Graphons, mergeons, and so on!

Justin Eldridge

with

Mikhail Belkin, Yusu Wang

THE OHIO STATE UNIVERSITY
theory of machine learning
theory of machine learning

classification

clustering
theory of machine learning

classification $\gg$ clustering
theory of machine learning

classification $\gg$ clustering
What is the **correct** clustering?

- In general, there is **no single answer**.
What is the correct clustering?

- In general, there is no single answer.
- But consider a statistical approach...
What is the **correct** clustering?

- In general, there is **no single answer**.
- But consider a **statistical approach**...
What is the **correct** clustering?

- In general, there is **no single answer**.
- But consider a **statistical approach**...
What is the **correct** clustering?

- In general, there is **no single answer**.
- But consider a **statistical approach**...

In the statistical approach, there is often a **natural ground truth clustering**.
In this talk, we develop a statistical theory of graph clustering:

0. We model the data as coming from a graphon.
1. We define the clusters of a graphon.
2. We develop a notion of convergence to the graphon’s clusters.
3. We provide a clustering algorithm which converges to the graphon’s clusters.
In this talk, we develop a statistical theory of graph clustering:

0. We model the data as coming from a graphon.
1. We define the clusters of a graphon.
2. We develop a notion of convergence to the graphon’s clusters.
3. We provide a clustering algorithm which converges to the graphon’s clusters.
Background: the stochastic blockmodel.

- Much of existing theory is in the **stochastic blockmodel**.
- This is a **model** for generating random graphs.
- Each node belongs to one of **$k$ blocks**, or **communities**.
- Edge probabilities parameterized by symmetric $k \times k$ matrix $P$:
  - Prob. of edge **within** community $i$ given by $P_{ii}$.
  - Prob. of edge **between** communities $i$ and $j$ given by $P_{ij}$.
- Example: 2-block model.
  - Social network of **girls** and **boys** at a school.
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
2. Sample edges with probability according to $P$. 

![Diagram showing sampling from a blockmodel]
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
2. Sample edges with probability according to $P$.

Add edge with probability $P$. 
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
2. Sample edges with probability according to $P$.

Add edge with probability $P$. 

\[ \left( \begin{array}{cccc}
\text{Community 1} & \text{Community 2} & \text{Community 3} & \text{Community 4} \\
\text{Community 2} & \text{Community 1} & \text{Community 3} & \text{Community 4} \\
\text{Community 3} & \text{Community 2} & \text{Community 1} & \text{Community 4} \\
\text{Community 4} & \text{Community 3} & \text{Community 2} & \text{Community 1} \\
\end{array} \right) \]
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
2. Sample edges with probability according to $P$.

Add edge with probability $P$. 
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
2. Sample edges with probability according to $P$.

Add edge with probability $P$. 
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
2. Sample edges with probability according to $P$.

Add edge with probability $P$. 
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
2. Sample edges with probability according to $P$.

Add edge with probability $P$. 
Sampling from a blockmodel.

We can generate a random graph with \( n \) nodes from \( P \) as follows...

1. Sample communities uniformly with replacement.
2. Sample edges with probability according to \( P \).

Repeat for all pairs of nodes.
Sampling from a blockmodel.

We can generate a random graph with $n$ nodes from $P$ as follows...

1. Sample communities uniformly with replacement.
2. Sample edges with probability according to $P$.
3. Forget community labels.
Equivalent parameterizations.

Permuting the rows/columns of $P$ does not change graph distribution.
Clustering theory in the stochastic blockmodel.

1. **Define the clusters of the blockmodel.**
   - The communities used to define the blockmodel.

2. **Develop a notion of convergence to the communities.**
   - Recover community labels exactly as $n \rightarrow \infty$.

3. **Construct consistent blockmodel clustering algorithms.**
   - Spectral methods, such as (McSherry, 2001).
Problem: Many real-world networks not well-fit by blockmodel.

- Large networks (Facebook, LinkedIn, etc.) are complicated.
- The 2-block model is very simple.
Problem: Many real-world networks not well-fit by blockmodel.

- Large networks (Facebook, LinkedIn, etc.) are complicated.
- The 2-blockmodel is very simple.
- **Solution:** Increase number of parameters, i.e., communities...
Problem: Many real-world networks not well-fit by blockmodel.

- Large networks (Facebook, LinkedIn, etc.) are complicated.
- The 2-blockmodel is very simple.
- Solution: Increase number of parameters, i.e., communities...
**Problem:** Many real-world networks not well-fit by blockmodel.

- Large networks (Facebook, LinkedIn, etc.) are complicated.
- The 2-blockmodel is very simple.

**Solution:** Increase number of parameters, i.e., communities...
**Problem:** Many real-world networks not well-fit by blockmodel.

- Large networks (Facebook, LinkedIn, etc.) are complicated.
- The 2-blockmodel is very simple.

**Solution:** Increase number of parameters, i.e., communities...
Problem: Many real-world networks not well-fit by blockmodel.

- Large networks (Facebook, LinkedIn, etc.) are complicated.
- The 2-blockmodel is very simple.
- Solution: Increase number of parameters, i.e., communities...
The limit of a blockmodel is...

$$\lim_{k \to \infty} \left( \begin{array}{c}
\begin{array}{cccccccc}
\text{Block 1} & \text{Block 2} & \text{Block 3} & \text{Block 4} & \text{Block 5} & \text{Block 6} & \text{Block 7} & \text{Block 8}
\end{array}
\end{array} \right), \left( \begin{array}{c}
\end{array} \right), \left( \begin{array}{c}
\end{array} \right), \left( \begin{array}{c}
\end{array} \right), \ldots$$
The limit of a blockmodel is...

$$\lim_{k \to \infty} \left( \begin{array}{ccc} \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots \end{array} \right), \ldots$$

$$= \ldots$$

...a graphon!

symmetric, measurable

$$W : [0, 1]^2 \to [0, 1]$$
The limit of a blockmodel is...

\[ \lim_{k \to \infty} \begin{pmatrix}
\begin{array}{ccc}
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot 
\end{array}
\end{pmatrix}, \begin{pmatrix}
\begin{array}{ccc}
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot 
\end{array}
\end{pmatrix}, \begin{pmatrix}
\begin{array}{ccc}
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot 
\end{array}
\end{pmatrix}, \begin{pmatrix}
\begin{array}{ccc}
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot 
\end{array}
\end{pmatrix}, \ldots \]

= 

...a graphon!

symmetric, measurable

\[ W : [0, 1]^2 \to [0, 1] \]

\(\dagger\) Convergence in so-called cut metric, (Lovász, 2012).
Interpretation: The adjacency of an infinite weighted graph.
Interpretation: The adjacency of an infinite weighted graph.

Graphon “nodes” are points $x, y \in [0, 1]$. 
**Interpretation:** The adjacency of an infinite weighted graph.

\[ W(x, y) \] is the weight of the “edge” \((x, y)\).
Sampling a graph from $\mathcal{W}$.

Graphon sampling is analogous to sampling from a blockmodel.
Sampling a graph from $\mathcal{W}$.

First, sample $n$ graphon nodes, i.e., points from Unif[0, 1].
Sampling a graph from $\mathcal{W}$.

First, sample $n$ graphon nodes, i.e., points from $\text{Unif}[0, 1]$. 
Sampling a graph from $\mathcal{W}$.

First, sample $n$ graphon nodes, i.e., points from $\text{Unif}[0, 1]$. 
Sampling a graph from $W$.

First, sample $n$ graphon nodes, i.e., points from $\text{Unif}[0, 1]$.
Sampling a graph from $W$.

First, sample $n$ graphon nodes, i.e., points from $\text{Unif}[0, 1]$. 
Sampling a graph from $\mathcal{W}$.

First, sample $n$ graphon nodes, i.e., points from Unif[0, 1].
Sampling a graph from $W$.

Include edge $(x_1, x_5)$ with probability $W(x_1, x_5)$. 
Sampling a graph from $\mathcal{W}$.

By chance, edge $(x_1, x_5)$ is included.
Sampling a graph from $W$.

Include edge $(x_3, x_6)$ with probability $W(x_3, x_6)$. 
Sampling a graph from $W$.

By chance, edge $(x_3, x_6)$ is omitted.
Sampling a graph from $\mathcal{W}$.

Repeat for all possible edges.
Sampling a graph from $W$.  

Forget node labels, obtaining undirected & unweighted graph.
Sampled graphs converge to the graphon they were sampled from.
Sampled graphs converge to the graphon they were sampled from.
Sampled graphs converge to the graphon they were sampled from.
Sampled graphs converge to the graphon they were sampled from.
Sampled graphs converge to the graphon they were sampled from.
Sampled graphs converge to the graphon they were sampled from.
Sampled graphs converge to the graphon they were sampled from.
A graphon $W$ defines a very rich distribution on graphs.

- Better models real-world data (Hoff, 2002).
- Subsumes many models, e.g., blockmodel:
A graphon $W$ defines a very rich distribution on graphs.

- Better models real-world data (Hoff, 2002).
- Subsumes many models, e.g., blockmodel:

\[
\begin{pmatrix}
    p_1 & q \\
    q & p_2
\end{pmatrix}
\]

Warning! Graphons can be much more complex than blockmodels.

- Present several unique and subtle technical issues.
Issue 1: A graphon node or edge is not meaningful by itself.

\[
\lim_{k \to \infty} \left( \begin{array}{cccc}
& & & \\
& & & \\
& & & \\
& & & \\
\end{array} \right), \left( \begin{array}{cccc}
& & & \\
& & & \\
& & & \\
& & & \\
\end{array} \right), \left( \begin{array}{cccc}
& & & \\
& & & \\
& & & \\
& & & \\
\end{array} \right), \left( \begin{array}{cccc}
& & & \\
& & & \\
& & & \\
& & & \\
\end{array} \right), \ldots
\]
Issue 1: A graphon node or edge is not meaningful by itself.

\[ \lim_{k \to \infty} \sum \text{graphs} \]

In a careful approach:

- Do not reference single nodes/edges in a graphon.
- Only deal with equivalence classes of sets of nodes modulo null sets.

In what follows, we largely ignore the issue in the interest of time and simplicity; see paper for details.
Recall: \( P_1 \) and \( P_2 \) define the same stochastic blockmodel if they are equivalent up to relabeling.

\[
\begin{pmatrix}
0 & 1 \\
1 & 0
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & 1 \\
1 & 0
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & 1 \\
1 & 0
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix}
\]

Issue 2: Similarly, \( W_1 \) and \( W_2 \) define the same graphon model \( \iff \) they are equivalent up to relabeling, (Lovász, 2012).
Issue 2: A graphon relabeling can be very complex.

- A relabeling is a map $\varphi : [0, 1] \rightarrow [0, 1]$.
- $\varphi$ must be “measure preserving”.
  - Only in one direction: preimage.
  - Can map a null set to a set of full measure!
- Does not need to be a bijection. Far from it!
Issue 2: A graphon relabeling can be very complex.

- A relabeling is a map \( \varphi : [0, 1] \rightarrow [0, 1] \).
- \( \varphi \) must be “measure preserving”.
  - Only in one direction: preimage.
  - Can map a null set to a set of full measure.

There is usually no **canonical** way to label a graphon.

- For presentation, we will use a “nice” labeling of “nice” graphons; i.e., piecewise constant.
- But our definitions will make sense for any labeling of any graphon; i.e., arbitrarily-complex measurable function.
A statistical theory of graphon clustering.

In this talk...

0. We model the data as coming from a graphon.

We give answers to the following:

1. What are the clusters of a graphon?
2. How do we define convergence to the graphon’s clusters?
   ▶ I.e., statistical consistency.
3. Which clustering algorithms are consistent?
What are the clusters of a graphon?

We interpret the graphon as the adjacency of an infinite weighted graph.
What are the **clusters** of a graphon?

Each link in this depiction corresponds to a region of the graphon.
What are the clusters of a graphon?

Each link in this depiction corresponds to a region of the graphon.
What are the clusters of a graphon?

Each link in this depiction corresponds to a region of the graphon.
What are the clusters of a graphon?

Each link in this depiction corresponds to a region of the graphon.
What are the clusters of a graphon?

Each link in this depiction corresponds to a region of the graphon.
What are the clusters of a graphon?

- We define clusters to be connected components.
- Use generalization of graph connectivity, extends (Janson, 2008).
- Key: Insensitive to null sets, e.g., single edges.
What are the clusters of a graphon?

- In fact, we can speak of the clusters at various levels.
- Intuitively: three clusters (connected components) at level $\lambda_3$.
- Any pair ($\bullet$, $\bullet$) are in same cluster at $\lambda_3$. Same for ($\bullet$, $\bullet$) & ($\bullet$, $\bullet$).
What are the **clusters** of a graphon?

- In fact, we can speak of the clusters at various levels.
- Intuitively: three clusters (connected components) at level $\lambda_3$.
- Any pair ($\bullet$, $\circ$) are in same cluster at $\lambda_3$. Same for ($\bullet$, $\bullet$) & ($\circ$, $\circ$).
- Naturally encoded as function $M(\bullet, \circ) = M(\bullet, \bullet) = M(\circ, \circ) = \lambda_3$.
What are the clusters of a graphon?

- In fact, we can speak of the clusters at various levels.
- Intuitively: red and blue clusters merge at level $\lambda_2$.
- Any pair ($\circ, \bullet$) are in same cluster at $\lambda_2$.
- Naturally encoded as $M(\circ, \bullet) = M(\bullet, \circ) = \lambda_2$. 

![Diagram showing clusters and levels $\lambda_1$, $\lambda_2$, $\lambda_3$.]
What are the clusters of a graphon?

- In fact, we can speak of the clusters at various levels.
- All clusters merge at level $\lambda_1$.
- Encoded as $M(\bigcirc, \bigcirc) = M(\bigotimes, \bigotimes) = \lambda_1$. 
We call $M$ the **mergeon**.
We call $M$ the **mergeon**.

- $M(x, y)$ encodes the first level at which $x$ & $y$ are in same cluster.
- As such, $M$ defines the **ground truth** clustering of a graphon.
- **Note**: Mergeon helps deal with subtle technical hurdles.
A mergeon has hierarchical structure. Clusters from higher levels nest within clusters from lower levels.

We call this structure the graphon cluster tree.
If graphons $W_1$ and $W_2$ are the same up to relabeling, then their mergeons and cluster trees are the same up to relabeling.

Surprisingly non-trivial to show.
A statistical theory of graphon clustering.

1. What is the ground truth clustering of a graphon?
   ▶ The mergeon, or, equivalently, the graphon cluster tree.

2. How do we define convergence?
A statistical theory of graphon clustering.

1. What is the ground truth clustering of a graphon?
   ▶ The mergeon, or, equivalently, the graphon cluster tree.

2. How do we define convergence?
A statistical theory of graphon clustering.

1. What is the ground truth clustering of a graphon?
   ▶ The mergeon, or, equivalently, the graphon cluster tree.

2. How do we define convergence?

   ![Graph and cluster tree diagram]

   as $n \rightarrow \infty$
The merge distortion

How “close” are $C$ and $C'$. 

Diagram showing the merge distortion.
Intuitively, corresponding pairs of nodes should merge at around the same height in each tree.
The merge distortion

$M(\bullet, \bullet)$

Merge heights are encoded in the mergeon.
The merge distortion

Merge heights are encoded in the mergeon.
The merge distortion

\[ |M(\circ, \bullet) - M'(\circ, \bullet)| \] is the difference in merge height of \( \circ, \bullet \).
The merge distortion

We introduce the merge distortion $d(C, C')$: the maximum difference in merge height over all pairs, i.e,

$$d(C, C') = \max_{M, M'} |M(\bullet, \circ) - M'(\bullet, \circ)|.$$
Convergence in merge distortion

We say $\hat{C}_n$ converges in merge distortion to $C$ if $d(C, \hat{C}_n) \to 0$ as $n \to \infty$.

Definition
An algorithm is consistent if its output converges in merge distortion to the graphon cluster tree w.h.p. as $n \to \infty$.

Consistency:

$\implies$ disjoint clusters are separated as $n \to \infty$.

$\implies$ strong consistency in the blockmodel.
A statistical theory of graphon clustering.

1. What is the ground truth clustering of a graphon?
   ▶ The mergeon, or, equivalently, the graphon cluster tree.

2. How do we define convergence/consistency?
   ▶ Convergence in merge distortion using the mergeon.

3. Which clustering algorithms are consistent?
Estimating edge probabilities.

Suppose we sample a graph from this graphon.
Estimating edge probabilities.

Edges within communities have probability $p$; edges across communities have probability $q$. 
Estimating edge probabilities.

If we knew these edge probabilities we could recover the correct clusters.
Estimating edge probabilities.

But the edge probabilities are unknown and the presence/absence of an edge \((i,j)\) tells us little about its probability, \(P_{ij}\).
Estimating edge probabilities.

But the edge probabilities are unknown and the presence/absence of an edge \((i, j)\) tells us little about its probability, \(P_{ij}\).

Idea: Compute estimate \(\hat{P}\) of edge probabilities from a single graph.
Theorem

If $\|P - \hat{P}\|_{max} \to 0$ in probability as $n \to \infty$, then single linkage clustering using $\hat{P}$ as the input similarity matrix is a consistent clustering method.
Theorem
If $\|P - \hat{P}\|_{\text{max}} \to 0$ in probability as $n \to \infty$, then single linkage clustering using $\hat{P}$ as the input similarity matrix is a consistent clustering method.

- There are many recent graphon & edge probability estimators.
- But all consistency results are in mean squared error.
- This is too weak. Need consistency in max-norm.
- We modify and analyze the neighborhood smoothing method of (Zhang et al., 2015) to obtain consistency in max-norm.
Given this graph...
Given this graph... estimate $P_{ij}$. 

Neighborhood smoothing
Build a neighborhood $N_i$ of nodes with similar connectivity to that of $i$. 
Neighborhood smoothing

- Average number edges from node in neighborhood $N_i$ to $j$.
- Estimated edge probability: $\hat{P}_{ij} = \frac{2}{6} = \frac{1}{3}$. 
Consistency of neighborhood smoothing.

Theorem

Our modified neighborhood smoothing edge probability estimator for $P$ is consistent in max norm.

Corollary

Consistent graphon clustering method:

1. Estimate edge probabilities with our modified neighborhood smoothing approach.
2. Apply single linkage clustering to estimated edge probabilities.
In summary, we develop a statistical theory of graph clustering in the graphon model:

1. We define the clusters of a graphon.
   ▶ The graphon cluster tree/mergeon.

2. We develop a notion of consistency.
   ▶ Convergence in merge distortion.

3. We provide a consistent algorithm.
   ▶ Modified neighborhood smoothing + single linkage.
THANK YOU!
THANK YOU!
Thank you!
THANK YOU

THANK YOU
Graphons, mergeons, and so on!

Justin Eldridge, Mikhail Belkin, Yusu Wang

- Poster #181, tonight’s session.
- Related work for prob. densities: Eldridge Belkin Wang, COLT 2015.
- Thank you to Stefanie Jegelka for helpful comments on presentation.