Graphons, mergeons, and so on!

Justin Eldridge

with

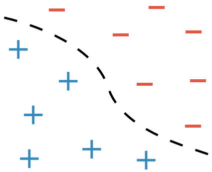
Mikhail Belkin, Yusu Wang

The Ohio State University

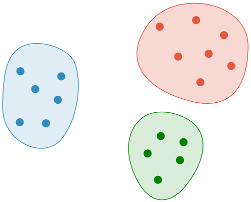
theory of machine learning

theory of machine learning

classification

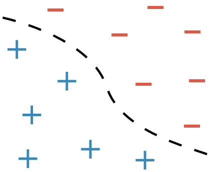


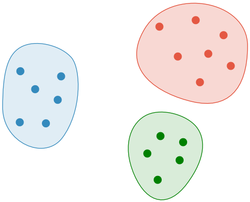
clustering



theory of machine learning

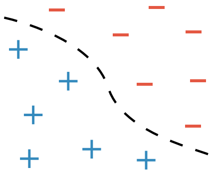
classification $\gg$ clustering

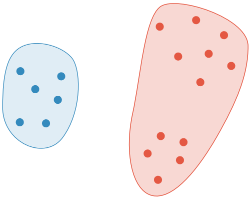




theory of machine learning

classification ≫ clustering



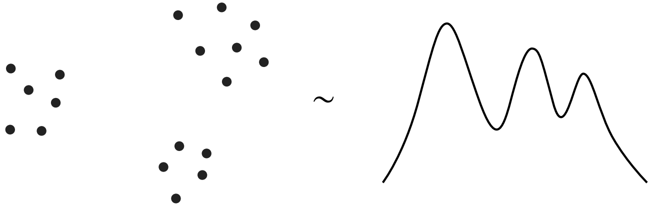


What is the correct clustering?

- In general, there is no single answer.

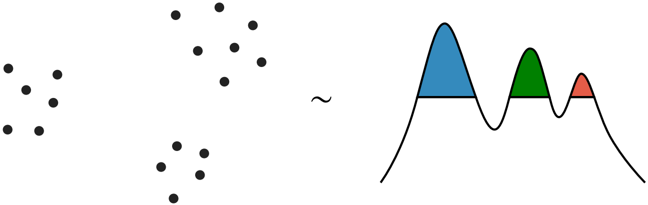
What is the correct clustering?

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



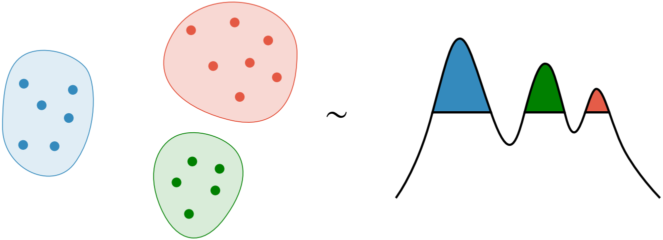
What is the correct clustering?

- 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...



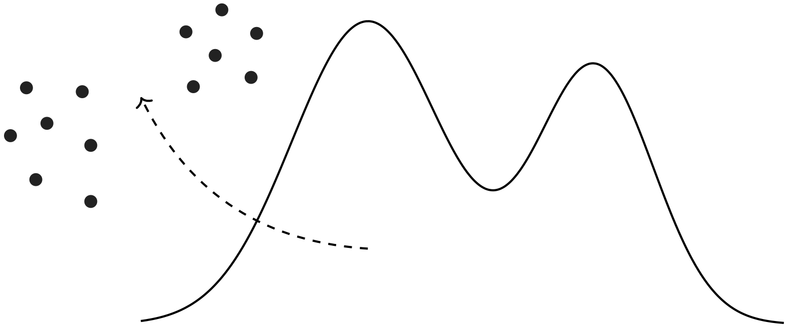
What is the correct clustering?

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

In the statistical approach, there is often a natural ground truth clustering.

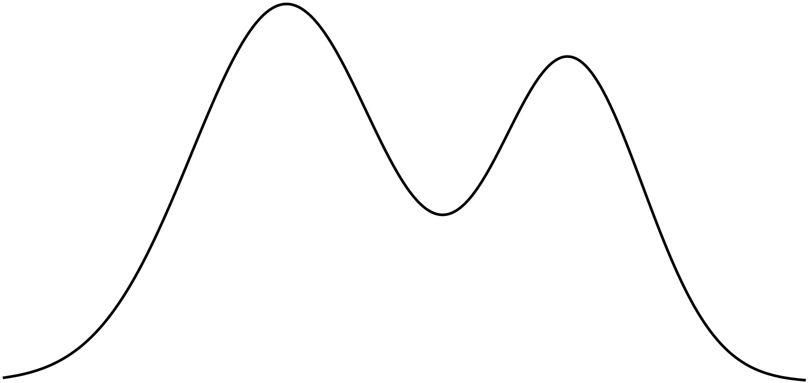
Example: the density model.

0. Model the data as coming from a probability density.



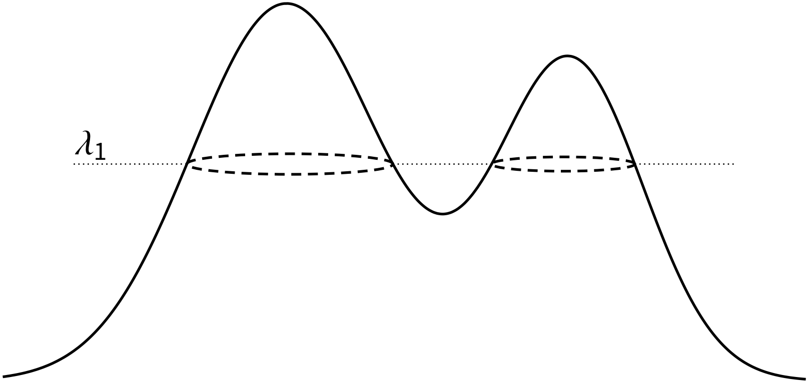
Example: the density model.

1. Define the clusters of the density.
    - Region of high probability.



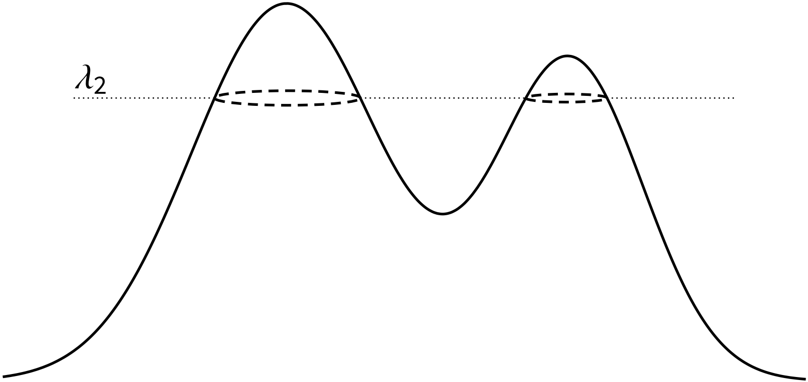
Example: the density model.

1. Define the clusters of the density.
   - Connected component of $\{f \geq \lambda_1\}$?



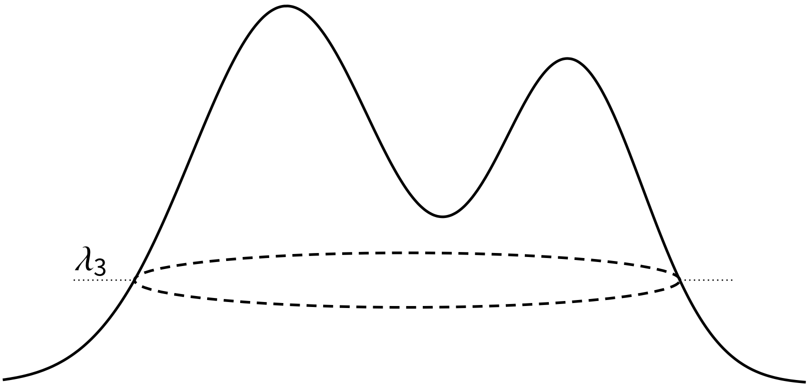
Example: the density model.

1. Define the clusters of the density.
   - Connected component of $\{f \geq \lambda_2\}$?



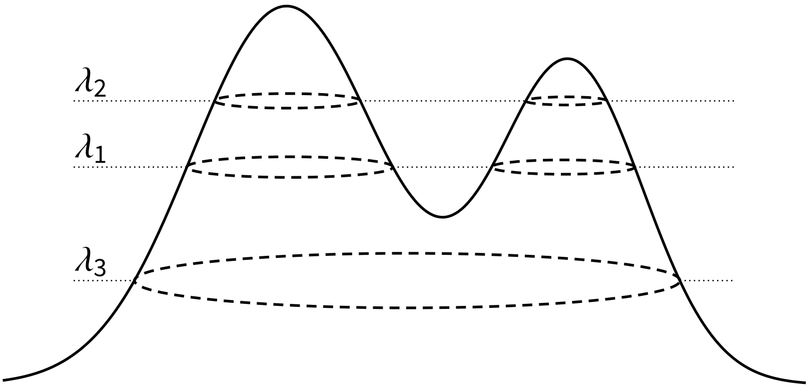
Example: the density model.

1. Define the clusters of the density.
   - Connected component of $\{f \geq \lambda_3\}$?



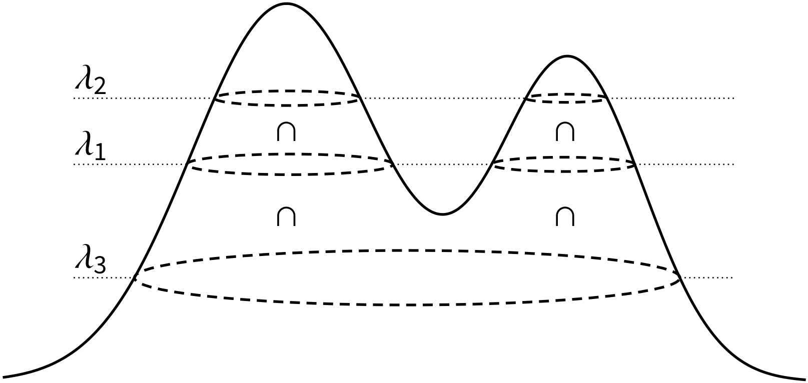
Example: the density model.

1. Define the clusters of the density.
   - Connected component of $\{f \geq \lambda\}$ for any $\lambda > 0$.



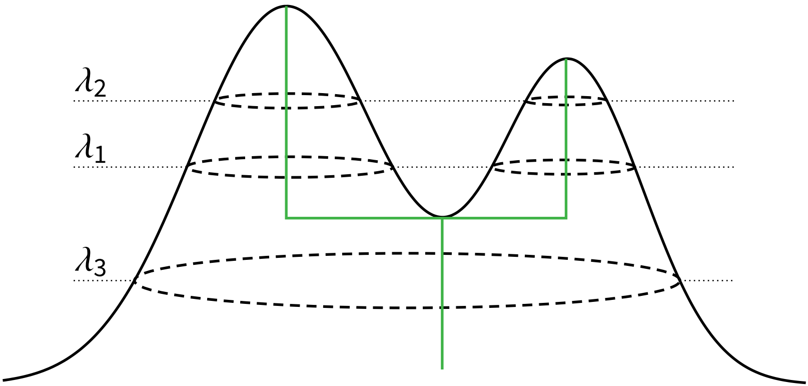
Example: the density model.

1. Define the clusters of the density.
   - Connected component of $\{f \geq \lambda\}$ for any $\lambda > 0$.



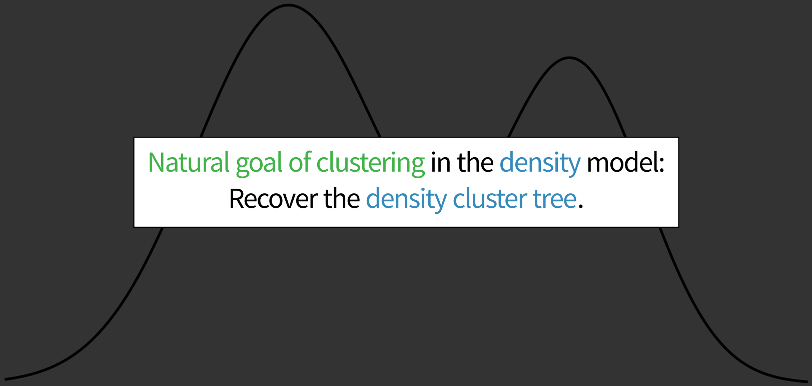
Example: the density model.

1. Define the clusters of the density.
   - Elements of the density cluster tree of f.



Example: the density model.

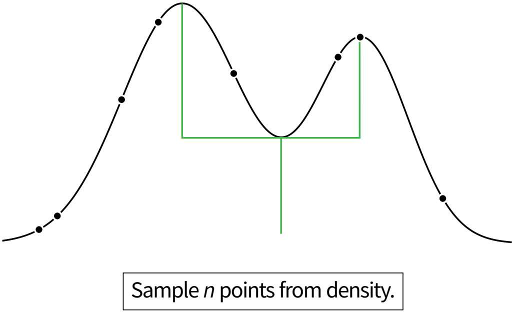
1. Define the clusters of the density.
   - Elements of the density cluster tree of f.



Natural goal of clustering in the density model:
Recover the density cluster tree.

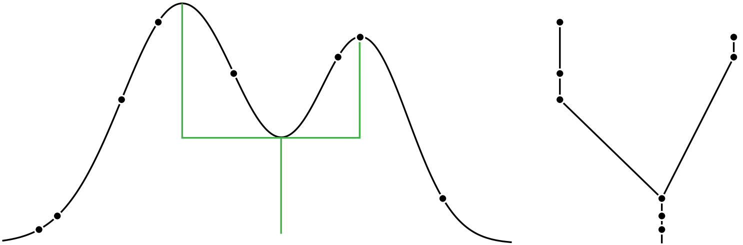
Example: the density model.

2. Develop a notion of convergence to the density cluster tree.



Example: the density model.

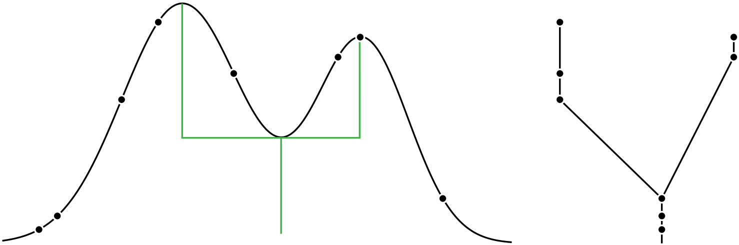
2. Develop a notion of convergence to the density cluster tree.



Apply hierarchical clustering algorithm.

Example: the density model.

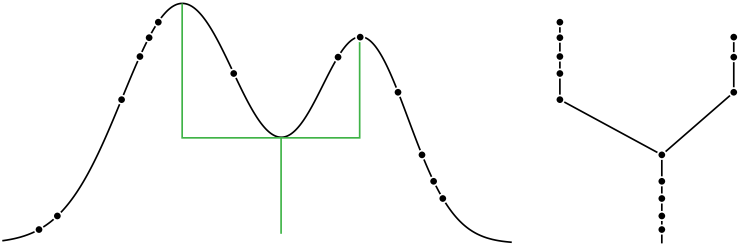
2. Develop a notion of convergence to the density cluster tree.



As $n \to \infty$...

Example: the density model.

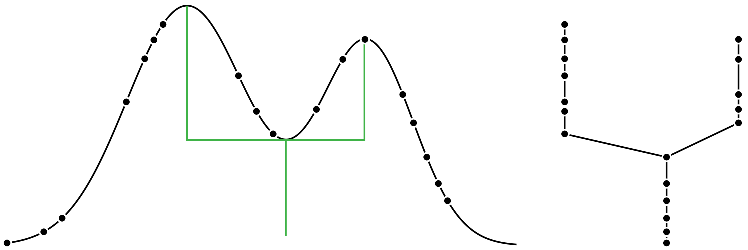
2. Develop a notion of convergence to the density cluster tree.



As $n \to \infty$...

Example: the density model.

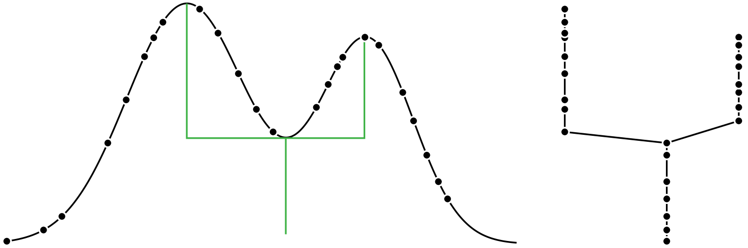
2. Develop a notion of convergence to the density cluster tree.



As $n \to \infty$...

Example: the density model.

2. Develop a notion of convergence to the density cluster tree.



As $n \to \infty$...

Example: the density model.

2. Develop a notion of convergence to the density cluster tree.
    - Weak notion: Hartigan consistency (1981).
        - Clusters disjoint in true tree should be disjoint in clustering.

Example: the density model.

2. Develop a notion of convergence to the density cluster tree.
   - Weak notion: Hartigan consistency (1981).
     - Clusters disjoint in true tree should be disjoint in clustering.

3. Construct consistent density clustering algorithms.

Example: the density model.

2. Develop a notion of convergence to the density cluster tree.
   - Weak notion: Hartigan consistency (1981).
     - Clusters disjoint in true tree should be disjoint in clustering.

3. Construct consistent density clustering algorithms.
   - Hartigan consistent:
     - Robust single linkage (Chaudhuri & Dasgupta, 2010)
     - Tree pruning (Kpotufe & von Luxburg, 2011)

Example: the density model.

2. Develop a notion of convergence to the density cluster tree.
   - Weak notion: Hartigan consistency (1981).
     - Clusters disjoint in true tree should be disjoint in clustering.
   - Strong notion: Merge distortion (**EBW**, 2015).
     - Pairs of points merge around same height in both trees.

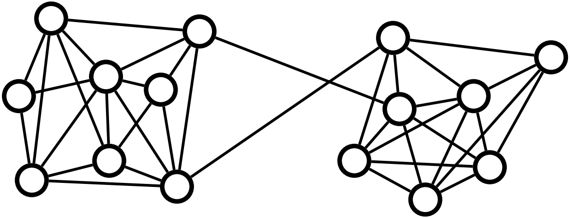
3. Construct consistent density clustering algorithms.
   - Hartigan consistent:
     - Robust single linkage (Chaudhuri & Dasgupta, 2010)
     - Tree pruning (Kpotufe & von Luxburg, 2011)

Example: the density model.

2. Develop a notion of convergence to the density cluster tree.
   - Weak notion: Hartigan consistency (1981).
     - Clusters disjoint in true tree should be disjoint in clustering.
   - Strong notion: Merge distortion (**EBW**, 2015).
     - Pairs of points merge around same height in both trees.

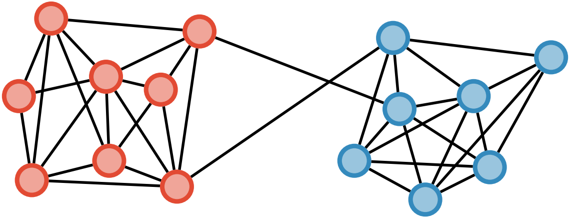
3. Construct consistent density clustering algorithms.
   - Hartigan consistent:
     - Robust single linkage (Chaudhuri & Dasgupta, 2010)
     - Tree pruning (Kpotufe & von Luxburg, 2011)
   - Consistent in merge distortion:
     - (**EBW**, 2015)

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.

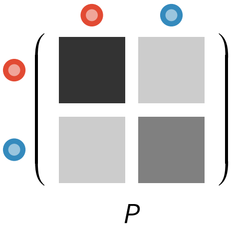
In this talk, we develop a statistical theory of graph clustering:



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.

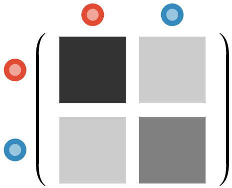
- Much of existing theory is in the stochastic blockmodel.
- This is a model for generating random graphs.
- Each node belongs to one of k blocks, or communities.
- Edge probabilities parameterized by symmetric $k \times k$ matrix P:
  - Prob. of edge within community i given by P_{ii}.
  - Prob. of 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...

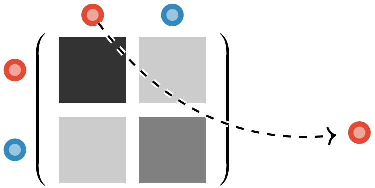
1. Sample communities uniformly with replacement.



Sampling from a blockmodel.

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

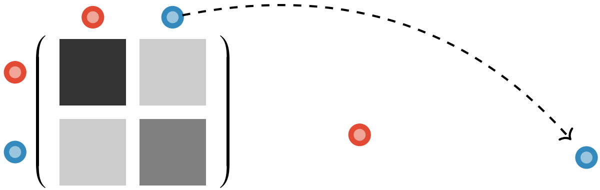
1. Sample communities uniformly with replacement.



Sampling from a blockmodel.

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

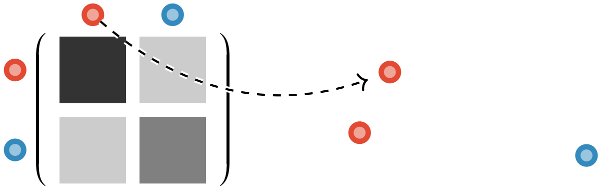
1. Sample communities uniformly with replacement.



Sampling from a blockmodel.

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

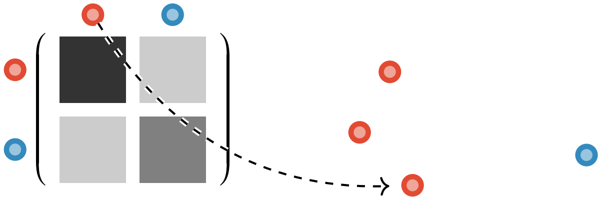
1. Sample communities uniformly with replacement.



Sampling from a blockmodel.

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

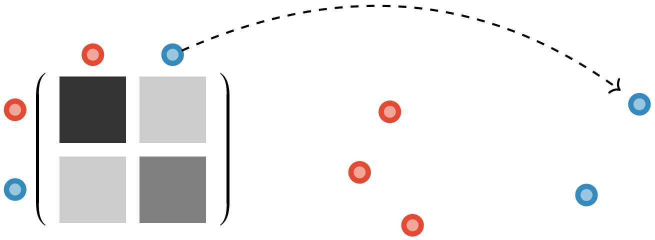
1. Sample communities uniformly with replacement.



Sampling from a blockmodel.

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

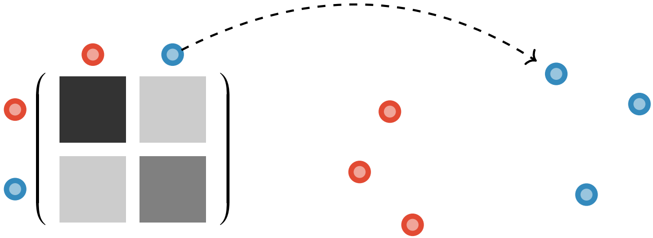
1. Sample communities uniformly with replacement.



Sampling from a blockmodel.

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

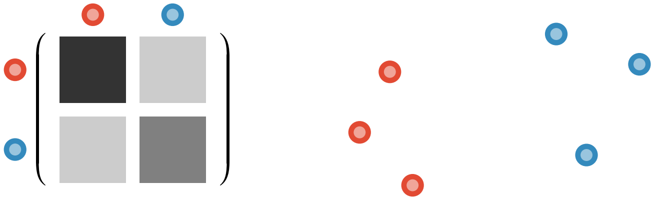
1. Sample communities uniformly with replacement.



Sampling from a blockmodel.

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

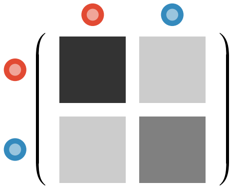
1. Sample communities uniformly with replacement.
2. Sample edges with probability according to P.

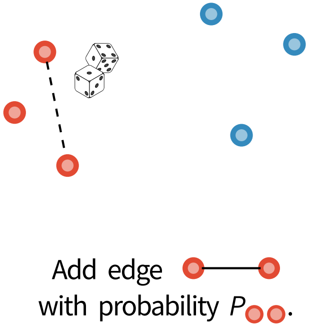


Sampling from a blockmodel.

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.

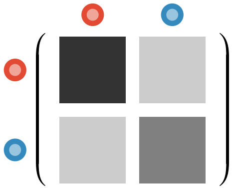


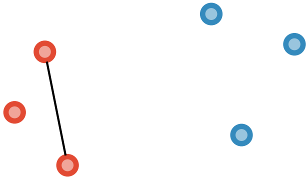


Sampling from a blockmodel.

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.

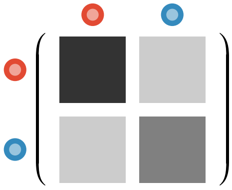


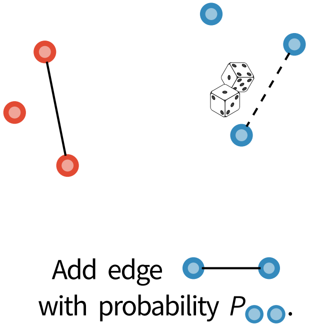


Sampling from a blockmodel.

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.

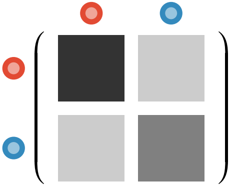


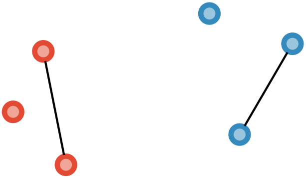


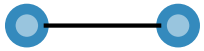
Sampling from a blockmodel.

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.



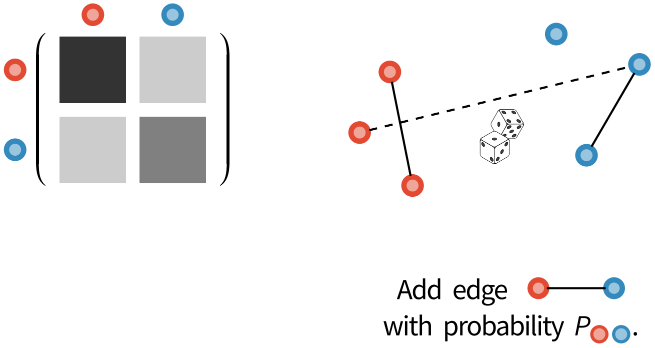


Add edge 
with probability P.

Sampling from a blockmodel.

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

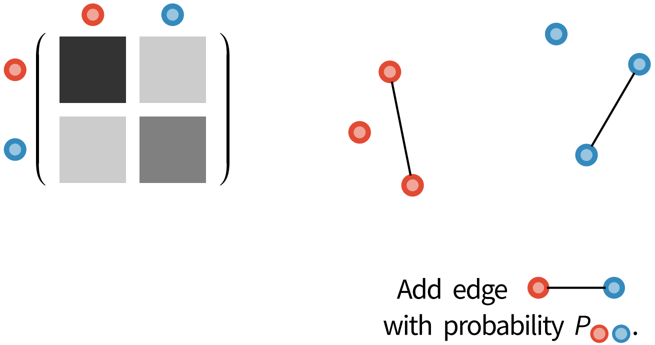
1. Sample communities uniformly with replacement.
2. Sample edges with probability according to P.



Sampling from a blockmodel.

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

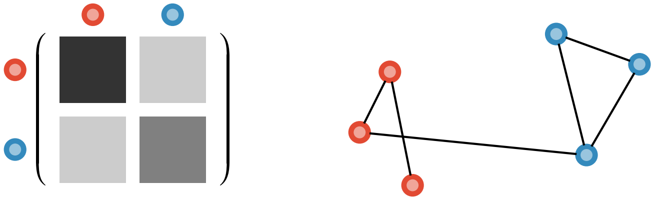
1. Sample communities uniformly with replacement.
2. Sample edges with probability according to P.



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.

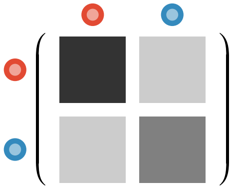


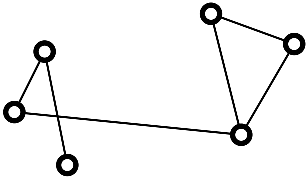
Repeat for all pairs of nodes.

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.





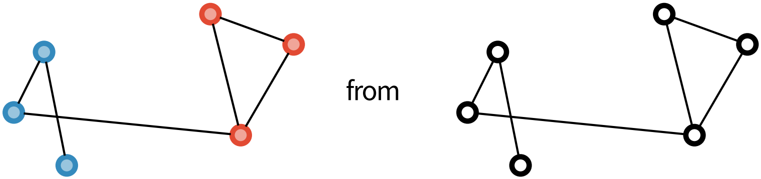
Equivalent parameterizations.

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.

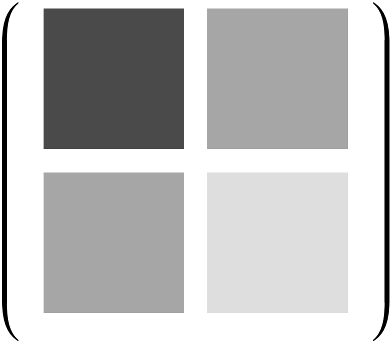
2. Develop a notion of convergence to the communities.
   - Recover community labels exactly as $n \to \infty$.



3. Construct consistent blockmodel clustering algorithms.
   - Spectral methods, such as (McSherry, 2001).

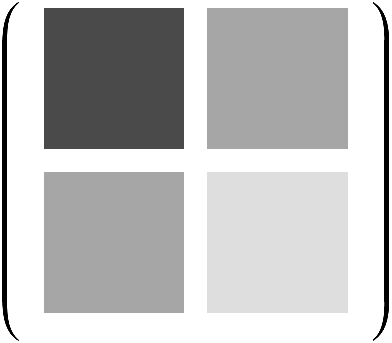
Problem: Many real-world networks not well-fit by blockmodel.

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



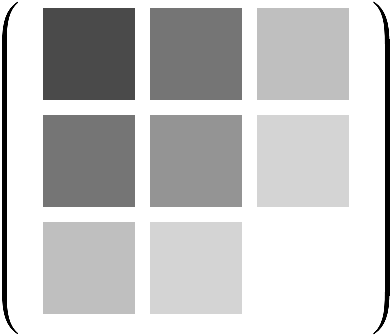
Problem: Many real-world networks not well-fit by blockmodel.

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



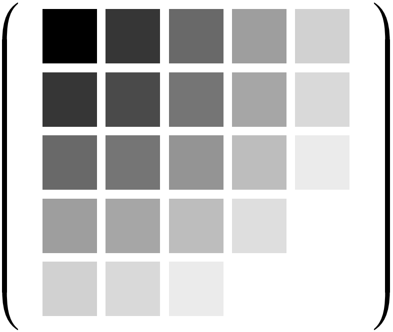
Problem: Many real-world networks not well-fit by blockmodel.

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



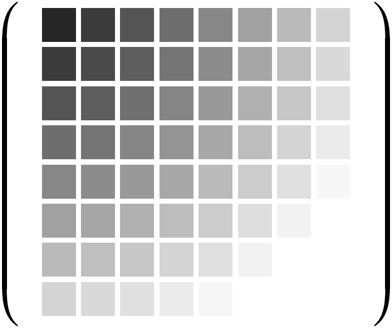
Problem: Many real-world networks not well-fit by blockmodel.

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



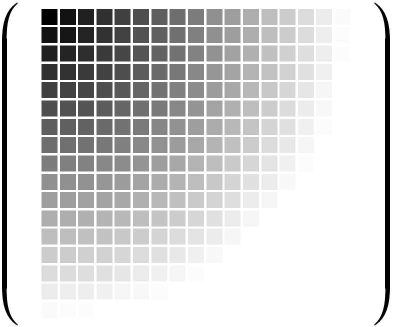
Problem: Many real-world networks not well-fit by blockmodel.

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



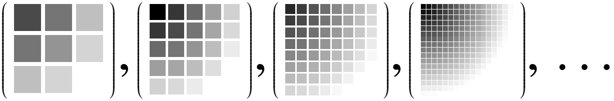
Problem: Many real-world networks not well-fit by blockmodel.

- 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...

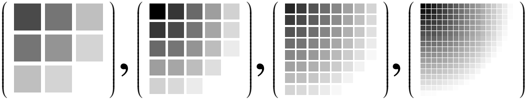
$$\lim_{k\to\infty}$$



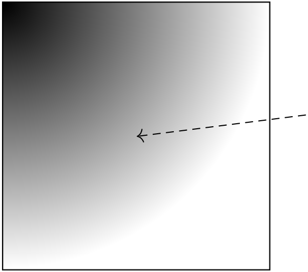
$$=$$

?

The limit of a blockmodel is...

$\lim_{k\to\infty}$  , . . .

=



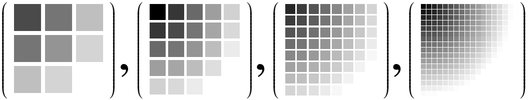
...a graphon!

symmetric,

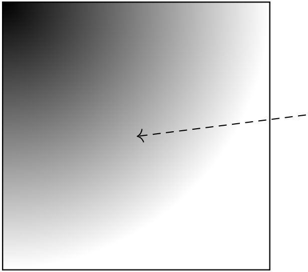
measurable

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

The limit of a blockmodel is...

$\lim_{k\to\infty}^{\dagger}$  , . . .

$=$

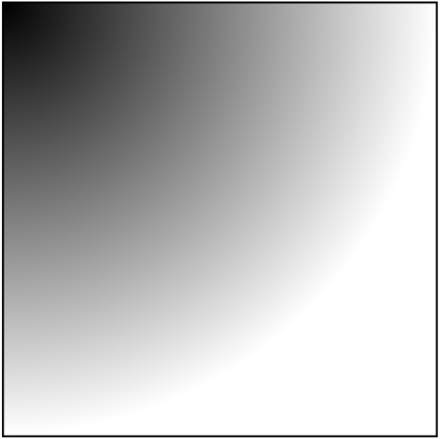


...a graphon!

symmetric,
measurable
$W : [0,1]^2 \to [0,1]$

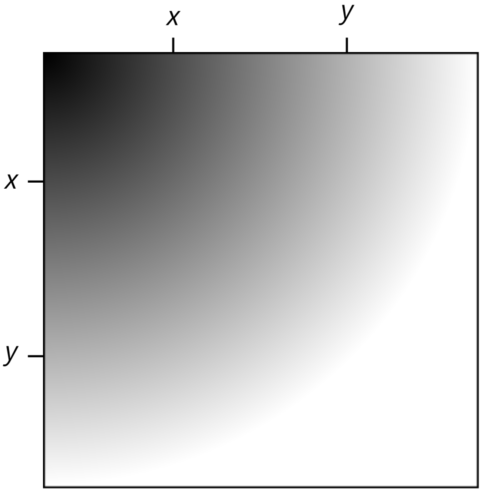
† Convergence in so-called cut metric, (Lovász, 2012).

Interpretation: The adjacency of an infinite weighted graph.



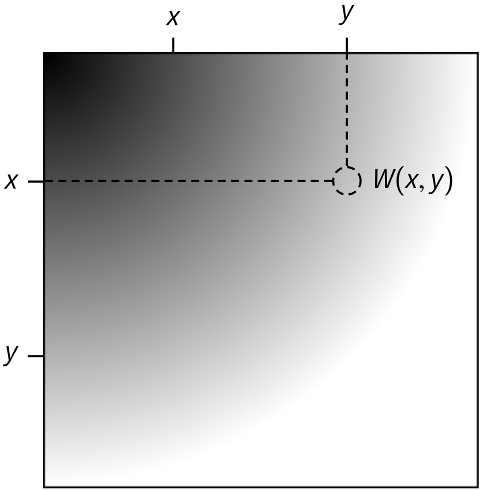
Interpretation: The adjacency of an infinite weighted graph.

Graphon "nodes" are points $x, y \in [0, 1]$.



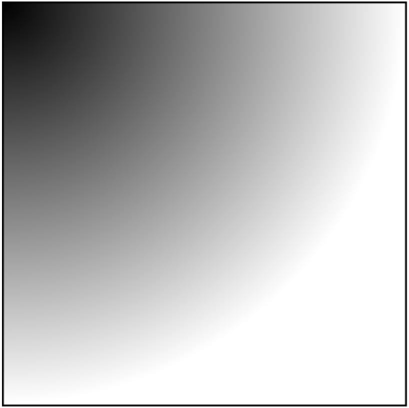
Interpretation: The adjacency of an infinite weighted graph.

$W(x, y)$ is the weight of the "edge" (x, y).



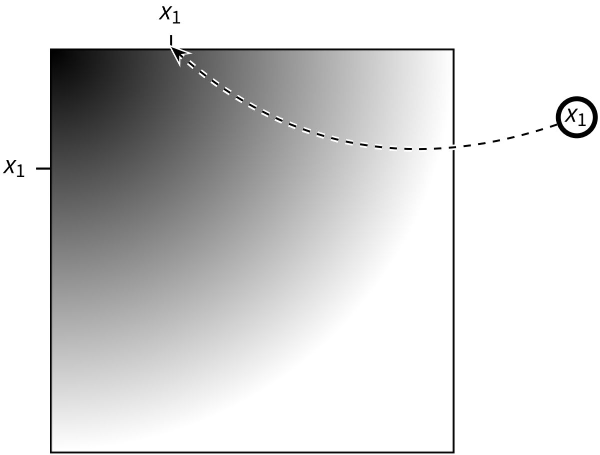
Sampling a graph from W.

Graphon sampling is analogous to sampling from a blockmodel.



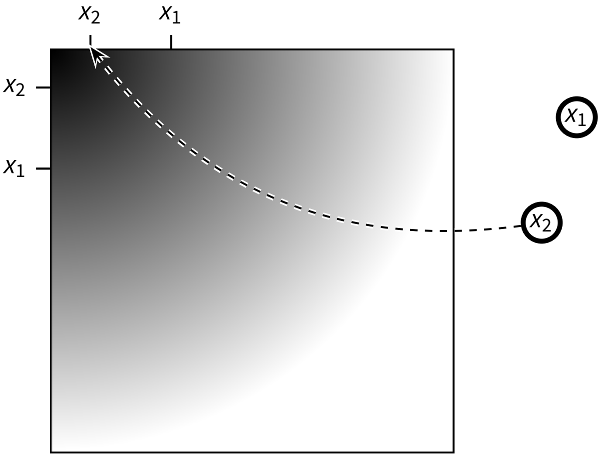
Sampling a graph from W.

First, sample n graphon nodes, i.e., points from Unif[0, 1].



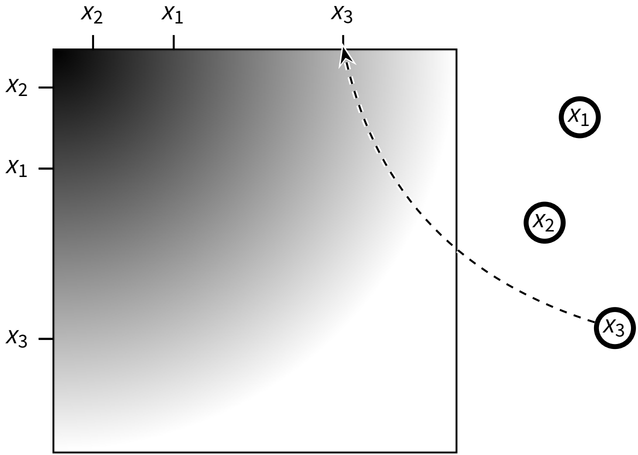
Sampling a graph from W.

First, sample n graphon nodes, i.e., points from Unif[0, 1].



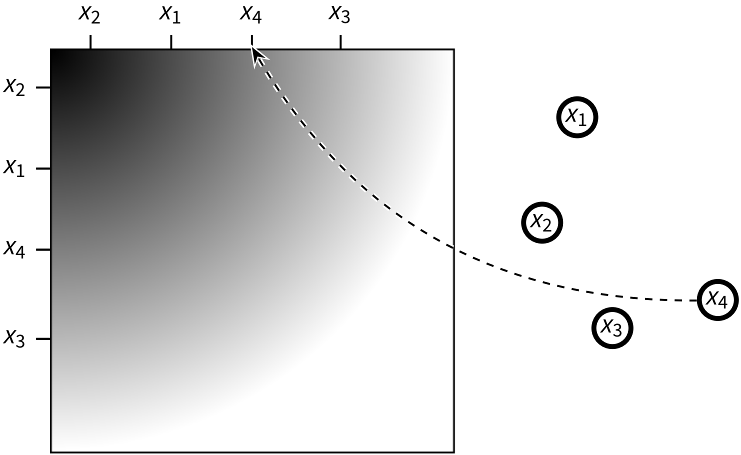
Sampling a graph from W.

First, sample n graphon nodes, i.e., points from Unif[0, 1].



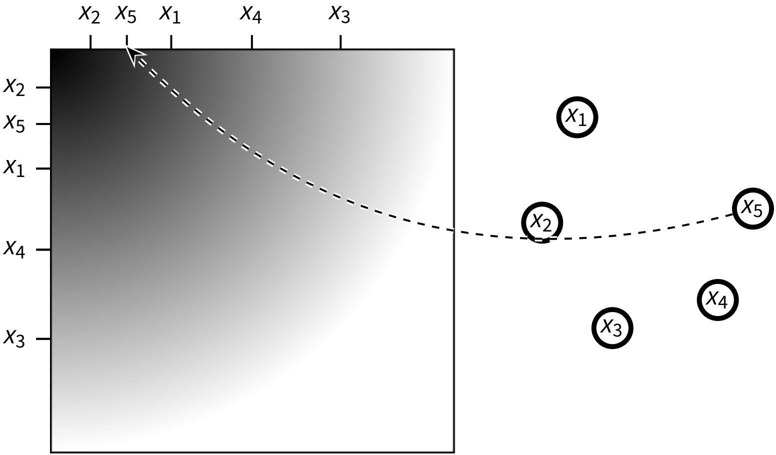
Sampling a graph from W.

First, sample n graphon nodes, i.e., points from Unif[0, 1].



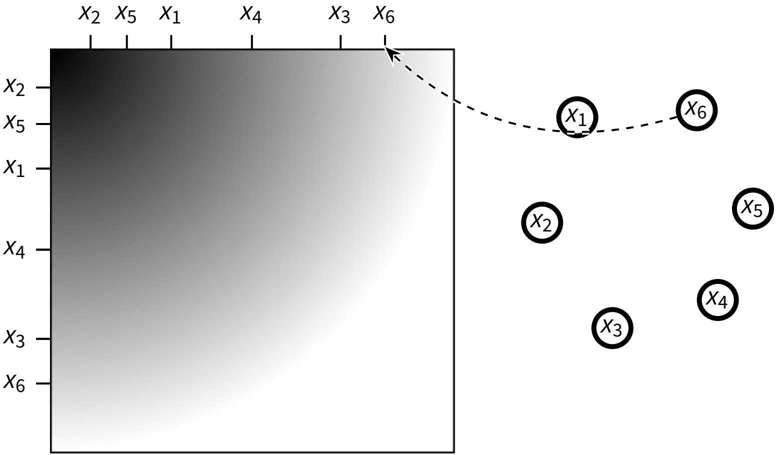
Sampling a graph from W.

First, sample n graphon nodes, i.e., points from Unif[0, 1].



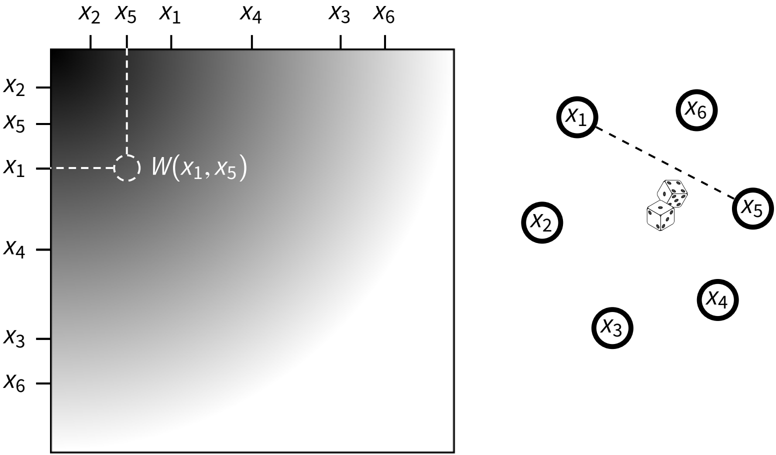
Sampling a graph from W.

First, sample n graphon nodes, i.e., points from Unif[0, 1].



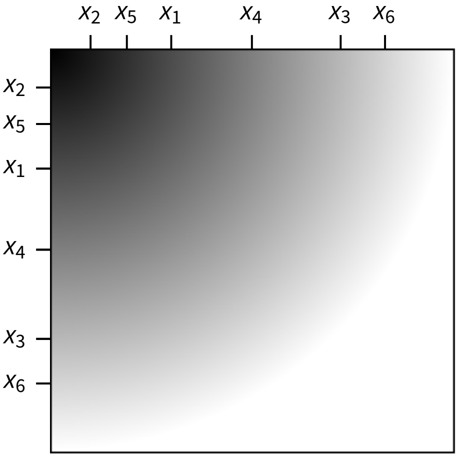
Sampling a graph from W.

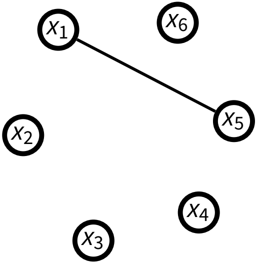
Include edge (x_1, x_5) with probability $W(x_1, x_5)$.



Sampling a graph from W.

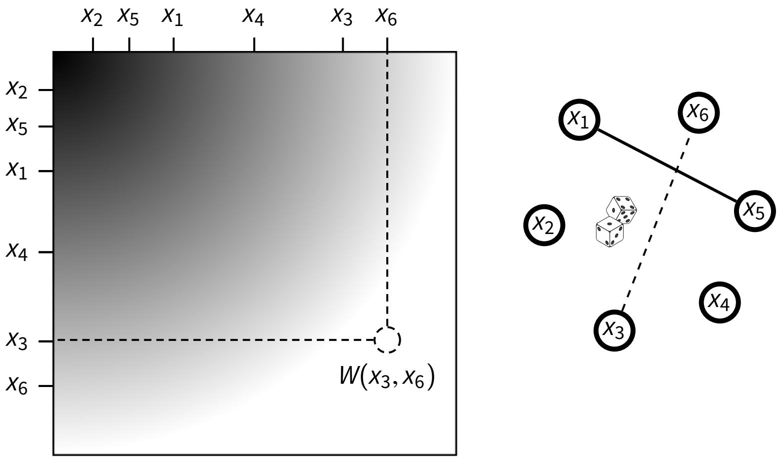
By chance, edge (x_1, x_5) is included.





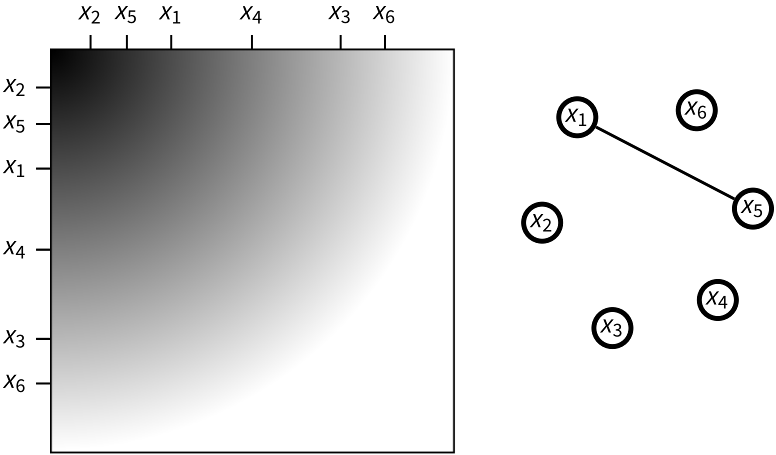
Sampling a graph from W.

Include edge (x_3, x_6) with probability $W(x_3, x_6)$.



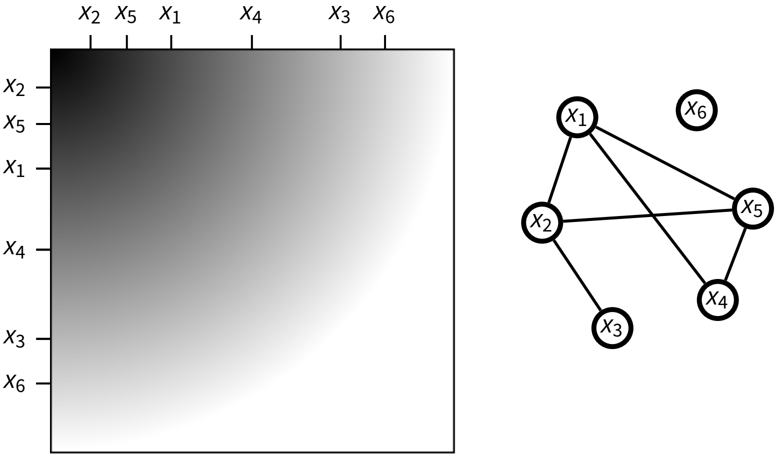
Sampling a graph from W.

By chance, edge (x_3, x_6) is omitted.



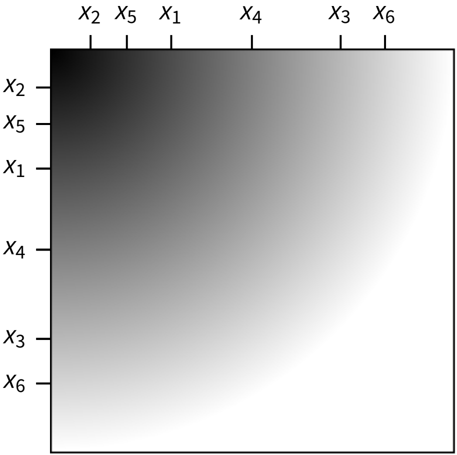
Sampling a graph from W.

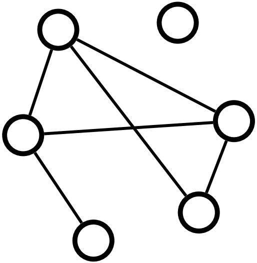
Repeat for all possible edges.



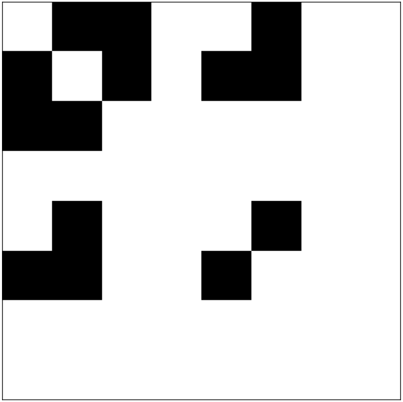
Sampling a graph from W.

Forget node labels, obtaining undirected & unweighted graph.

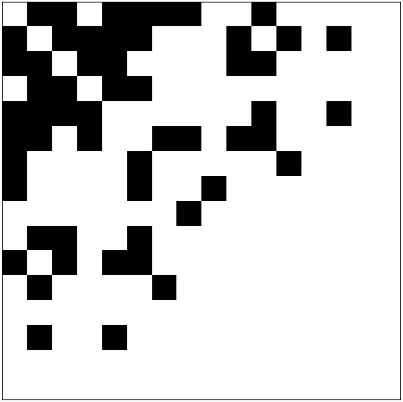




Sampled graphs converge to the graphon they were sampled from.



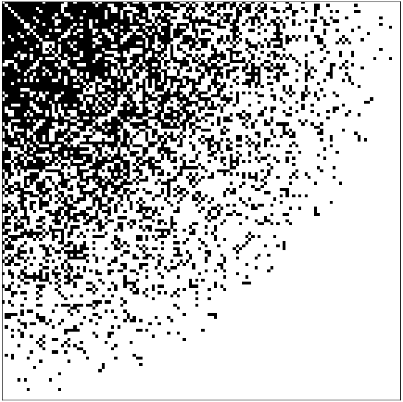
Sampled graphs converge to the graphon they were sampled from.



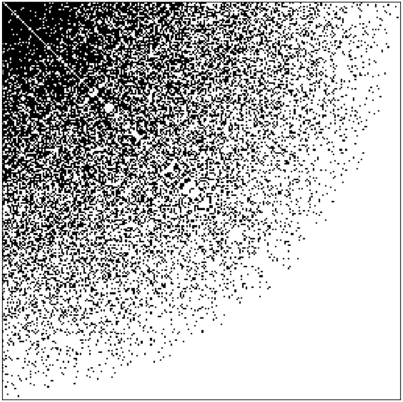
Sampled graphs converge to the graphon they were sampled from.

Sampled graphs converge to the graphon they were sampled from.

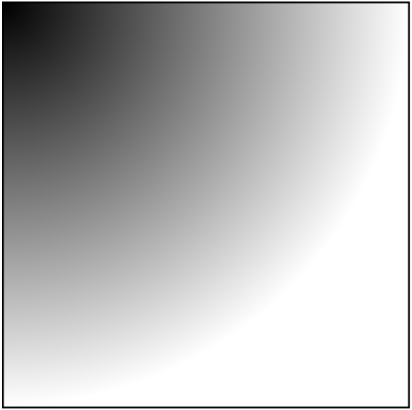
Sampled graphs converge to the graphon they were sampled from.



Sampled graphs converge to the graphon they were sampled from.

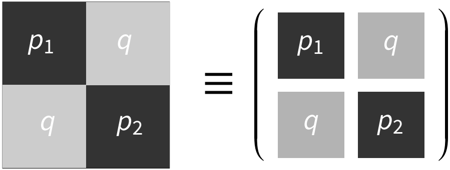


Sampled graphs converge to the graphon they were sampled from.



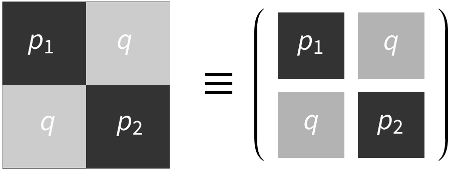
A graphon W defines a very rich distribution on graphs.

- 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:

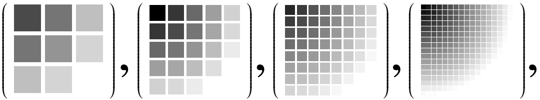


Warning! Graphons can be much more complex than blockmodels.

- Present several unique and subtle technical issues.

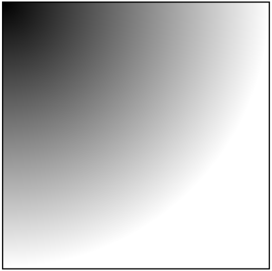
Issue 1: A graphon node or edge is not meaningful by itself.

$\lim_{k\to\infty}$

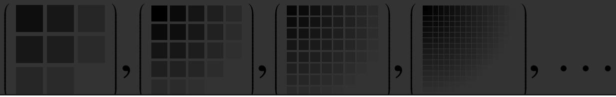


$, \ldots$

$=$



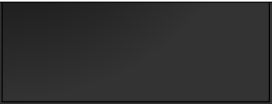
Issue 1: A graphon node or edge is not meaningful by itself.

$\lim_{k\to\infty}$ 

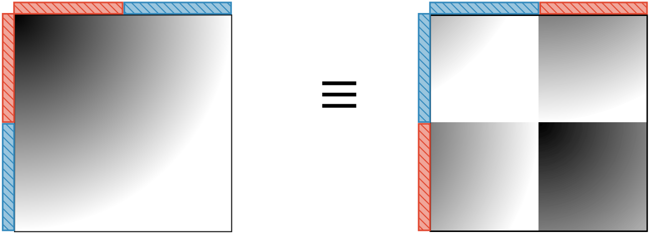
In a careful approach:

- 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.



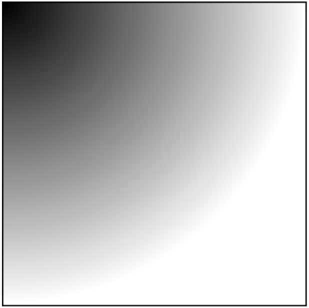
Recall: P_1 and P_2 define the same stochastic blockmodel if they are equivalent up to relabeling.



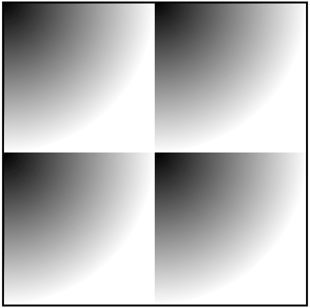
Issue 2: Similarly, W_1 and W_2 define the same graphon model $\iff$ they are equivalent up to relabeling, (Lovász, 2012).

Issue 2: A graphon relabeling can be very complex.

- A relabeling is a map $\varphi : [0, 1] \to [0, 1]$.
- φ 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!



$\equiv$

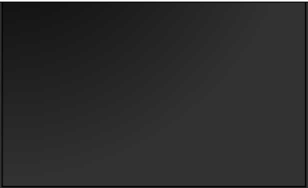


Issue 2: A graphon relabeling can be very complex.

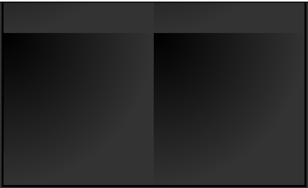
- A relabeling is a map $\varphi : [0,1] \to [0,1]$.
- φ must be "measure preserving".
  - Only in one direction: preimage.
  - Can map a null set to a set of full measure!

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.



$\equiv$



A statistical theory of graphon clustering.

In this talk...

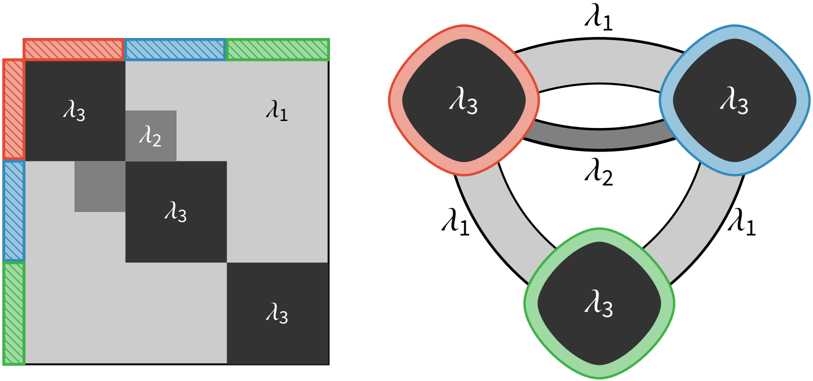
0. We model the data as coming from a graphon.

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?

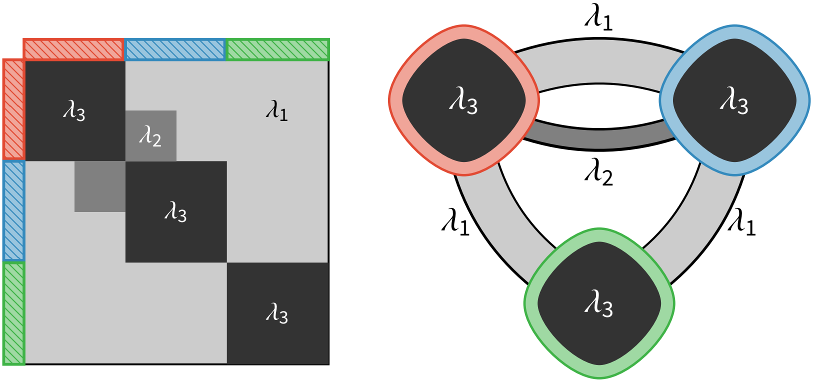
What are the clusters of a graphon?

We interpret the graphon as the adjacency of an infinite weighted graph.



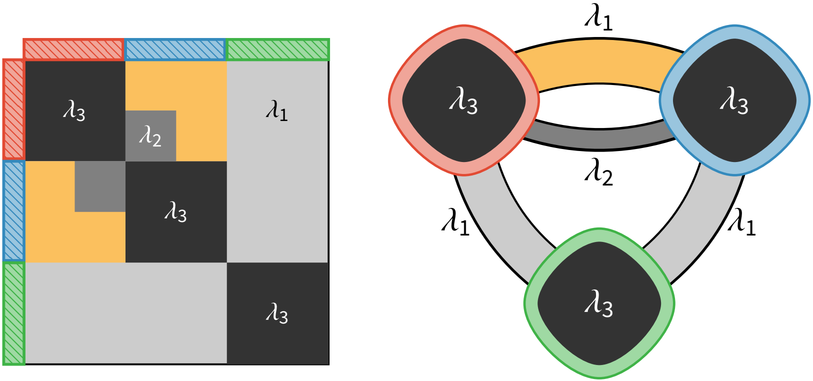
What are the clusters of a graphon?

Each link in this depiction corresponds to a region of the graphon.



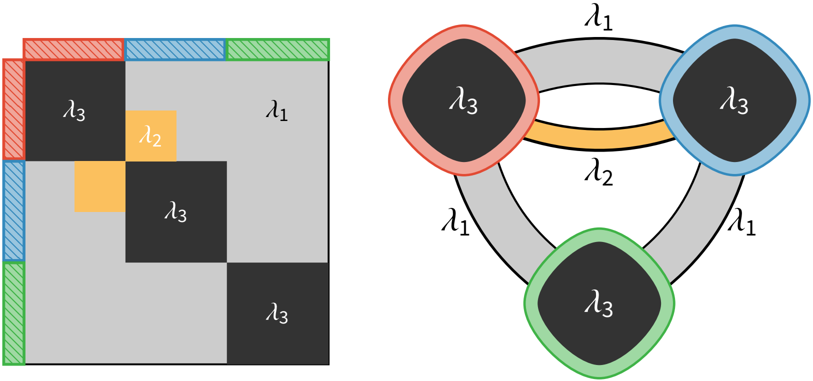
What are the clusters of a graphon?

Each link in this depiction corresponds to a region of the graphon.



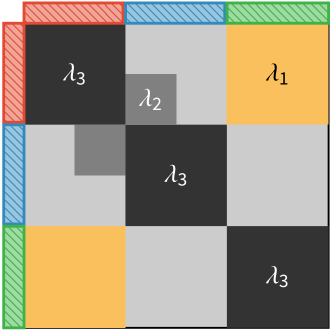
What are the clusters of a graphon?

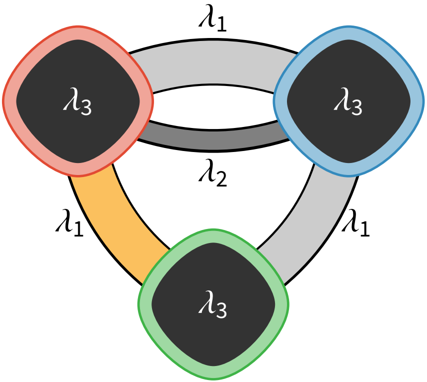
Each link in this depiction corresponds to a region of the graphon.



What are the clusters of a graphon?

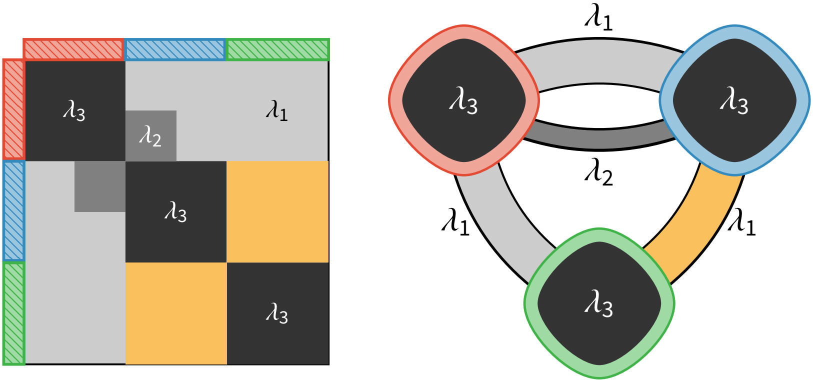
Each link in this depiction corresponds to a region of the graphon.





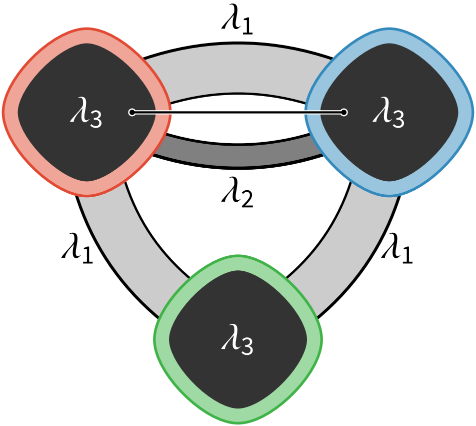
What are the clusters of a graphon?

Each link in this depiction corresponds to a region of the graphon.



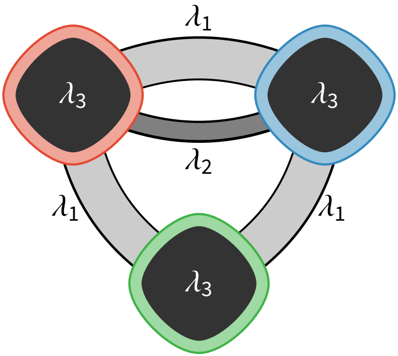
What are the clusters of a graphon?

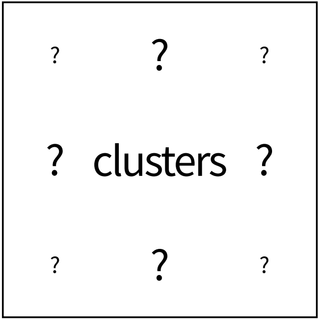
- We define clusters to be connected components.
- Use generalization of graph connectivity, extends (Janson, 2008).
- Key: Insensitive to null sets, e.g., single edges.



What are the clusters of a graphon?

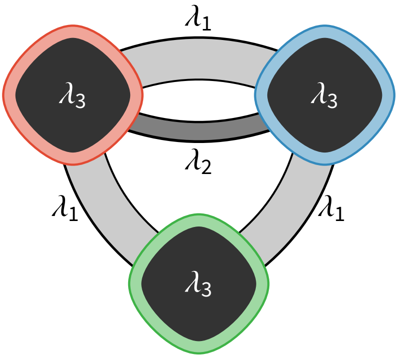
- In fact, we can speak of the clusters at various levels.
- Intuitively: three clusters (connected components) at level λ_3.
- Any pair (●, ●) are in same cluster at λ_3. Same for (●, ●) & (●, ●).

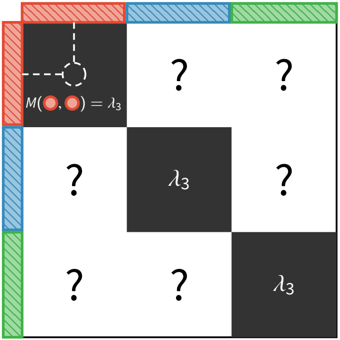




What are the clusters of a graphon?

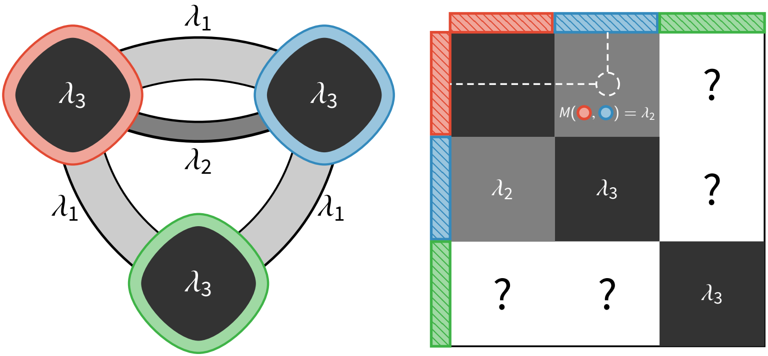
- In fact, we can speak of the clusters at various levels.
- Intuitively: three clusters (connected components) at level λ_3.
- Any pair (●, ●) are in same cluster at λ_3. Same for (●, ●) & (●, ●).
- Naturally encoded as function $M(●, ●) = M(●, ●) = M(●, ●) = \lambda_3$





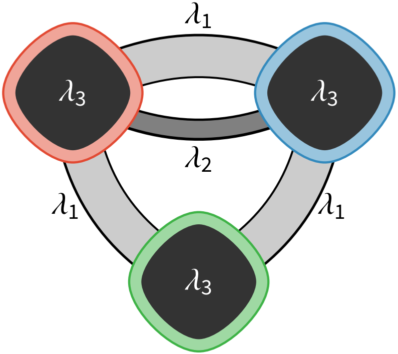
What are the clusters of a graphon?

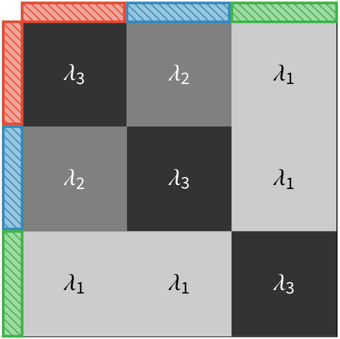
- In fact, we can speak of the clusters at various levels.
- Intuitively: red and blue clusters merge at level λ_2.
- Any pair (●, ●) are in same cluster at λ_2.
- Naturally encoded as $M(●, ●) = M(●, ●) = \lambda_2$.



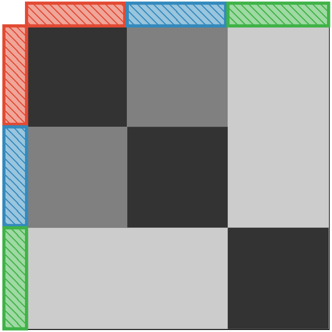
What are the clusters of a graphon?

- In fact, we can speak of the clusters at various levels.
- All clusters merge at level λ_1.
- Encoded as $M(\circ, \circ) = M(\circ, \circ) = \lambda_1$.

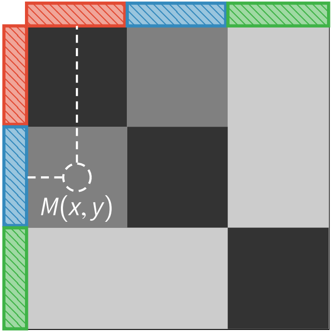




We call M the mergeon.



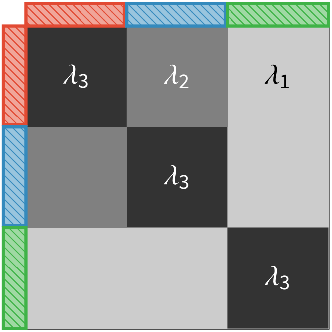
We call M the mergeon.

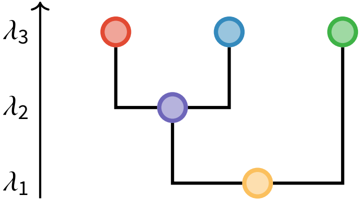


- $M(x, y)$ encodes the first level at which x & y are in same cluster.
- As such, M defines the ground truth clustering of a graphon.
- Note: Mergeon helps deal with subtle technical hurdles.

A mergeon has hierarchical structure.

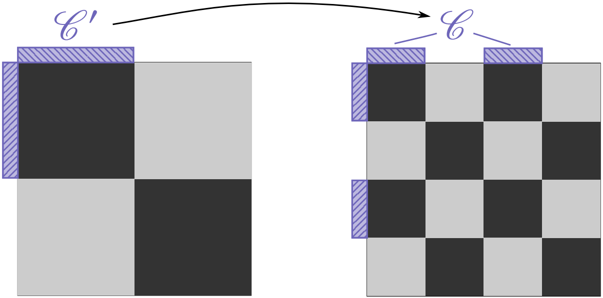
Clusters from higher levels nest within clusters from lower levels.





We call this structure the graphon cluster tree.

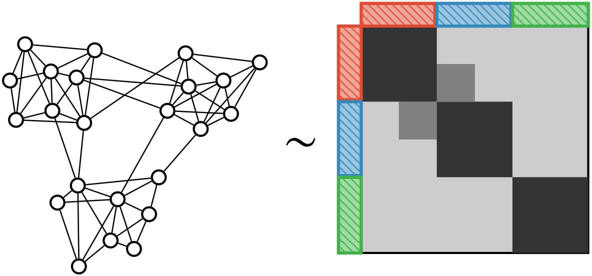
If graphons W_1 and W_2 are the same up to relabeling, then their mergeons and cluster trees are the same up to relabeling.



Surprisingly non-trivial to show.

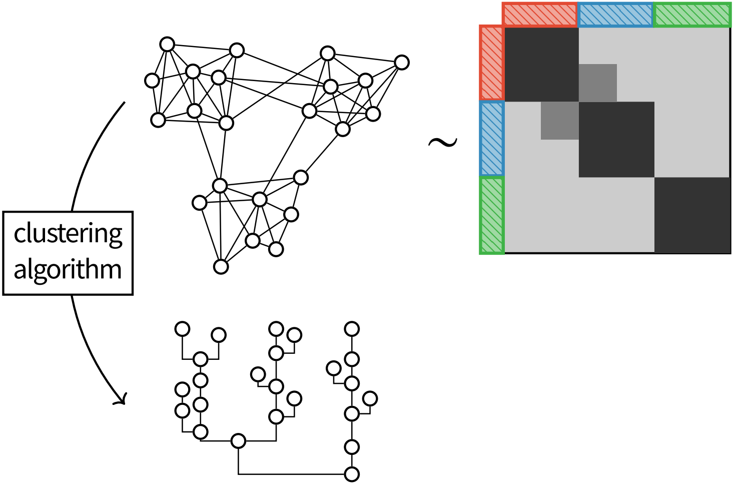
A statistical theory of graphon clustering.

1. What is the ground truth clustering of a graphon?
   - The mergeon, or, equivalently, the graphon cluster tree.
2. How do we define convergence?



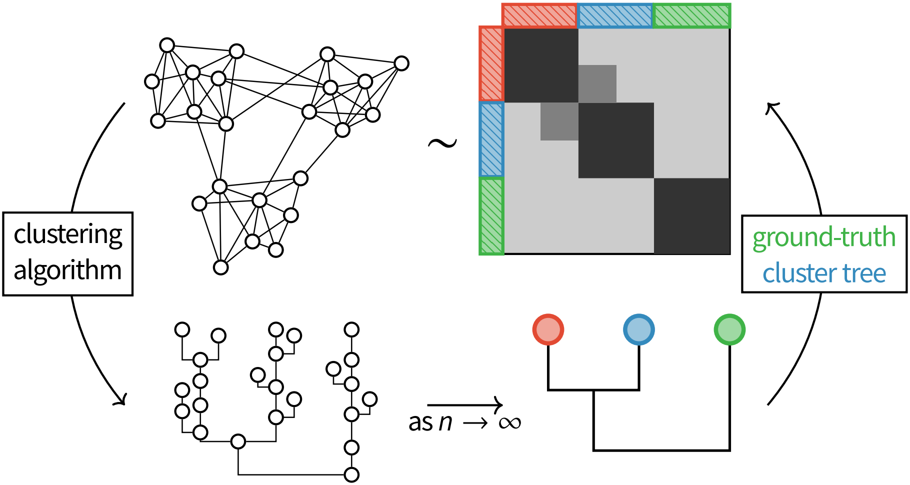
A statistical theory of graphon clustering.

1. What is the ground truth clustering of a graphon?
   - The mergeon, or, equivalently, the graphon cluster tree.
2. How do we define convergence?

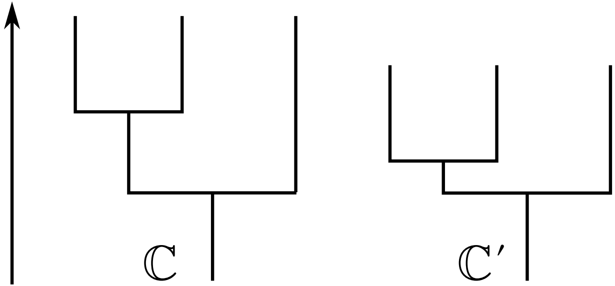


A statistical theory of graphon clustering.

1. What is the ground truth clustering of a graphon?
   - The mergeon, or, equivalently, the graphon cluster tree.
2. How do we define convergence?

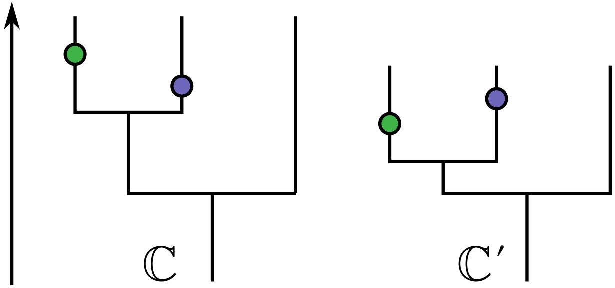


The merge distortion



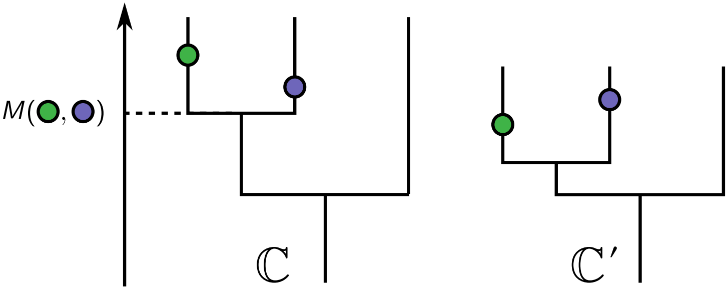
How “close” are $\mathbb{C}$ and $\mathbb{C}'$?

The merge distortion



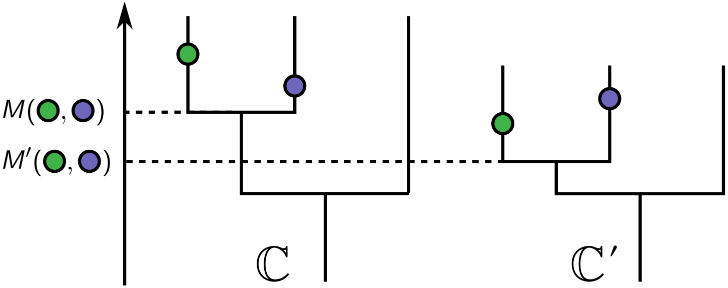
Intuitively, corresponding pairs of nodes should merge at around the same height in each tree.

The merge distortion



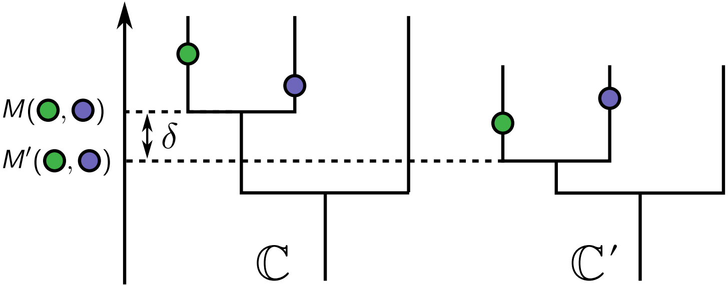
Merge heights are encoded in the mergeon.

The merge distortion



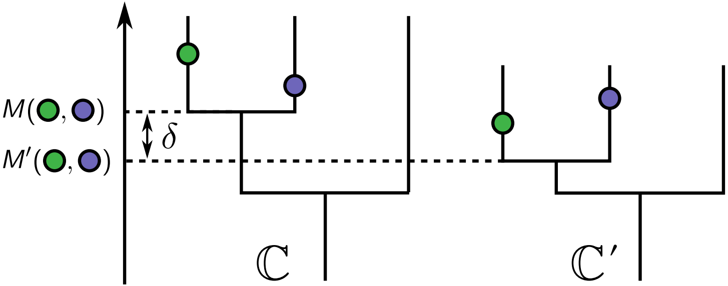
Merge heights are encoded in the mergeon.

The merge distortion



$|M(●, ●) - M'(●, ●)|$ is the difference in merge height of ●, ●.

The merge distortion

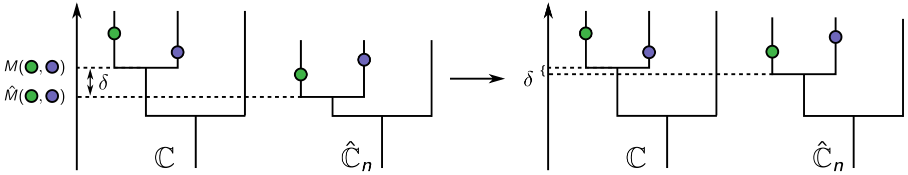


We introduce the merge distortion $d(\mathbb{C}, \mathbb{C}')$:
the maximum difference in merge height over all pairs, i.e,

$$d(\mathbb{C}, \mathbb{C}') = \max_{{\color{green}\bullet}, {\color{blue}\bullet}} \left| M({\color{green}\bullet}, {\color{blue}\bullet}) - M'({\color{green}\bullet}, {\color{blue}\bullet}) \right|.$$

Convergence in merge distortion

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



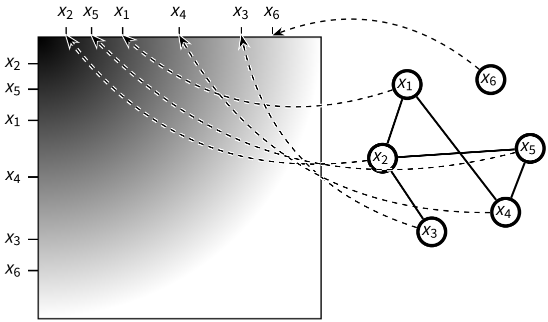
Definition

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

- Consistency $\Longrightarrow$ disjoint clusters are separated as $n \to \infty$.

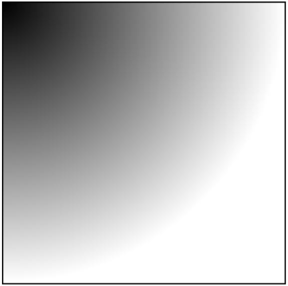
A technical detail...

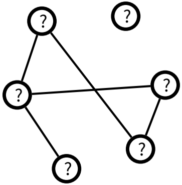
We imagine that the nodes of the graph correspond to graphon nodes.



A technical detail...

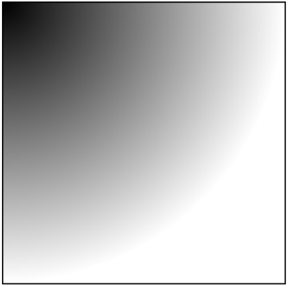
We imagine that the nodes of the graph correspond to graphon nodes.
But this correspondence is latent and unrecoverable.

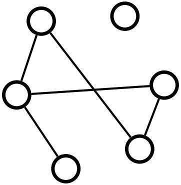




A technical detail...

We imagine that the nodes of the graph correspond to graphon nodes.
But this correspondence is latent and unrecoverable.





- Need correspondence to compute merge distortion.
- Solution: Compute distortion for all possible correspondences.
- Set of correspondences which result in large merge distortion shrinks as $n \to \infty$.

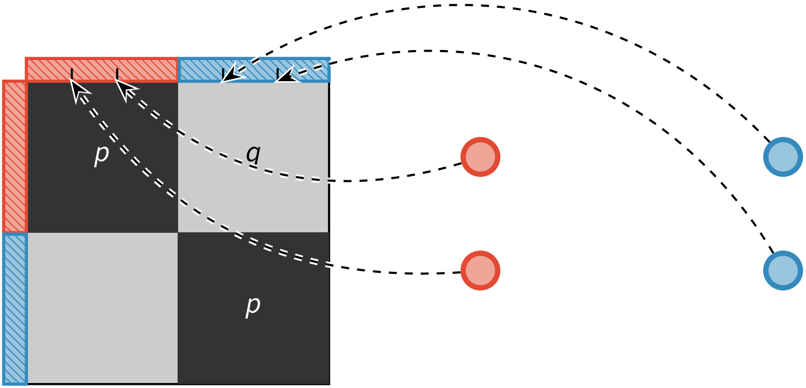
A statistical theory of graphon clustering.

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.

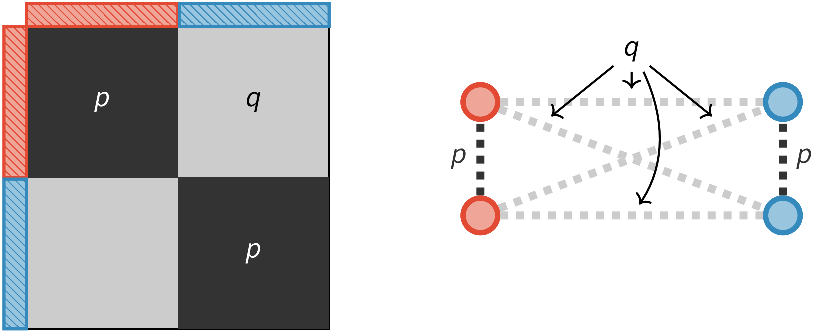
3. Which clustering algorithms are consistent?

Estimating edge probabilities.



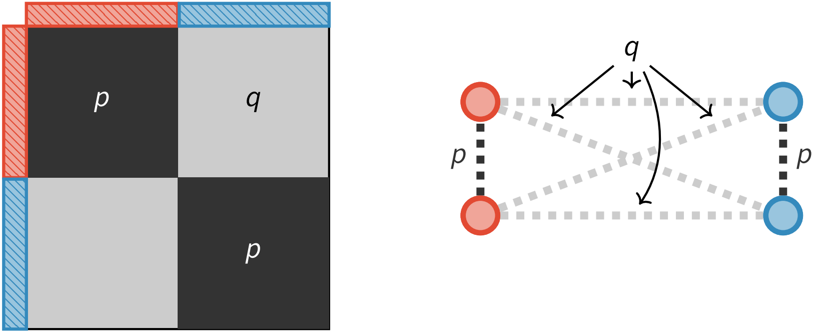
Suppose we sample a graph from this graphon.

Estimating edge probabilities.



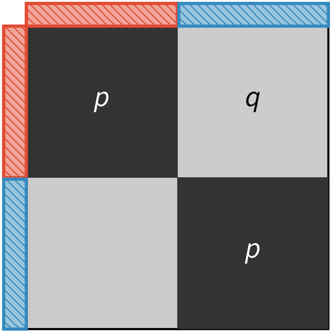
Edges within communities have probability p;
edges across communities have probability q.

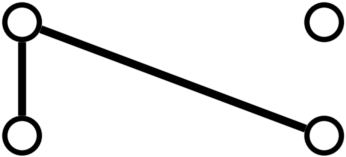
Estimating edge probabilities.



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

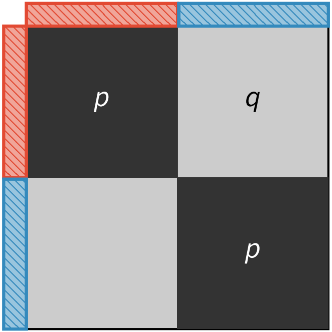
Estimating edge probabilities.

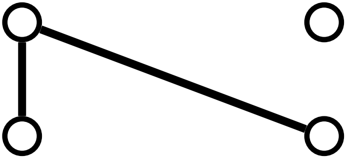




But the edge probabilities are unknown and the presence/absence of an edge (i, j) tells us little about its probability, P_{ij}.

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

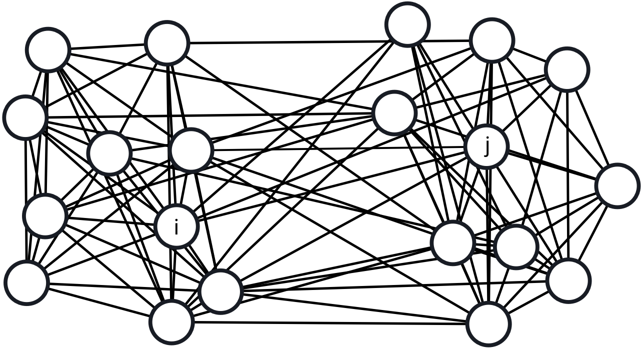
If $||P - \hat{P}||_{max} \to 0$ *in probability as* $n \to \infty$*, then single linkage clustering using* $\hat{P}$ *as the input similarity matrix is a consistent clustering method.*

Theorem

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

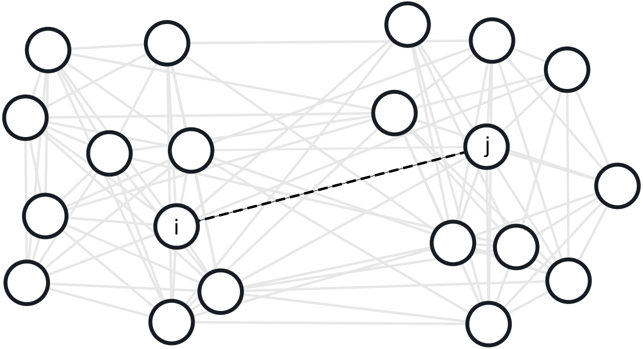
- There are many recent graphon & edge probability estimators.
- But all consistency results are in mean squared error.
- This is too weak. Need consistency in max-norm.
- We modify and analyze the neighborhood smoothing method of (Zhang et al., 2015) to obtain consistency in max-norm.

Neighborhood smoothing



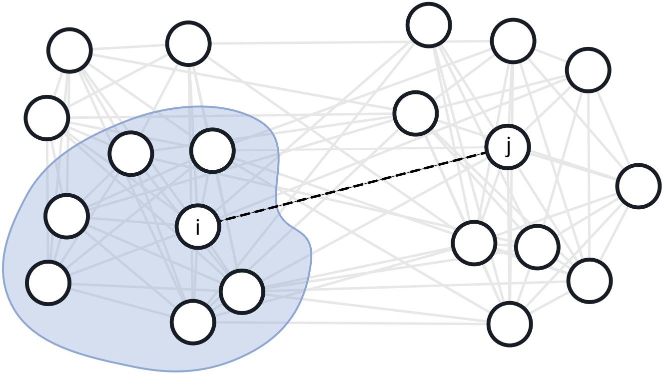
Given this graph...

Neighborhood smoothing



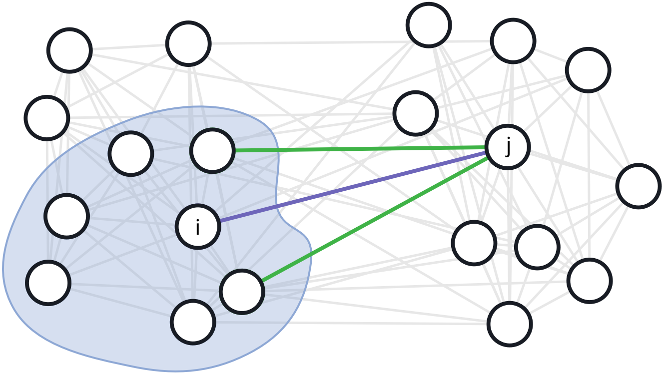
Given this graph... estimate P_{ij}.

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: $\hat{P}_{ij} = {}^2\!/_6 = {}^1\!/_3$.

Consistency of neighborhood smoothing.

Theorem

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

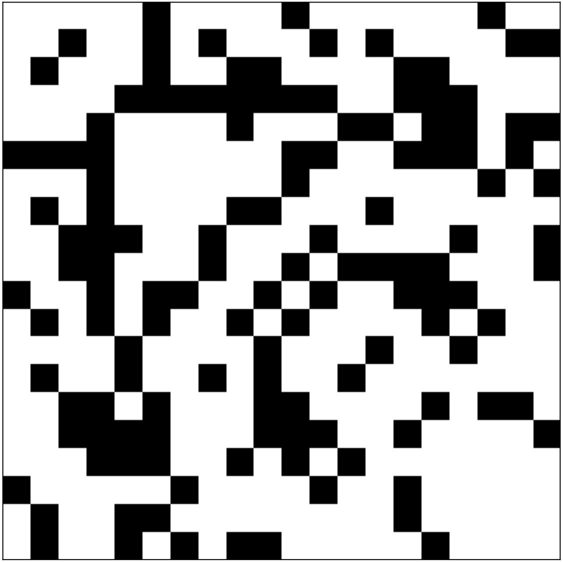
Corollary

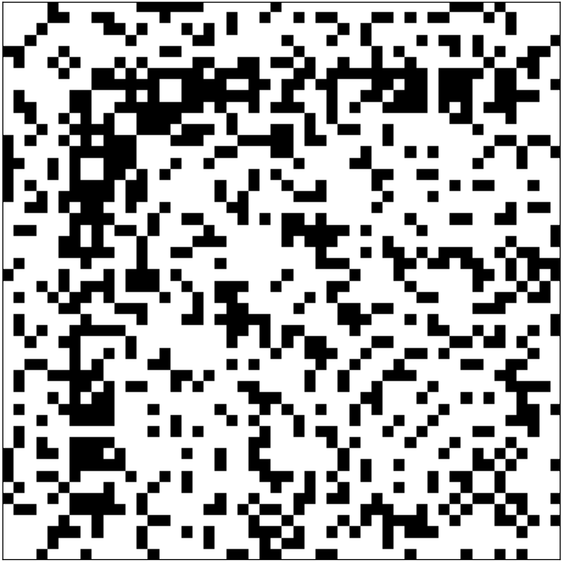
Consistent graphon clustering method:

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.






GRAZIE!
GRAZIE!


GRAZIE!
GRAZIE!


GRAZIE!
GRAZIE!


GRAZIE!
GRAZIE!

Weak isomorphism

- Any graphon W defines a graph distribution.
- Not uniquely! Many graphons define the same distribution.
- The distribution is uniquely determined up to relabeling of W.

Weak isomorphism

- Any graphon W defines a graph distribution.
- Not uniquely! Many graphons define the same distribution.
- The distribution is uniquely determined up to relabeling of W.

Definition

A measure preserving transformation (i.e., graphon relabeling) $\varphi : [0,1] \to [0,1]$ is a Lebesgue-measurable function whose preimage preserves measure. That is, $\mu(\varphi^{-1}(A)) = \mu(A)$ for all measurable $A \subset [0,1]$.

Notation: $W^{\varphi}(x,y) = W(\varphi(x), \varphi(y))$.

Weak isomorphism

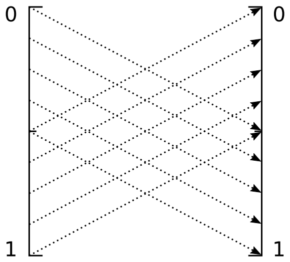
- Any graphon W defines a graph distribution.
- Not uniquely! Many graphons define the same distribution.
- The distribution is uniquely determined up to relabeling of W.

Definition

A measure preserving transformation (i.e., graphon relabeling) $\varphi : [0,1] \to [0,1]$ is a Lebesgue-measurable function whose preimage preserves measure. That is, $\mu(\varphi^{-1}(A)) = \mu(A)$ for all measurable $A \subset [0,1]$.

Notation: $W^{\varphi}(x,y) = W(\varphi(x), \varphi(y))$.

$$\varphi(x) = \begin{cases} x + 1/2 & x \leq 1/2, \\ x - 1/2 & x > 1/2 \end{cases}$$



Weak isomorphism

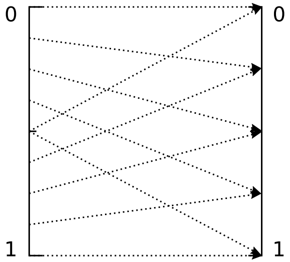
- Any graphon W defines a graph distribution.
- Not uniquely! Many graphons define the same distribution.
- The distribution is uniquely determined up to relabeling of W.

Definition

A measure preserving transformation (i.e., graphon relabeling) $\varphi : [0,1] \to [0,1]$ is a Lebesgue-measurable function whose preimage preserves measure. That is, $\mu(\varphi^{-1}(A)) = \mu(A)$ for all measurable $A \subset [0,1]$.

Notation: $W^{\varphi}(x,y) = W(\varphi(x), \varphi(y))$.

$$\varphi(x) = 2x \mod 1$$



Weak isomorphism

Definition (Lovász)

Two graphons W_1 and W_2 are weakly isomorphic if there exist measure preserving transformations φ_1 and φ_2 such that $W_1^{\varphi_1} \stackrel{\text{a.e.}}{=} W_2^{\varphi_2}$.

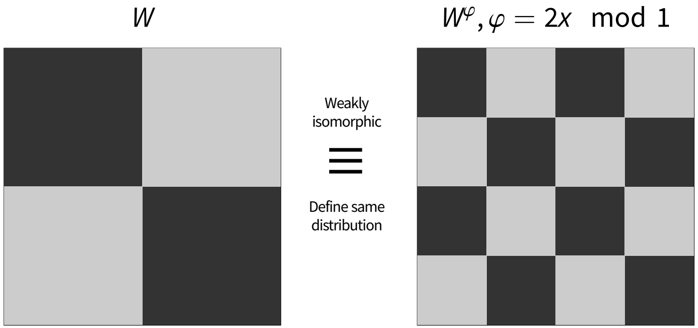
- W_1 and W_2 define the same distribution iff they are weakly isomorphic.

Weak isomorphism

Definition (Lovász)

Two graphons W_1 and W_2 are weakly isomorphic if there exist measure preserving transformations φ_1 and φ_2 such that $W_1^{\varphi_1} \stackrel{\text{a.e.}}{=} W_2^{\varphi_2}$.

- W_1 and W_2 define the same distribution iff they are weakly isomorphic.



The clusters of a graphon

1. Collect all subsets of $[0,1]$ which should be clustered at λ:

$$\mathfrak{A}_\lambda = \{A \subset [0,1] : \mu(A) > 0 \text{ and } A \text{ is connected } \forall\, \lambda' < \lambda.\}$$

2. If $A_1, A_2, A \in \mathfrak{A}_\lambda$, and $A_1 \cup A_2 \subset A$, then A_1, A_2, and A should all be in the same cluster at λ. Consider them equivalent.
   - Define equivalence relation on $\mathfrak{A}_\lambda$:

$$A_1 \multimap\!\circ_\lambda A_2 \iff \exists A \in \mathfrak{A}_\lambda, A \supset A_1 \cup A_2.$$

   - Read: A_1 is clustered with A_2 at level λ.
   - $\multimap\!\circ_\lambda$ partitions $\mathfrak{A}_\lambda$ into equivalence classes of sets which should be in the same cluster.

The clusters of a graphon

3. Define clusters to be "largest" element of each equivalence class.
   - Subtlety in defining "largest":
     - Suppose $\mathscr{A} \in \mathfrak{A}_\lambda / \multimap_\lambda$ is such an equivalence class.
     - Let A be any representative from $\mathscr{A}$, let Z be a set of zero measure.
     - $A' = A \cup Z$ is a representative of $\mathscr{A}$.
   - In general there is no representative of $\mathscr{A}$ which strictly contains all other representatives in $\mathscr{A}$
   - We can find reps. which contain every other rep. up to a null set, called the "essential maxima" of $\mathscr{A}$:

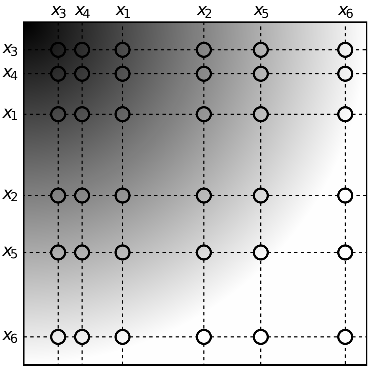
$$\text{ess max}\,\mathscr{A} = \{A \in \mathscr{A} : \forall A' \in \mathscr{A},\ \mu(A' \setminus A) = 0\}$$

   - The clusters of W at level λ are the essential maxima of each equivalence class:

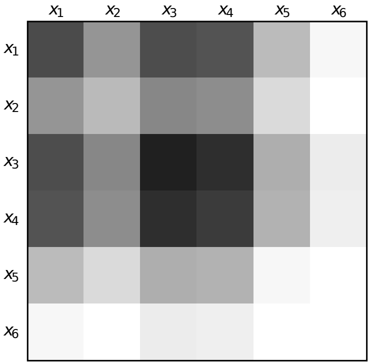
$$\mathbb{C}_W(\lambda) = \{\text{ess max}\,\mathscr{A} : \mathscr{A} \in \mathfrak{A}_\lambda / \multimap_\lambda.\}$$

Consistent algorithms

- Intuitively, estimating the graphon is related to clustering.
- It suffices to estimate the so-called edge probability matrix.



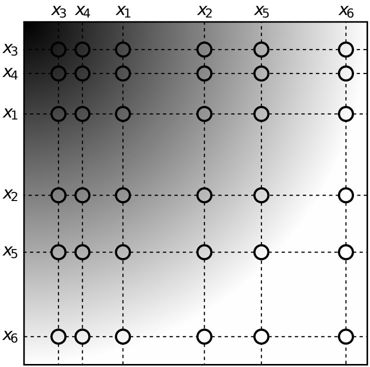
W



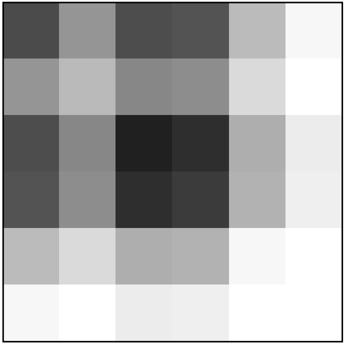
$P : P_{ij} = W(x_i, x_j)$

Consistent algorithms

- Intuitively, estimating the graphon is related to clustering.
- It suffices to estimate the so-called edge probability matrix.



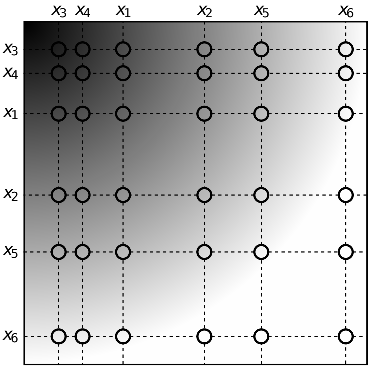
W



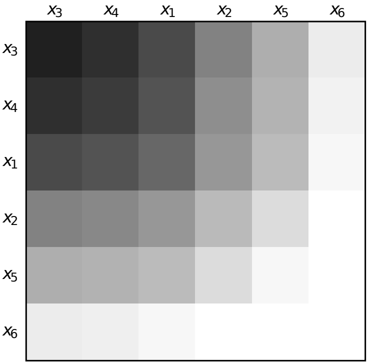
$P : P_{ij} = W(x_i, x_j)$

Consistent algorithms

- Intuitively, estimating the graphon is related to clustering.
- It suffices to estimate the so-called edge probability matrix.

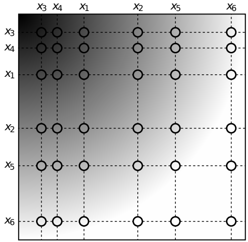


W

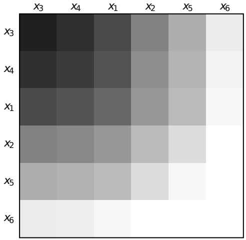


P (artificially permuted)

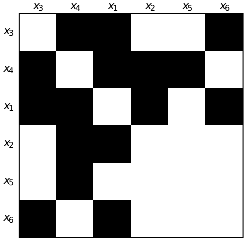
Sample an adjacency matrix A from P:



W

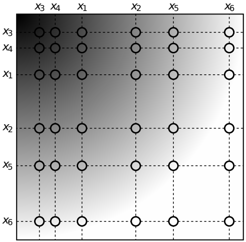


P
(artificially permuted)

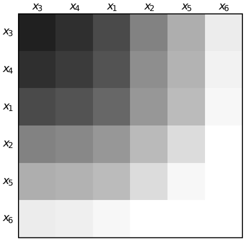


A

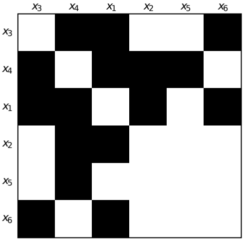
Sample an adjacency matrix A from P:



W



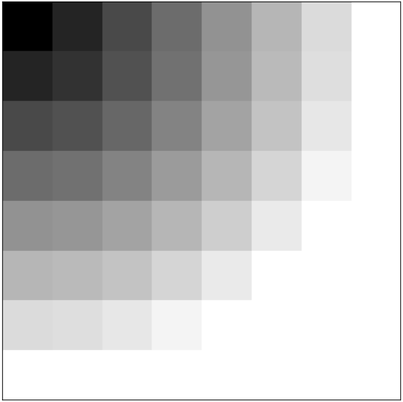
P
(artificially permuted)



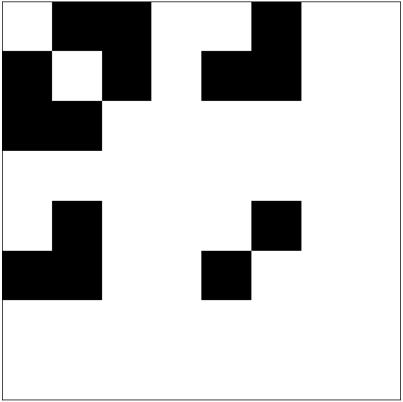
A

A is a poor estimate of P.

$n = 8$

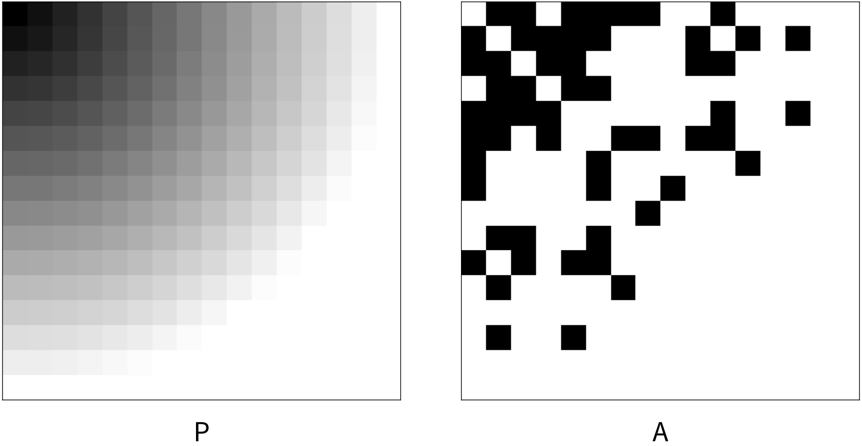


P

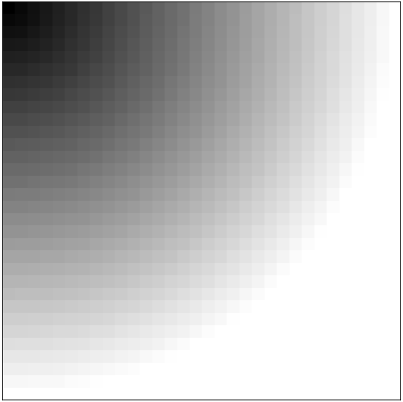


A

$n = 16$



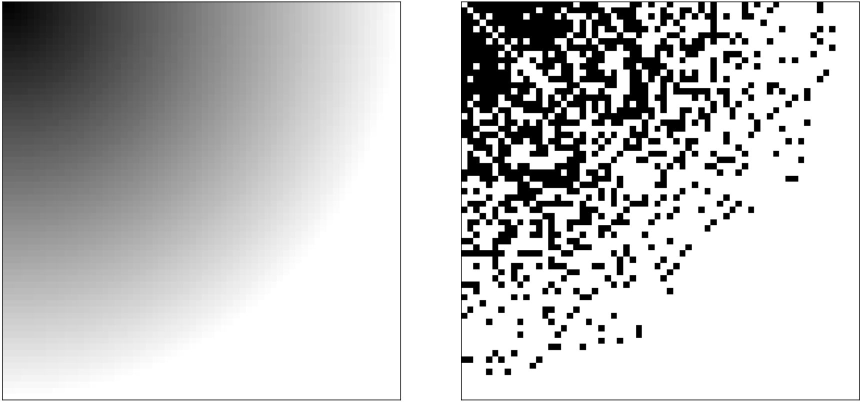
$n = 32$



P

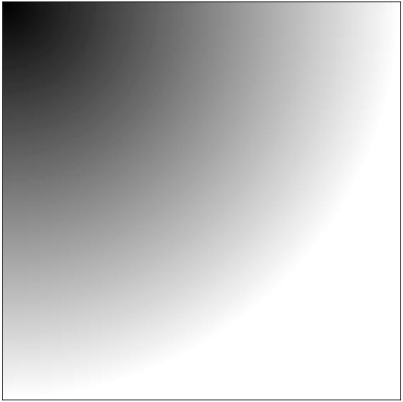
A

$n = 64$

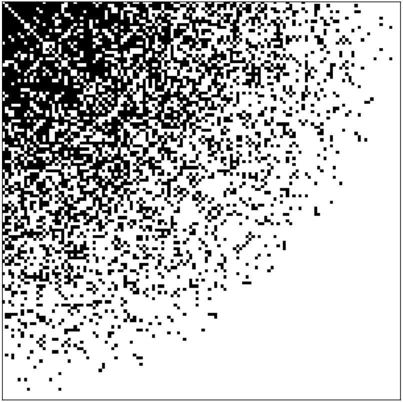


P A

$n = 128$

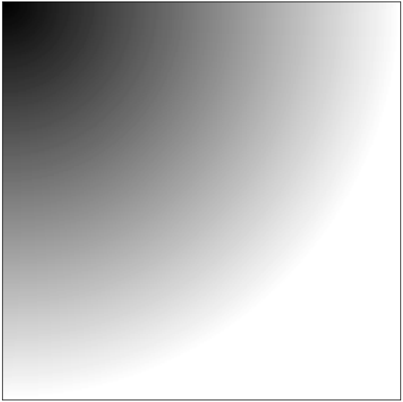


P

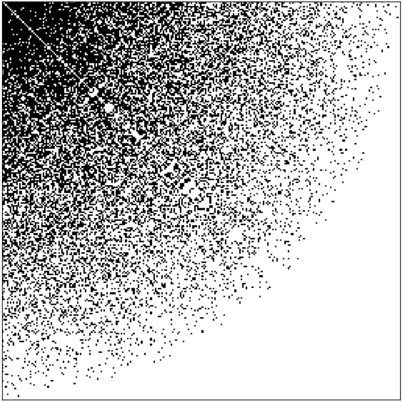


A

$n = 256$



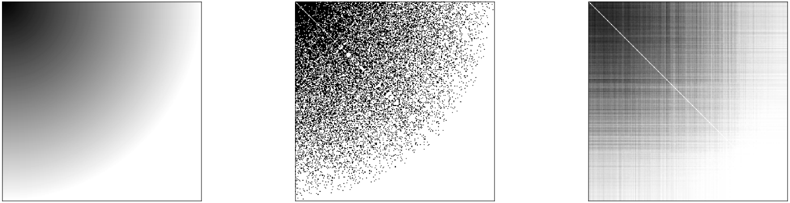
P



A

Edge probability estimation

Goal: Compute estimated edge probabilities $\hat{P}$ from A.



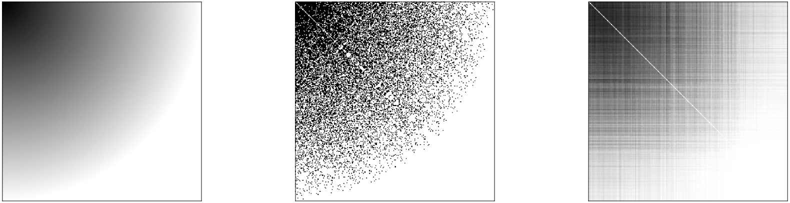
P A $\hat{P}$

Theorem

If $\|P - \hat{P}\|_{max} \to 0$ in probability as $n \to \infty$, then single linkage clustering on $\hat{P}$ is a consistent clustering method.

Edge probability estimation

Goal: Compute estimated edge probabilities $\hat{P}$ from A.



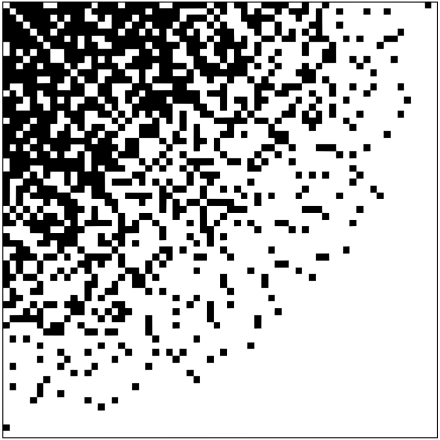
P $\qquad$ A $\qquad$ $\hat{P}$

Theorem

If $\|P - \hat{P}\|_{max} \to 0$ in probability as $n \to \infty$, then single linkage clustering on $\hat{P}$ is a consistent clustering method.

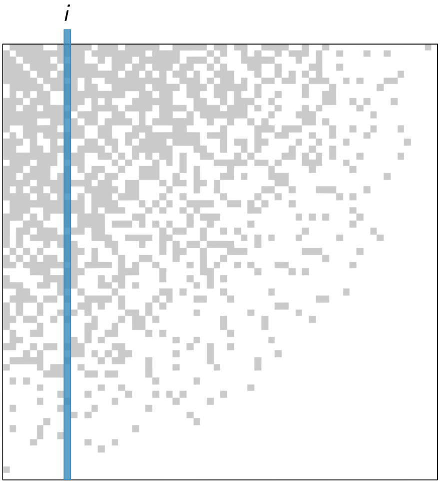
- We need a suitable estimator $\hat{P}$ of edge probabilities.
- Recently, Zhang et al. (2015) proposed neighborhood smoothing.

Neighborhood smoothing



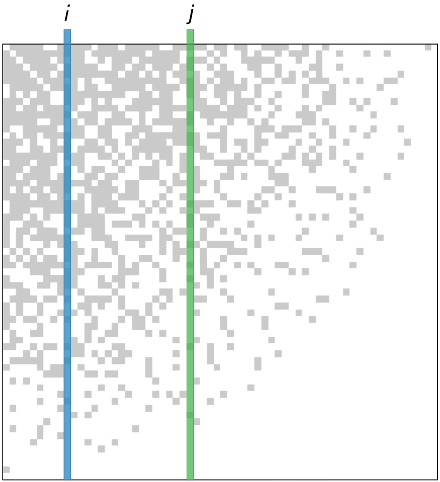
Given A, the adjacency matrix of a sampled graph...

Neighborhood smoothing



Consider a node i and its corresponding column of A.

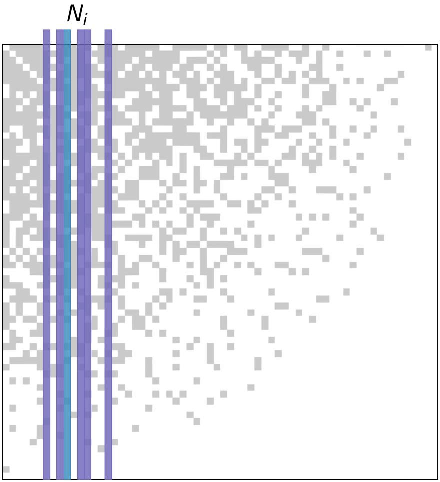
Neighborhood smoothing



Measure similarity to every other node j:

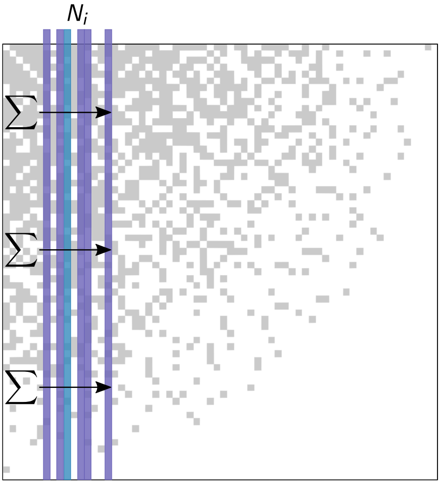
$$d(i,j) = \max_{k \neq i,j} \left| (A^2)_{ik} - (A^2)_{jk} \right|$$

Neighborhood smoothing



Form neighborhood N_i of nodes most similar to i.

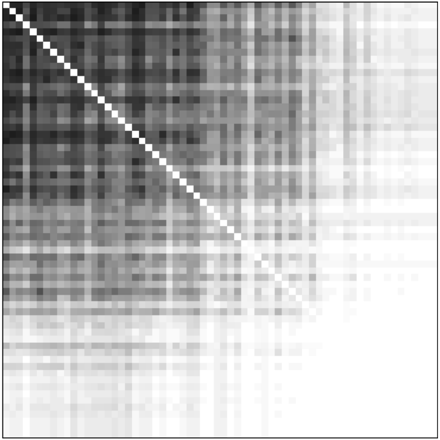
Neighborhood smoothing



Average within neighborhood to estimate edge probability:

$$\hat{P}_{ij} = \frac{1}{2|N_i|} \sum_{i' \in N_i} A_{i'j} + \frac{1}{2|N_j|} \sum_{j' \in N_j} A_{ij'}$$

Neighborhood smoothing



The result is a smoothed estimate $\hat{P}$ of edge probabilities.