSE}(\widehat{w})]=0.$$

where

$$\text{MSE}(\widehat{w}) = \frac{1}{n^2}\sum_{i_v=1}^{n}\sum_{j_v=1}^{n}\left(w(u_{i_v},u_{j_v})-\widehat{w}(u_{i_v},u_{j_v})\right)^2$$

$$\text{MAE}(\widehat{w}) = \frac{1}{n^2}\sum_{i_v=1}^{n}\sum_{j_v=1}^{n}\left|w(u_{i_v},u_{j_v})-\widehat{w}(u_{i_v},u_{j_v})\right|.$$

Choosing parameter

In practice, we estimate Δ using a cross-validation scheme to find the optimal 2D histogram bin width. The idea is to test a sequence of potential values of Δ and seek the one that minimizes the cross validation risk, defined as

$$\widehat{J}(\Delta) = \frac{2}{h(n-1)} - \frac{n+1}{h(n-1)}\sum_{j=1}^{K}\widehat{p}_j^2,$$

where $\widehat{p}_j = |\widehat{B}_j|/n$ and $h=1/K$.

Algorithm 2: Cross Validation

```
Input: Graphs G_1,...,G_2T
Output: Blocks B̂_1,...,B̂_K, and optimal Δ
for a sequence of Δ's do
    Estimate blocks B̂_1,...,B̂_K from G_1,...,G_2T. [Algorithm 1];
    Compute p̂_j = |B̂_j|/n, for j = 1,...,K;
    Compute Ĵ(Δ) = 2/(h(n-1)) - (n+1)/(h(n-1)) Σ_{j=1}^K p̂_j², with h = 1/K;
end
Pick the Δ with minimum Ĵ(Δ), and the corresponding B̂_1,...,B̂_K;
```

Experiments

For the purpose of comparison, we consider (i) the universal singular value thresholding (USVT) [Cha2012]; (ii) the largest-gap algorithm (LG) [CRD2012]; (iii) matrix completion from few entries (OptSpace) [KMO2010].

- **Estimating stochastic blockmodels** We generate (arbitrarily) a graphon

$$w = \begin{bmatrix} 0.8 & 0.9 & 0.4 & 0.5 \\ 0.1 & 0.6 & 0.3 & 0.2 \\ 0.3 & 0.2 & 0.8 & 0.3 \\ 0.4 & 0.1 & 0.2 & 0.9 \end{bmatrix}, \qquad (1)$$

which represents a piecewise constant function with 4×4 equi-space blocks. The following figures show the asymptotic behavior of the algorithms when n grows (left), and the estimation error of SBA algorithm as T grows for graphs of size 200 vertices (right).

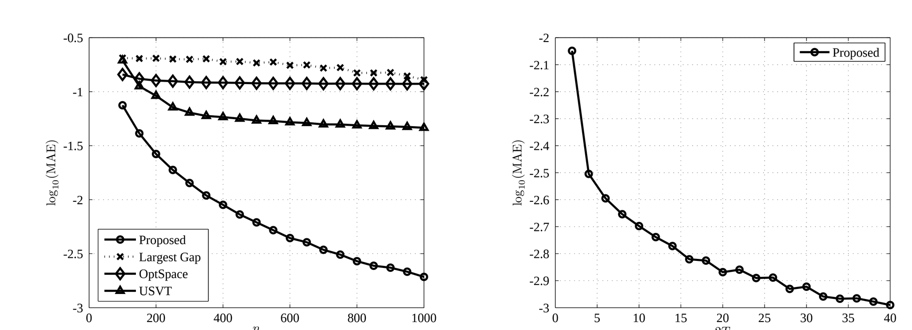

Figure 2: [Left] MAE reduces as graph size grows. For the fairness of the amount of data that can be used, we use $\frac{n}{2}\times\frac{n}{2}\times 2$ observations for SBA, and $n\times n\times 1$ observation for USVT and LG. [Right] MAE of the proposed SBA algorithm reduces when more observations T is available. Both plots are averaged over 100 independent trials.

- **Accuracy as a function of growing number of blocks**

Our second experiment is to evaluate the performance of the algorithms as K, the number of blocks, increases. To this end, we consider a sequence of K, and for each K we generate a graphon w of $K\times K$ blocks. Each entry of the block is a random number generated from Uniform$[0,1]$. Same as the previous experiment, we fix $n=200$ and $T=1$. The experiment is repeated over 100 trials so that in every trial a different graphon is generated. The result shown in (a) indicates that while estimation error increases as K grows, the proposed SBA algorithm still attains the lowest MAE for all K.

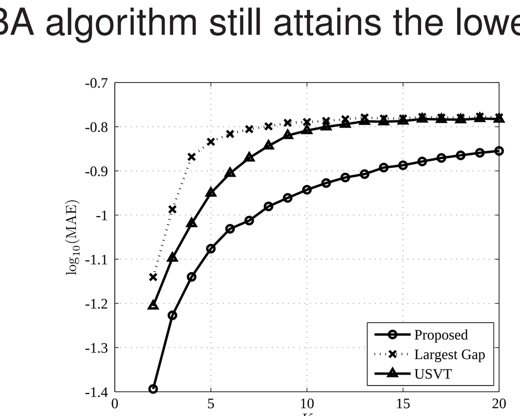

Figure 3: As K increases, SBA still attains the lowest MAE. Here, we use $\frac{n}{2}\times\frac{n}{2}\times 2$ observations for SBA, and $n\times n\times 1$ observation for USVT and LG

Experiments

- **Estimation with missing edges** Our next experiment is to evaluate the performance of proposed SBA algorithm when there are missing edges in the observed graph. To model missing edges, we construct an $n\times n$ binary matrix M with probability $\Pr[M[i,j]=0]=\xi$, where $0\le\xi\le 1$ defines the percentage of missing edges. Given ξ, $2T$ matrices are generated with missing edges, and the observed graphs are defined as $M_1\odot G_1,\ldots,M_{2T}\odot G_{2T}$, where $\odot$ denotes the element-wise multiplication. The goal is to study how well SBA can reconstruct the graphon $\widehat{w}$ in the presence of missing links.

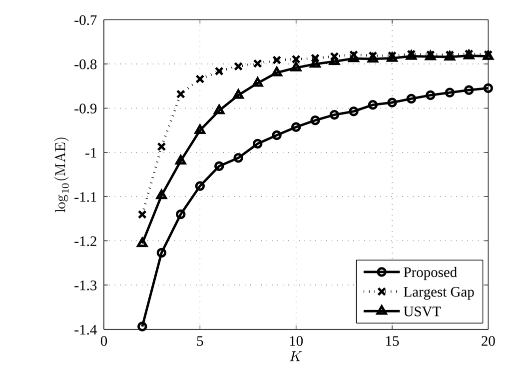

Figure 4: Estimation of graphon in the presence of missing links: As the amount of missing links increases, estimation error also increases.

- **Estimating continuous graphons**

Our final experiment is to evaluate the proposed SBA algorithm in estimating continuous graphons. Here, we consider two of the graphons reported in [Cha 2012]:

$$w_1(u,v) = \frac{1}{1+\exp\{-50(u^2+v^2)\}}, \quad \text{and} \quad w_2(u,v)=uv,$$

where $u,v\in[0,1]$. Here, w_2 can be considered as a special

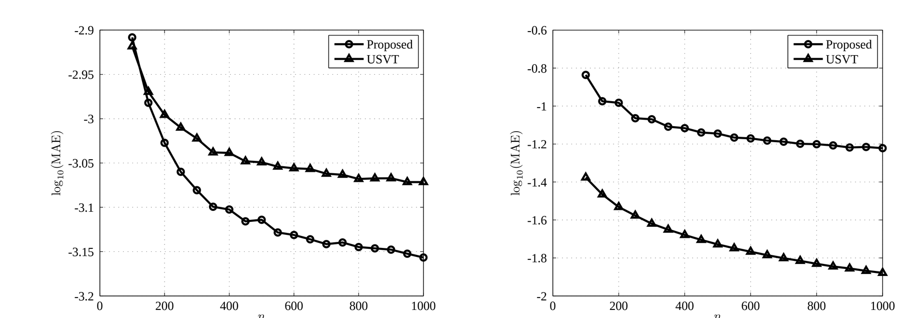

Figure 5: Comparison between SBA and USVT in estimating two continuous graphons w_1 and w_2. Evidently, SBA performs better for high-rank w_1 (left) and worse for low-rank w_2 (right).

Concluding remarks

We presented a new computational tool for estimating graphons. The proposed algorithm approximates the continuous graphon by a stochastic block-model, in which the first step is to cluster the unknown vertex labels into blocks by using an empirical estimate of the distance between two graphon slices, and the second step is to build an empirical histogram to estimate the graphon. Complete consistency analysis of the algorithm is derived. The algorithm was evaluated experimentally, and we found that the algorithm is effective in estimating block structured graphons.

References

- [Cha2012] S. Chatterjee. Matrix estimation by universal singular value thresholding. ArXiv:1212.1247. 2012.
- [CDR2012] A. Channarond, J. Daudin, and S. Robin. Classification and estimation in the Stochastic Blockmodel based on the empirical degrees. Electronic Journal of Statistics, 6:2574-2601, 2012.
- [KMO2010] R.H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. IEEE Trans. Information Theory, 56:2980-2998, Jun. 2010.
- [LOGR2012] J.R. Lloyd, P. Orbanz, Z. Ghahramani, and D.M. Roy. Random function priors for exchangeable arrays with applications to graphs and relational data. In Neural Information Processing Systems (NIPS), 2012.
- [MGJ2009] K.T. Miller, T.L. Griffiths, and M.I. Jordan. Nonparametric latent fature models for link prediction. In Neural Information Processing Systems (NIPS), 2009.