Exponentially smoothed spectral clustering and dynamic stochastic block model

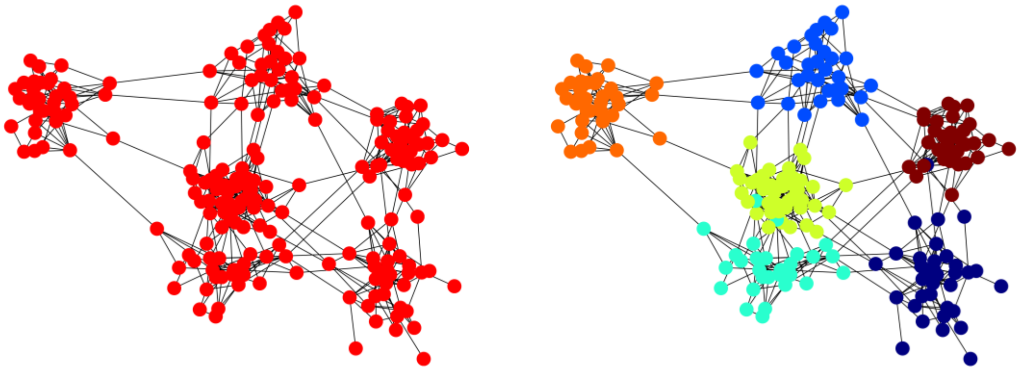
N. Keriven[1], Samuel Vaiter[2]

[1]Ecole Normale Supérieure, Paris (CFM-ENS chair)

[2]Institut de Mathématiques de Bourgogne

GraphSig May 2019

Spectral Clustering

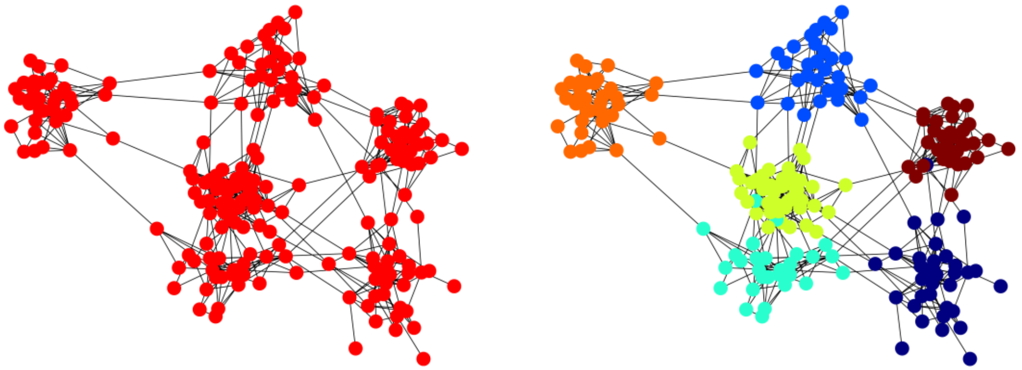


Cluster the nodes of a graph using its structure.

Application in :

- Social network analysis
- Protein analysis
- etc

Spectral Clustering

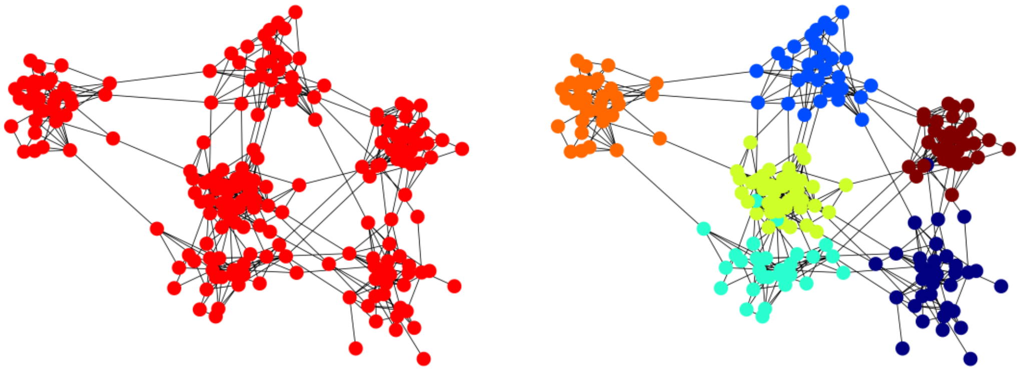


Cluster the nodes of a graph using its structure.

Application in :
- Social network analysis
- Protein analysis
- etc

Classical algorithm:

Spectral Clustering



Cluster the nodes of a graph using its structure.

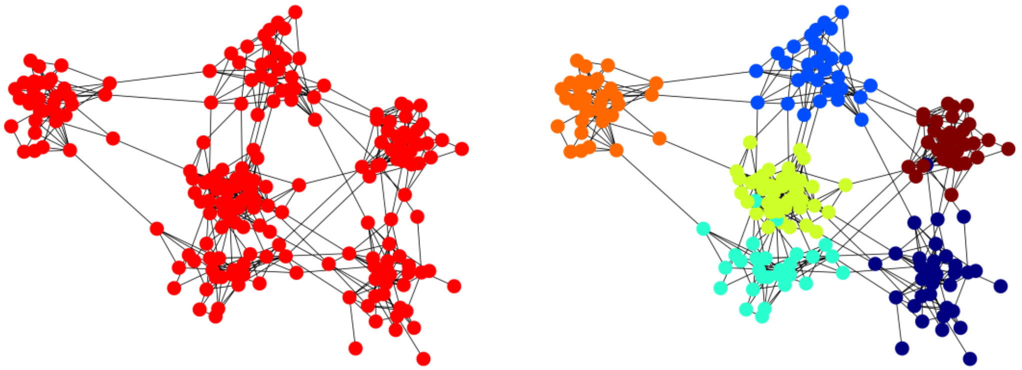
Application in :

- Social network analysis
- Protein analysis
- etc

Classical algorithm:

- Take $W = \begin{cases} A \\ D - A \\ Id - D^{-1/2} A D^{-1/2} \end{cases}$

Spectral Clustering



Cluster the nodes of a graph using its structure.

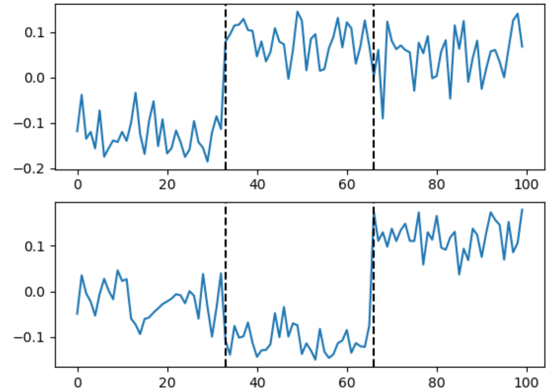
Application in :
- Social network analysis
- Protein analysis
- etc

Classical algorithm:

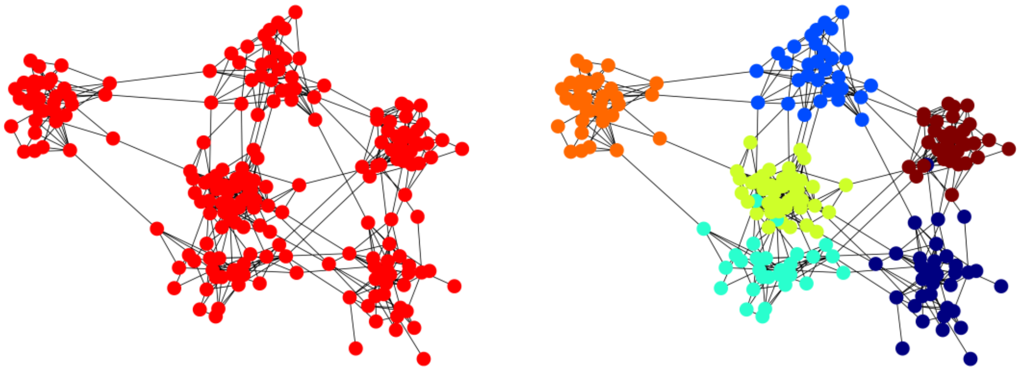
- Take $W = \begin{cases} A \\ D - A \\ Id - D^{-1/2} A D^{-1/2} \end{cases}$

- Compute its k-SVD $W = U \Delta U^\top,\ \tilde{U} = U_{:,1:k}$

Eigenvectors :



Spectral Clustering



Cluster the nodes of a graph using its structure.

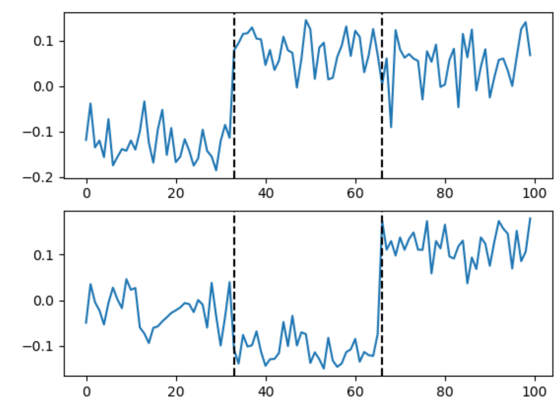
Application in :

- Social network analysis
- Protein analysis
- etc

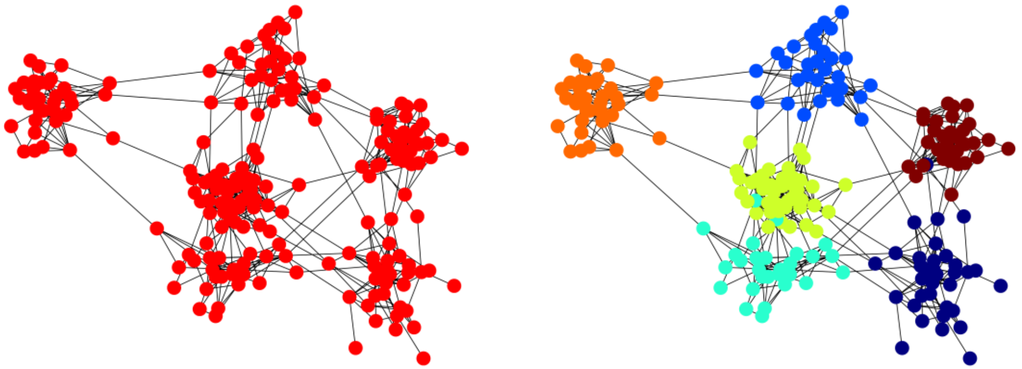
Classical algorithm:

- Take $W = \begin{cases} A \\ D - A \\ Id - D^{-1/2} A D^{-1/2} \end{cases}$
- Compute its k-SVD $W = U \Delta U^\top, \tilde{U} = U_{:,1:k}$
- Cluster the rows of $\tilde{U}$ with k-means

Eigenvectors :



Spectral Clustering



Cluster the nodes of a graph using its structure.

Application in :

- Social network analysis
- Protein analysis
- etc

Classical algorithm:

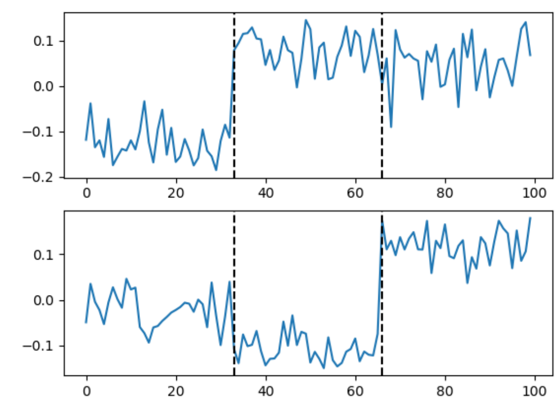
- Take $W = \begin{cases} A \\ D - A \\ Id - D^{-1/2} A D^{-1/2} \end{cases}$

- Compute its k-SVD $W = U \Delta U^\top, \tilde{U} = U_{:,1:k}$

- Cluster the rows of $\tilde{U}$ with k-means

- Many many (fast) variants…

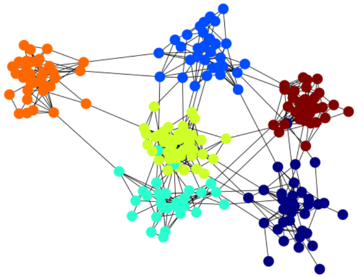
Eigenvectors :



SBM : theoretical analysis

Stochastic Block Model (SBM)

$$\begin{cases} a_{ij} \sim \mathrm{Ber}(B_{k\ell}) \\ \text{when } \Theta_{ik} = 1, \Theta_{j\ell} = 1 \end{cases}$$



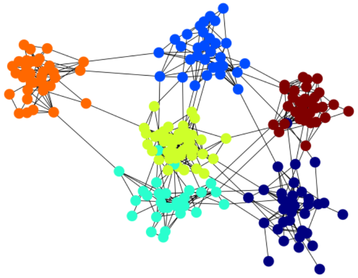
$\Theta \in \{0,1\}^{n \times K}$: *matrix of communities (only one 1 by row)*

Of course : $B_{kk} > B_{k\ell}$

SBM : theoretical analysis

Stochastic Block Model (SBM)

$$\begin{cases} a_{ij} \sim \mathrm{Ber}(B_{k\ell}) \\ \text{when } \Theta_{ik} = 1, \Theta_{j\ell} = 1 \end{cases}$$



$\Theta \in \{0,1\}^{n \times K}$: *matrix of communities (only one 1 by row)*

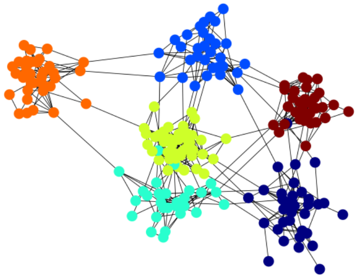
Of course : $B_{kk} > B_{k\ell}$

Theoretical results (non-exhaustive…)

SBM : theoretical analysis

Stochastic Block Model (SBM)

$$\begin{cases} a_{ij} \sim \mathrm{Ber}(B_{k\ell}) \\ \text{when } \Theta_{ik} = 1, \Theta_{j\ell} = 1 \end{cases}$$



$\Theta \in \{0,1\}^{n\times K}$: *matrix of communities (only one 1 by row)*

Of course : $B_{kk} > B_{k\ell}$

Theoretical results (non-exhaustive…)

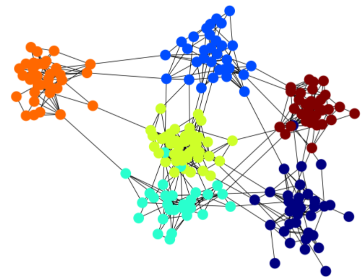
Recently solved conjecture(s)

[Krzakala et al., Mossel et al, Massoulie et al…]

SBM : theoretical analysis

Stochastic Block Model (SBM)

$$\begin{cases} a_{ij} \sim \mathrm{Ber}(B_{k\ell}) \\ \text{when } \Theta_{ik} = 1, \Theta_{j\ell} = 1 \end{cases}$$



$\Theta \in \{0,1\}^{n \times K}$: *matrix of communities (only one 1 by row)*

Of course : $B_{kk} > B_{k\ell}$

Theoretical results (non-exhaustive…)

Recently solved conjecture(s)

[Krzakala et al., Mossel et al, Massoulie et al…]

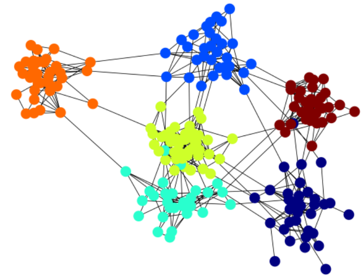
- **Sparse** SBM: $B_{kk} = \frac{a}{n}, B_{k\ell} = \frac{b}{n}$

SBM : theoretical analysis

Stochastic Block Model (SBM)

$$\begin{cases} a_{ij} \sim \mathrm{Ber}(B_{k\ell}) \\ \text{when } \Theta_{ik} = 1, \Theta_{j\ell} = 1 \end{cases}$$

$\Theta \in \{0,1\}^{n \times K}$*: matrix of communities (only one 1 by row)*



Of course : $B_{kk} > B_{k\ell}$

Theoretical results (non-exhaustive…)

Recently solved conjecture(s)

[Krzakala et al., Mossel et al, Massoulie et al…]

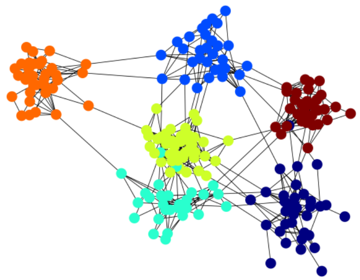
- **Sparse** SBM: $B_{kk} = \frac{a}{n}, B_{k\ell} = \frac{b}{n}$
- When $n \to \infty$, **detectability threshold** for any algorithm wrt a, b

SBM : theoretical analysis

Stochastic Block Model (SBM)

$$\begin{cases} a_{ij} \sim \mathrm{Ber}(B_{k\ell}) \\ \text{when } \Theta_{ik} = 1, \Theta_{j\ell} = 1 \end{cases}$$

$\Theta \in \{0,1\}^{n \times K}$*: matrix of communities (only one 1 by row)*



Of course : $B_{kk} > B_{k\ell}$

Theoretical results (non-exhaustive…)

Recently solved conjecture(s)

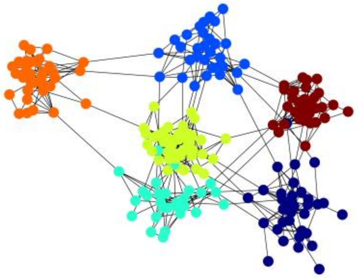
[Krzakala et al., Mossel et al, Massoulie et al…]

- **Sparse** SBM: $B_{kk} = \frac{a}{n}, B_{k\ell} = \frac{b}{n}$
- When $n \to \infty$, **detectability threshold** for any algorithm wrt a, b
- Case $K \geq 3$ still open…

SBM : theoretical analysis

Stochastic Block Model (SBM)

$$\begin{cases} a_{ij} \sim \mathrm{Ber}(B_{k\ell}) \\ \text{when } \Theta_{ik} = 1, \Theta_{j\ell} = 1 \end{cases}$$



$\Theta \in \{0,1\}^{n \times K}$: *matrix of communities (only one 1 by row)*

Of course : $B_{kk} > B_{k\ell}$

Theoretical results (non-exhaustive…)

Recently solved conjecture(s)

[Krzakala et al., Mossel et al, Massoulie et al…]

- **Sparse** SBM: $B_{kk} = \frac{a}{n}, B_{k\ell} = \frac{b}{n}$
- When $n \to \infty$, **detectability threshold** for any algorithm wrt a, b
- Case $K \geq 3$ still open…

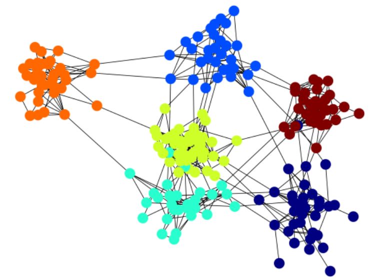
Non-asymptotic analysis

[Lei 2015]

SBM : theoretical analysis

Stochastic Block Model (SBM)

$$\begin{cases} a_{ij} \sim \mathrm{Ber}(B_{k\ell}) \\ \text{when } \Theta_{ik} = 1, \Theta_{j\ell} = 1 \end{cases}$$



$\Theta \in \{0,1\}^{n \times K}$: *matrix of communities (only one 1 by row)*

Of course : $B_{kk} > B_{k\ell}$

Theoretical results (non-exhaustive…)

Recently solved conjecture(s)

[Krzakala et al., Mossel et al, Massoulie et al…]

- **Sparse** SBM: $B_{kk} = \frac{a}{n}, B_{k\ell} = \frac{b}{n}$
- When $n \to \infty$, **detectability threshold** for any algorithm wrt a, b
- Case $K \geq 3$ still open…

Non-asymptotic analysis

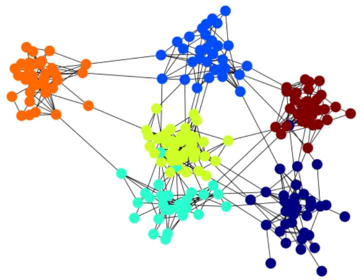
[Lei 2015]

- SC with $W = A$

SBM : theoretical analysis

Stochastic Block Model (SBM)

$$\begin{cases} a_{ij} \sim \mathrm{Ber}(B_{k\ell}) \\ \text{when } \Theta_{ik} = 1, \Theta_{j\ell} = 1 \end{cases}$$



$\Theta \in \{0,1\}^{n\times K}$: *matrix of communities (only one 1 by row)*

Of course : $B_{kk} > B_{k\ell}$

Theoretical results (non-exhaustive…)

Recently solved conjecture(s)

[Krzakala et al., Mossel et al, Massoulie et al…]

- **Sparse** SBM: $B_{kk} = \frac{a}{n}, B_{k\ell} = \frac{b}{n}$
- When $n \to \infty$, **detectability threshold** for any algorithm wrt a, b
- Case $K \geq 3$ still open…

Non-asymptotic analysis

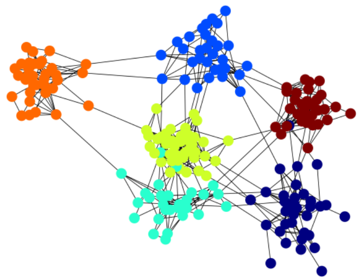
[Lei 2015]

- SC with $W = A$
- ***Almost*** sparse: $B_{k\ell} \sim \alpha_n \geq \frac{\log n}{n}$

SBM : theoretical analysis

Stochastic Block Model (SBM)

$$\begin{cases} a_{ij} \sim \mathrm{Ber}(B_{k\ell}) \\ \text{when } \Theta_{ik} = 1, \Theta_{j\ell} = 1 \end{cases}$$



$\Theta \in \{0,1\}^{n \times K}$: *matrix of communities (only one 1 by row)*

Of course : $B_{kk} > B_{k\ell}$

Theoretical results (non-exhaustive…)

Recently solved conjecture(s)

[Krzakala et al., Mossel et al, Massoulie et al…]

- **Sparse** SBM: $B_{kk} = \frac{a}{n}, B_{k\ell} = \frac{b}{n}$
- When $n \to \infty$, **detectability threshold** for any algorithm wrt a, b
- Case $K \geq 3$ still open…

Non-asymptotic analysis

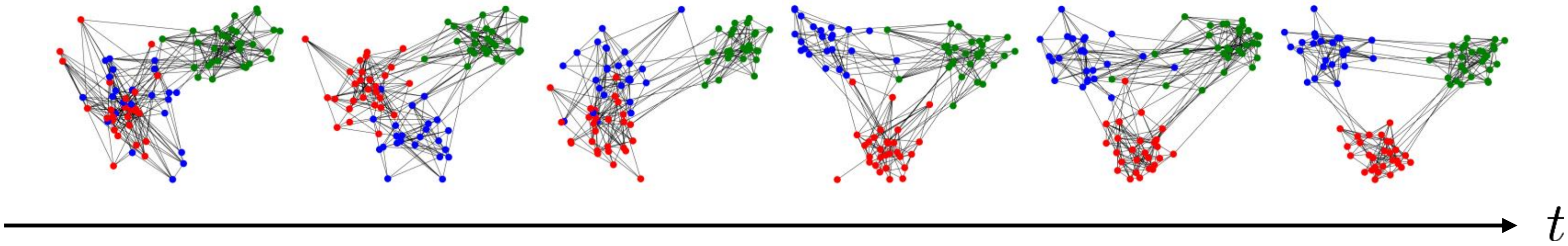
[Lei 2015]

- SC with $W = A$
- ***Almost*** sparse: $B_{k\ell} \sim \alpha_n \geq \frac{\log n}{n}$
- With proba $1 - n^{-r}$:

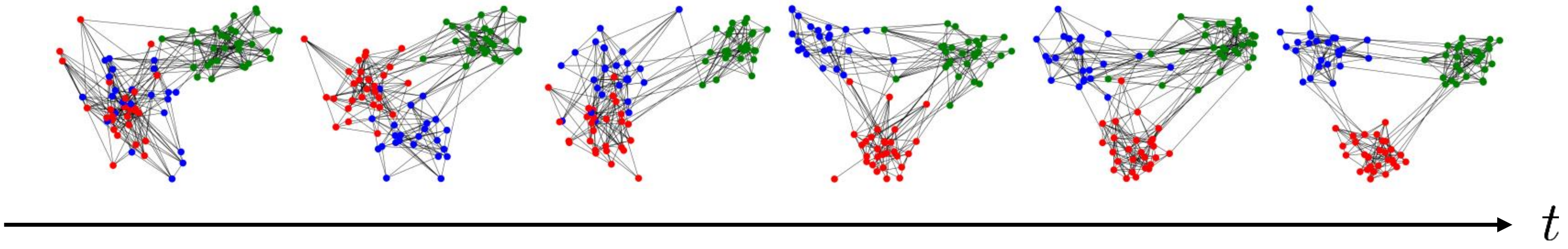
$$\boxed{L(\hat{\Theta}, \Theta) \lesssim \frac{K^2}{n\alpha_n}}$$

$$L(\hat{\Theta}, \Theta) = \min_P \|\hat{\Theta}P - \Theta\|_0$$

Dynamic Spectral Clustering



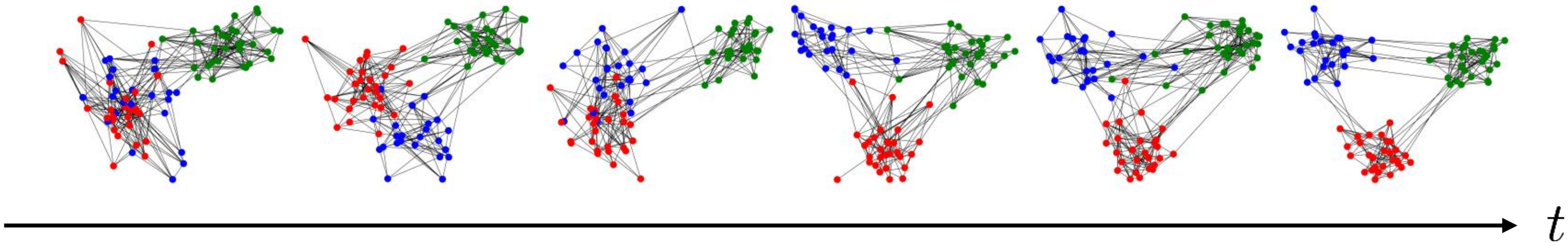
Dynamic Spectral Clustering



Goal

Exploit past data to:

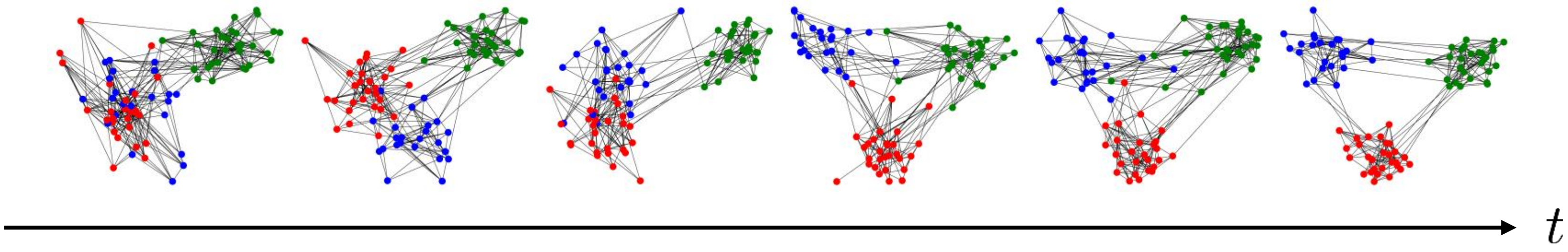
Dynamic Spectral Clustering



Goal

Exploit past data to:
- Track communities

Dynamic Spectral Clustering

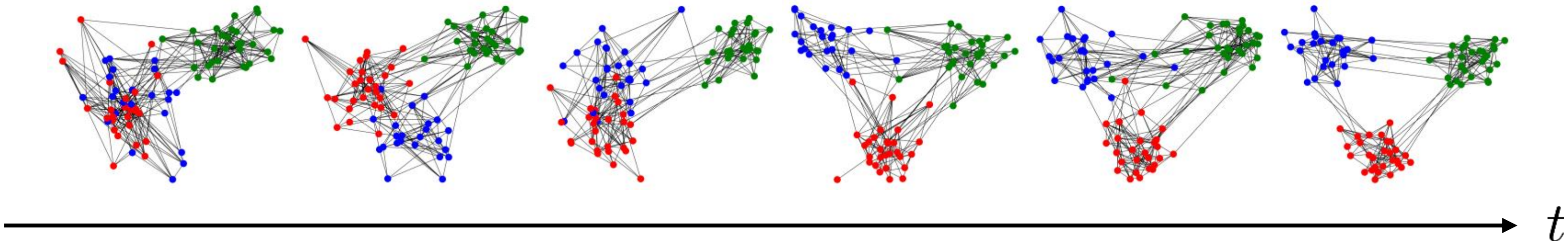


Goal

Exploit past data to:

- Track communities
- Enforce smoothness/consistency

Dynamic Spectral Clustering

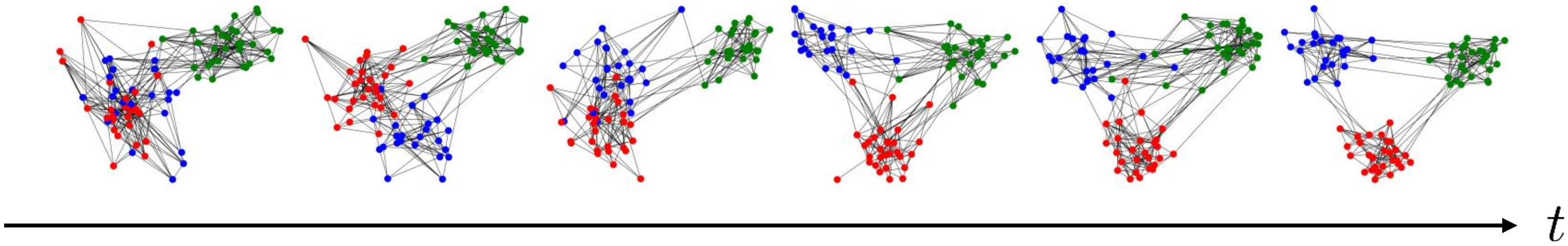


Goal

Exploit past data to:

- Track communities
- Enforce smoothness/consistency
- **Improve result at time t**

Dynamic Spectral Clustering

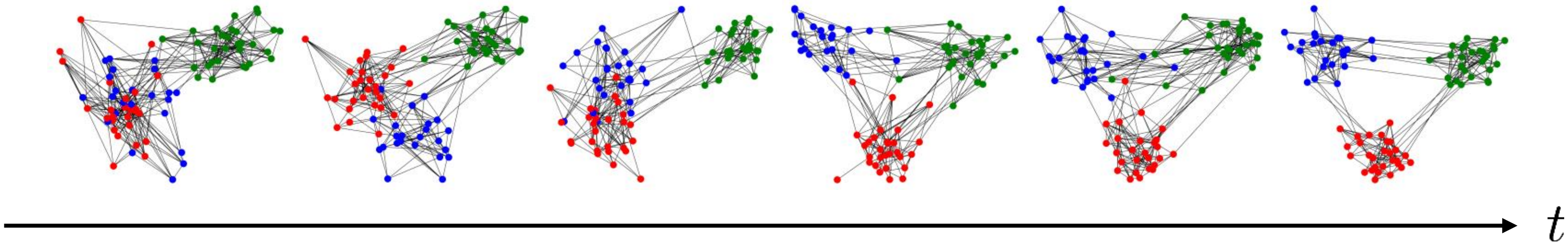


Goal

Exploit past data to:

- Track communities
- Enforce smoothness/consistency
- **Improve result at time t**
  - *Does not want to apply SC several times !*

Dynamic Spectral Clustering



Goal

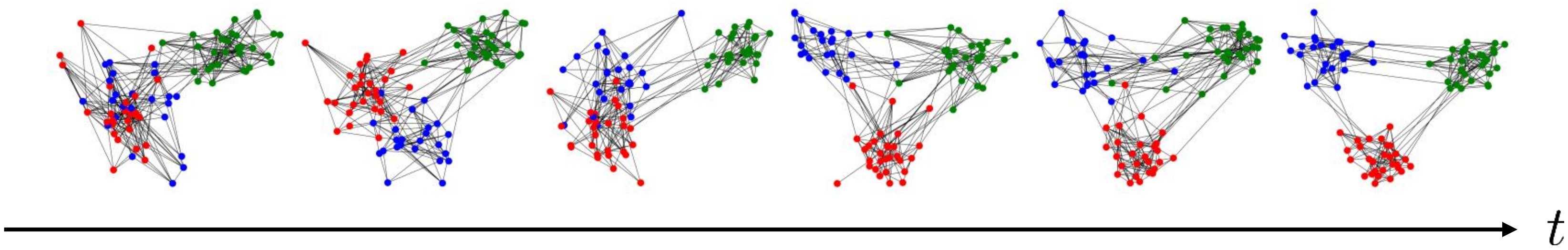
Exploit past data to:

- Track communities
- Enforce smoothness/consistency
- **Improve result at time t**
    - *Does not want to apply SC several times !*

Many approaches :

- Incremental / hierarchical
- Maximum Likelihood / Bayesian
- Variational…

Dynamic Spectral Clustering



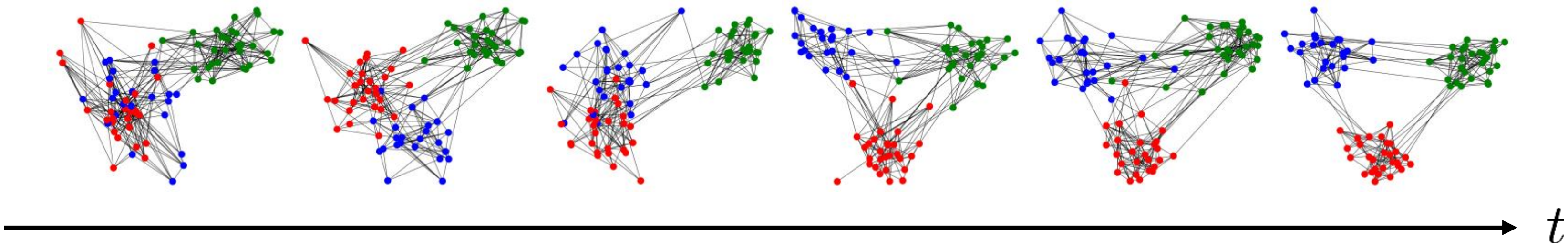
Goal

Exploit past data to:
- Track communities
- Enforce smoothness/consistency
- **Improve result at time t**
  - *Does not want to apply SC several times !*

Many approaches :
- Incremental / hierarchical
- Maximum Likelihood / Bayesian
- Variational…

Simple(st): **Smoothing of adjacency matrix + SC**

Dynamic Spectral Clustering



Goal

Exploit past data to:

- Track communities
- Enforce smoothness/consistency
- **Improve result at time t**
    - *Does not want to apply SC several times !*

Many approaches :

- Incremental / hierarchical
- Maximum Likelihood / Bayesian
- Variational…

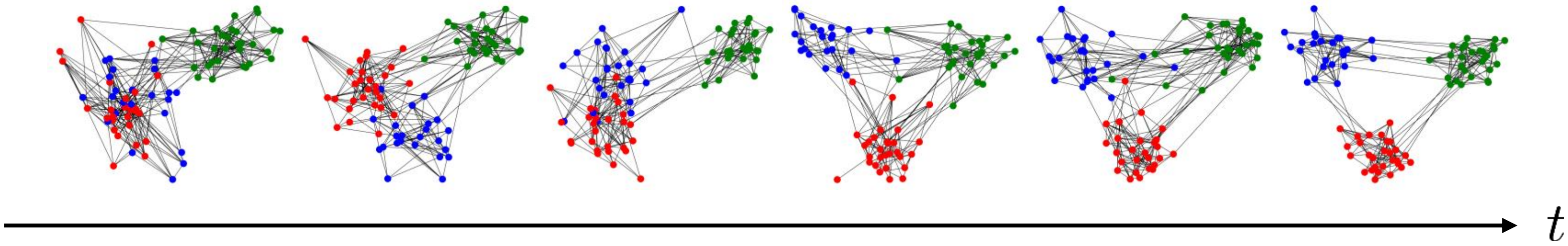
Simple(st): **Smoothing of adjacency matrix + SC**

- Uniform average ? *[Pensky 2017]*

$$\bar{A}_t = \sum_{l=t-w}^{t} A_l$$

    - *May need to keep a lot of past data in memory…*

Dynamic Spectral Clustering



Goal

Exploit past data to:
- Track communities
- Enforce smoothness/consistency
- **Improve result at time t**
  - *Does not want to apply SC several times !*

Many approaches :
- Incremental / hierarchical
- Maximum Likelihood / Bayesian
- Variational…

Simple(st): **Smoothing of adjacency matrix + SC**

- Uniform average ? *[Pensky 2017]*

$$\bar{A}_t = \sum_{l=t-w}^{t} A_l$$

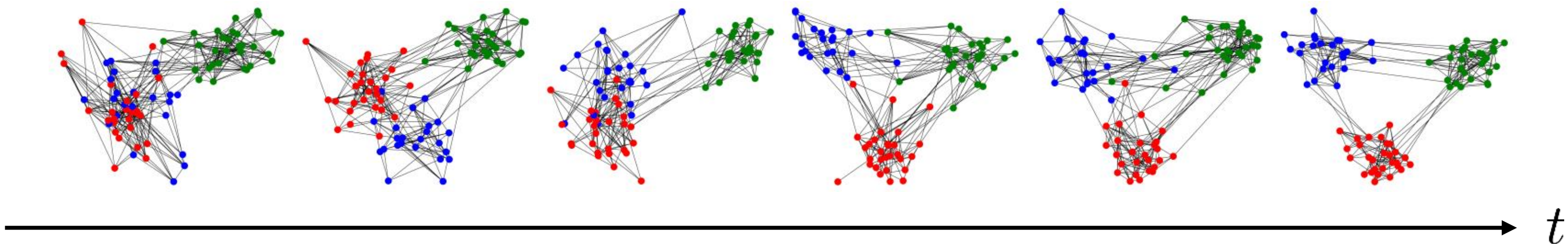
  - *May need to keep a lot of past data in memory…*

- Here : **Exponential Smoothing** *[Chi 2007, Xu 2010…]*

$$\bar{A}_t = (1-\lambda)\bar{A}_{t-1} + \lambda A_t$$

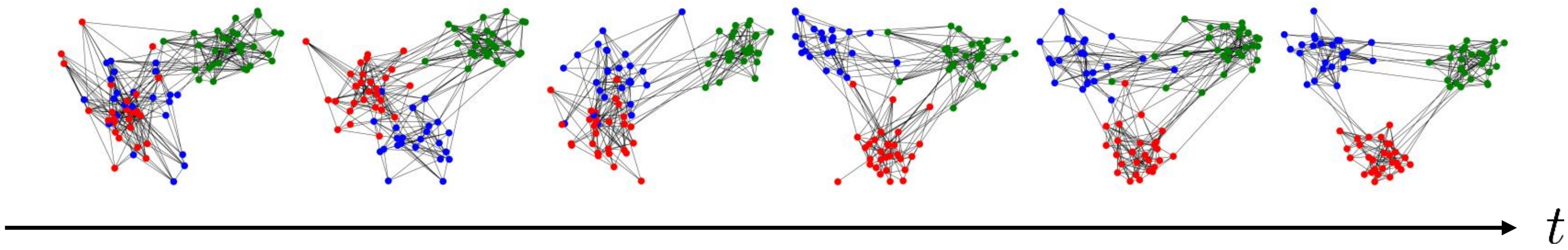
  - More appropriate for online computing

Dynamic SBM



Dynamic Stochastic Block Model (SBM)

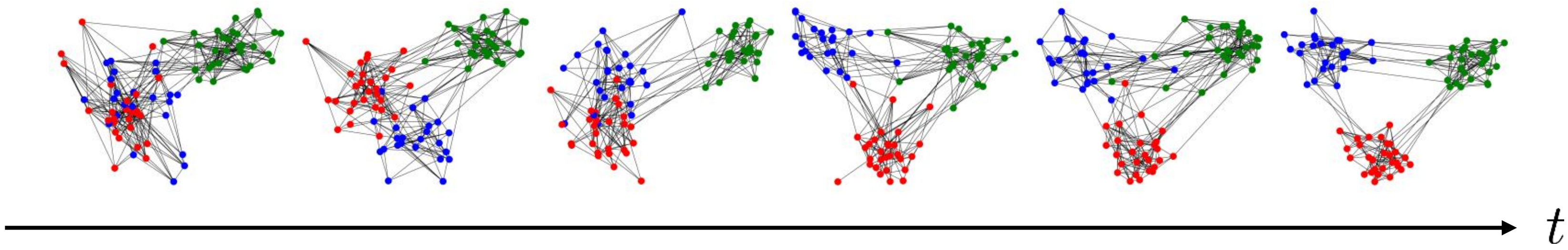
Dynamic SBM



Dynamic Stochastic Block Model (SBM)

Hidden Markov Model (HMM)

Dynamic SBM



$$\Theta_1 \longrightarrow \Theta_2 \longrightarrow \cdots \longrightarrow \Theta_t$$

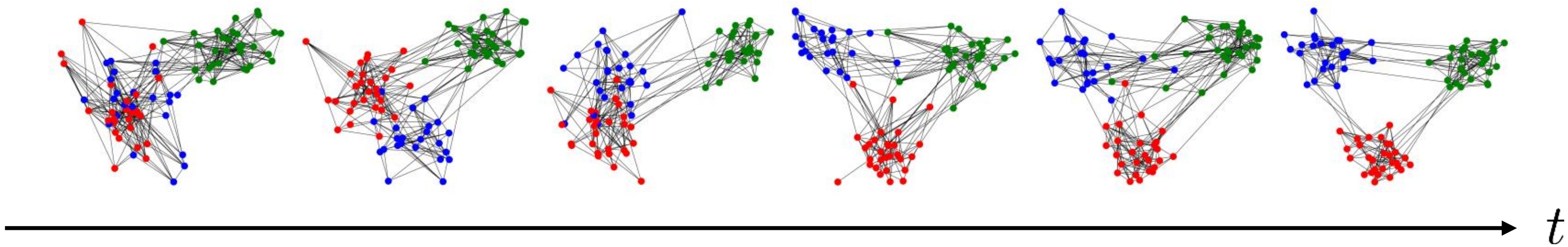
Dynamic Stochastic Block Model (SBM)

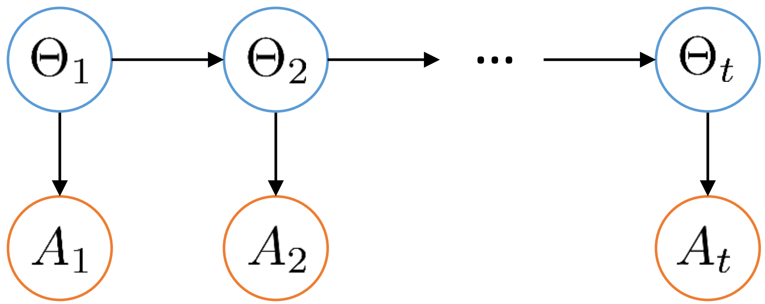
Hidden Markov Model (HMM)

- At each time step, each node change community with proba ε

$$\begin{cases} \mathbb{P}(\Theta_{ik}^{t} = 1 | \Theta_{ik}^{t-1} = 1) = 1 - \varepsilon \\ \mathbb{P}(\Theta_{ik}^{t} = 1 | \Theta_{i\ell}^{t-1} = 1) = \varepsilon / (K - 1) \end{cases}$$

Dynamic SBM





Dynamic Stochastic Block Model (SBM)

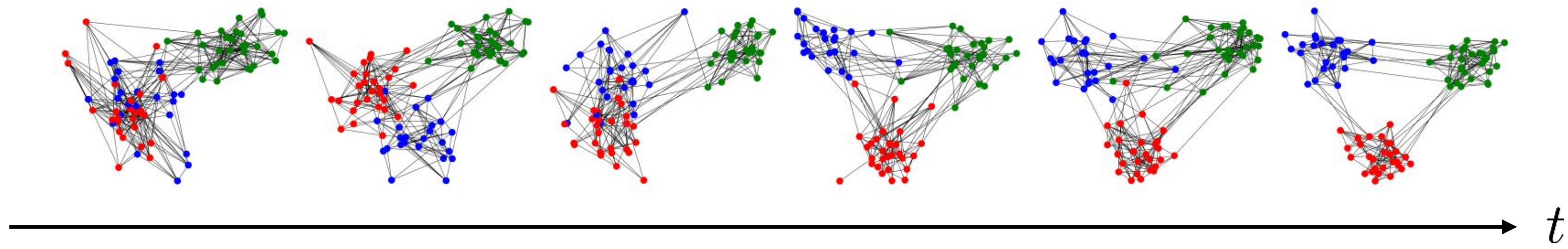
Hidden Markov Model (HMM)

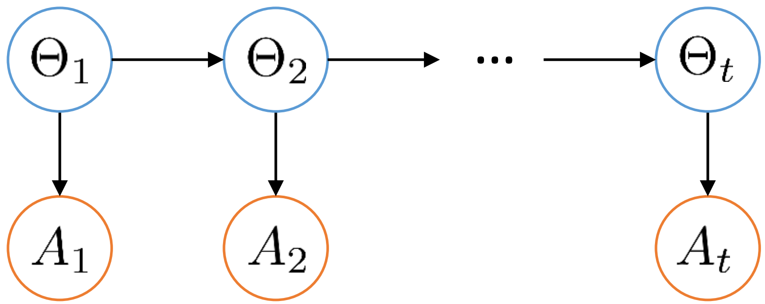
- At each time step, each node change community with proba ε
- Given cluster membership, SBM is generated

$$\begin{cases} \mathbb{P}(\Theta_{ik}^t = 1 | \Theta_{ik}^{t-1} = 1) = 1 - \varepsilon \\ \mathbb{P}(\Theta_{ik}^t = 1 | \Theta_{i\ell}^{t-1} = 1) = \varepsilon / (K-1) \end{cases}$$

$$\begin{cases} a_{ij}^t \sim \mathrm{Ber}(B_{k\ell}) \\ \text{when } \Theta_{ik}^t = 1, j \in \Theta_{j\ell}^t = 1 \end{cases}$$

Dynamic SBM





Dynamic Stochastic Block Model (SBM)

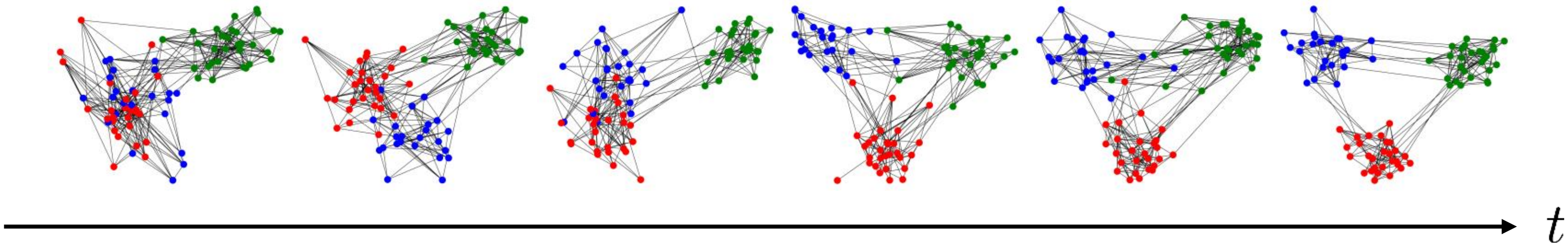
Hidden Markov Model (HMM)

- At each time step, each node change community with proba ε
- Given cluster membership, SBM is generated
- To simplify, connectivity matrix does not change

$$\begin{cases} \mathbb{P}(\Theta_{ik}^{t} = 1 | \Theta_{ik}^{t-1} = 1) = 1 - \varepsilon \\ \mathbb{P}(\Theta_{ik}^{t} = 1 | \Theta_{i\ell}^{t-1} = 1) = \varepsilon / (K-1) \end{cases}$$

$$\begin{cases} a_{ij}^{t} \sim \mathrm{Ber}(B_{k\ell}) \\ \text{when } \Theta_{ik}^{t} = 1, j \in \Theta_{j\ell}^{t} = 1 \end{cases}$$

Dynamic SBM

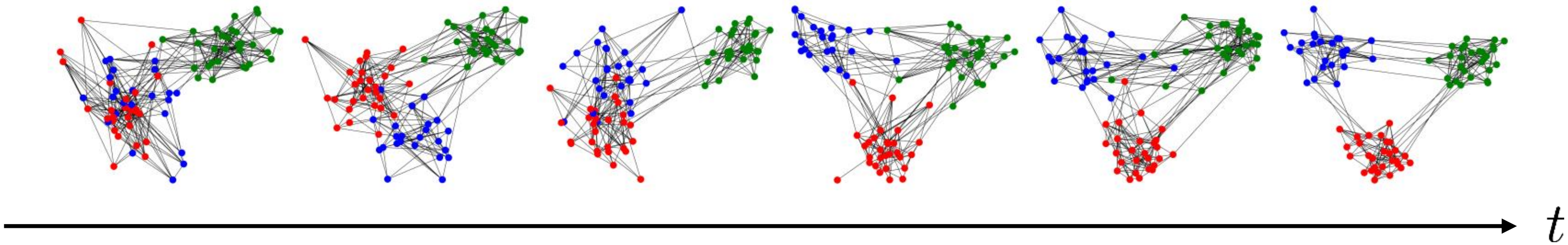


Uniform Average *[Pensky et al. 2017]*

- Uniform smoothing

$$\bar{A}_t = \sum_{l=t-w}^{t} A_l$$

Dynamic SBM

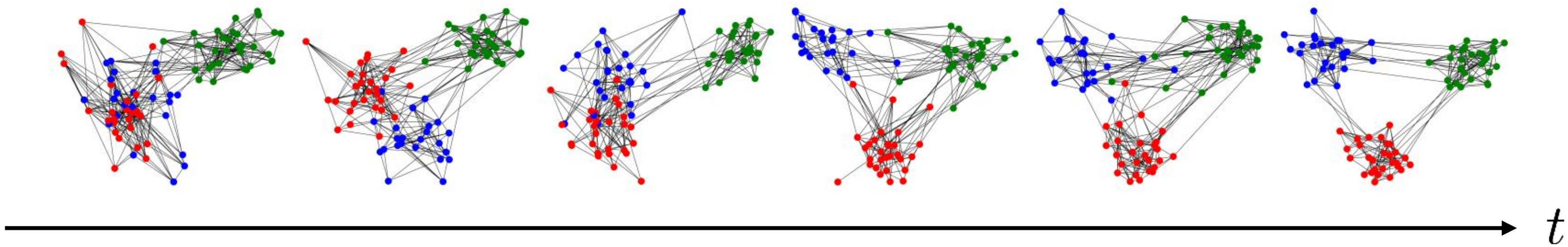


Uniform Average *[Pensky et al. 2017]*

- Uniform smoothing
- **Simpler model**, **deterministic communities**

$$\bar{A}_t = \sum_{l=t-w}^{t} A_l$$

Dynamic SBM

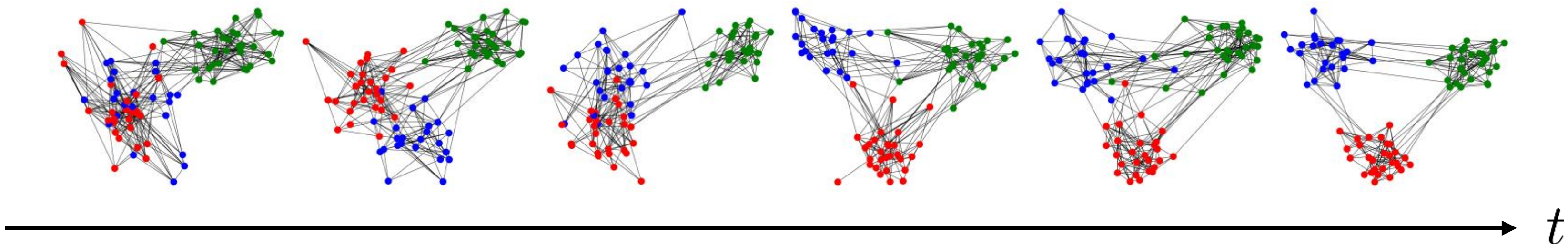


Uniform Average *[Pensky et al. 2017]*

- Uniform smoothing
- **Simpler model**, **deterministic communities**
  - at most s nodes change between time steps

$$\bar{A}_t = \sum_{l=t-w}^{t} A_l$$

Dynamic SBM



Uniform Average *[Pensky et al. 2017]*

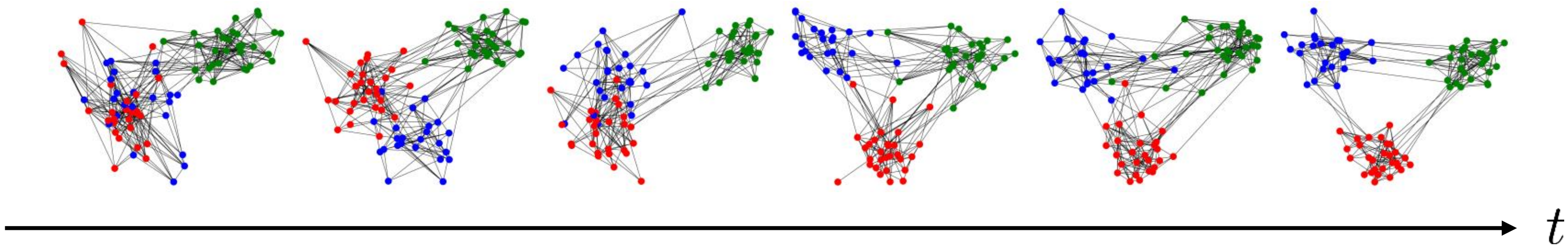
- Uniform smoothing
- **Simpler model**, **deterministic communities**
  - at most s nodes change between time steps

$$\bar{A}_t = \sum_{l=t-w}^{t} A_l$$

Under same hypotheses, ***with optimal window size****:*

$$L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n\alpha_n} \min\left(1, \sqrt{s\alpha_n}\right)$$

Dynamic SBM



Uniform Average *[Pensky et al. 2017]*

- Uniform smoothing
- **Simpler model**, **deterministic communities**
  - at most s nodes change between time steps

$$\bar{A}_t = \sum_{l=t-w}^{t} A_l$$

*Under same hypotheses, **with optimal window size**:*

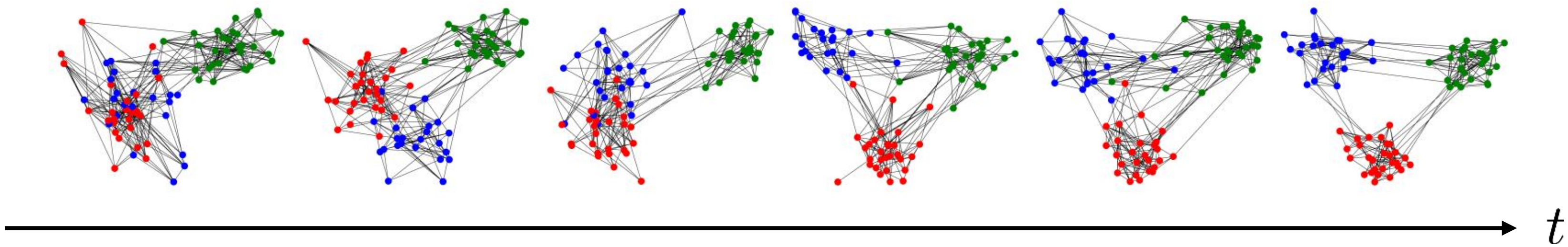
$$L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n\alpha_n} \min\left(1, \sqrt{s\alpha_n}\right)$$

Asymptotically better if:

$$\frac{s}{n} = o\left(\frac{1}{n\alpha_n}\right) = o\left(\frac{1}{\log n}\right)$$

« The more people (in each group), the less likely you are to change communities… »

Dynamic SBM



Uniform Average *[Pensky et al. 2017]*

- Uniform smoothing
- **Simpler model**, **deterministic communities**
  - at most s nodes change between time steps

$$\bar{A}_t = \sum_{l=t-w}^{t} A_l$$

Under same hypotheses, ***with optimal window size****:*

$$L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n\alpha_n} \min\left(1, \sqrt{s\alpha_n}\right)$$

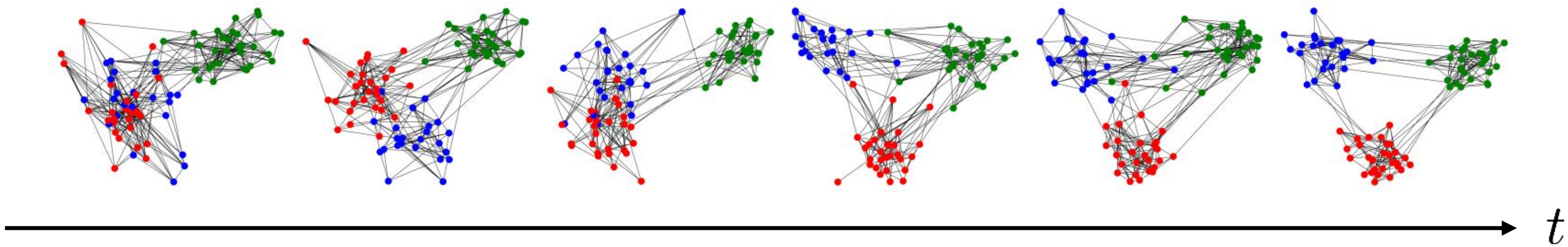
Asymptotically better if:

$$\frac{s}{n} = o\left(\frac{1}{n\alpha_n}\right) = o\left(\frac{1}{\log n}\right)$$

« The more people (in each group), the less likely you are to change communities… »

- **Choice of window size problematic !**
  - May necessitate to keep every data in memory…
  - The method indicated in *[Pensky2017]* does not work in practice !

Dynamic SBM



Uniform Average *[Pensky et al. 2017]*

- Uniform smoothing
- **Simpler model**, **deterministic communities**
  - at most s nodes change between time steps

$$\bar{A}_t = \sum_{l=t-w}^{t} A_l$$

*Under same hypotheses, **with optimal window size**:*

$$L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n\alpha_n} \min\left(1, \sqrt{s\alpha_n}\right)$$

Asymptotically better if:

$$\frac{s}{n} = o\left(\frac{1}{n\alpha_n}\right) = o\left(\frac{1}{\log n}\right)$$

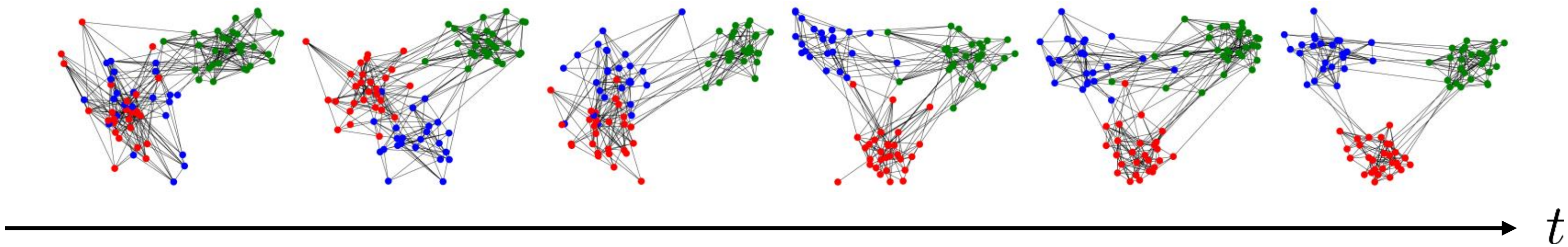
« The more people (in each group), the less likely you are to change communities… »

- **Choice of window size problematic !**
  - May necessitate to keep every data in memory…
  - The method indicated in *[Pensky2017]* does not work in practice !

Q

- How does the exponential model perform ? $\bar{A}_t = (1-\lambda)\bar{A}_{t-1} + \lambda A_t$

Dynamic SBM



Uniform Average *[Pensky et al. 2017]*

- Uniform smoothing
- **Simpler model**, **deterministic communities**
  - at most s nodes change between time steps

$$\bar{A}_t = \sum_{l=t-w}^{t} A_l$$

*Under same hypotheses, **with optimal window size**:*

$$L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n\alpha_n} \min\left(1, \sqrt{s\alpha_n}\right)$$

Asymptotically better if:

$$\frac{s}{n} = o\left(\frac{1}{n\alpha_n}\right) = o\left(\frac{1}{\log n}\right)$$

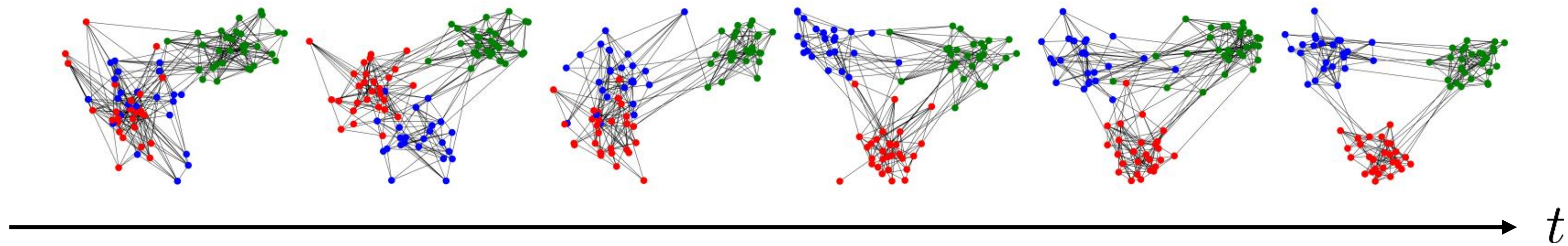
« The more people (in each group), the less likely you are to change communities… »

- **Choice of window size problematic !**
  - May necessitate to keep every data in memory…
  - The method indicated in *[Pensky2017]* does not work in practice !

Q

- How does the exponential model perform ? $\bar{A}_t = (1-\lambda)\bar{A}_{t-1} + \lambda A_t$
- Can we get the same / **improve** theoretical guarantees ? For HMM ?

Dynamic SBM



Uniform Average *[Pensky et al. 2017]*

- Uniform smoothing
- **Simpler model**, **deterministic communities**
  - at most s nodes change between time steps

$$\bar{A}_t = \sum_{l=t-w}^{t} A_l$$

Under same hypotheses, ***with optimal window size****:*

$$L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n\alpha_n} \min\left(1, \sqrt{s\alpha_n}\right)$$

Asymptotically better if:

$$\frac{s}{n} = o\left(\frac{1}{n\alpha_n}\right) = o\left(\frac{1}{\log n}\right)$$

« The more people (in each group), the less likely you are to change communities… »

- **Choice of window size problematic !**
  - May necessitate to keep every data in memory…
  - The method indicated in *[Pensky2017]* does not work in practice !

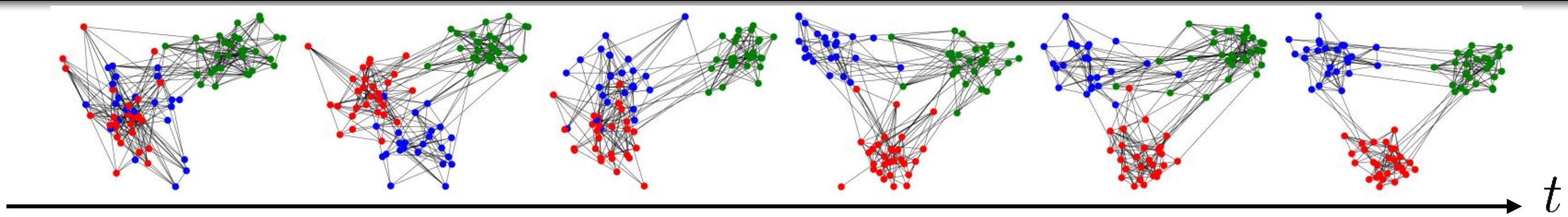
Q

- How does the exponential model perform ? $\bar{A}_t = (1-\lambda)\bar{A}_{t-1} + \lambda A_t$
- Can we get the same / **improve** theoretical guarantees ? For HMM ?
- Can we design an efficient way to select the forgetting factor ?

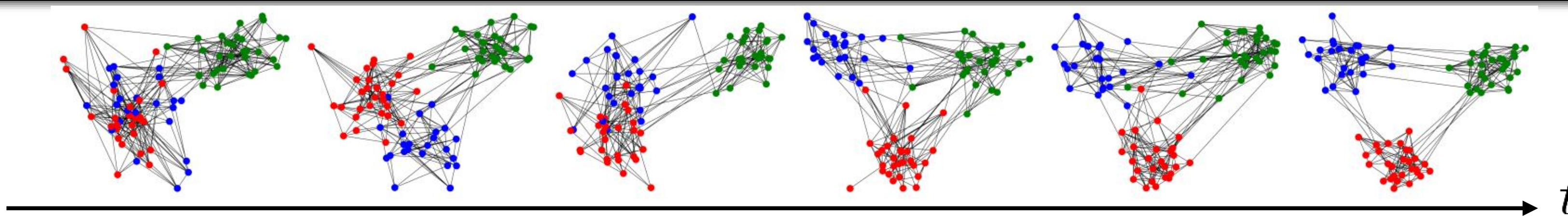
Outline

1. Main result
2. Choosing the forgetting factor
3. Experiments
4. Conclusion *(Bonus : GNN ?)*

Result

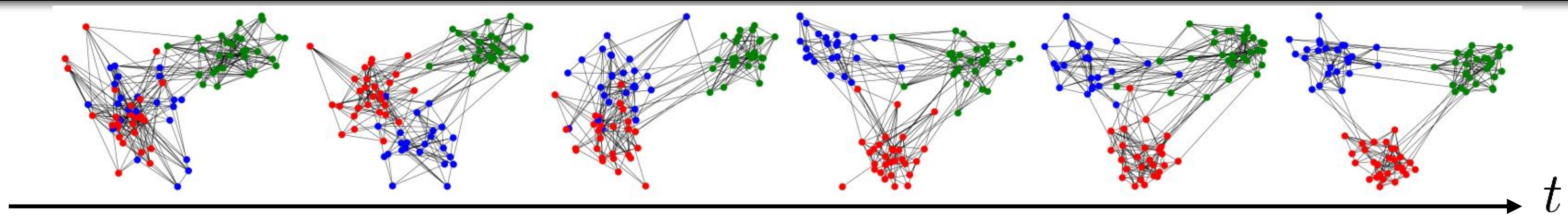


Result



$$\bar{A}_t = (1-\lambda)\bar{A}_{t-1} + \lambda A_t \longrightarrow \boxed{\textit{Your favorite (approx. ?) SC algo.}}$$

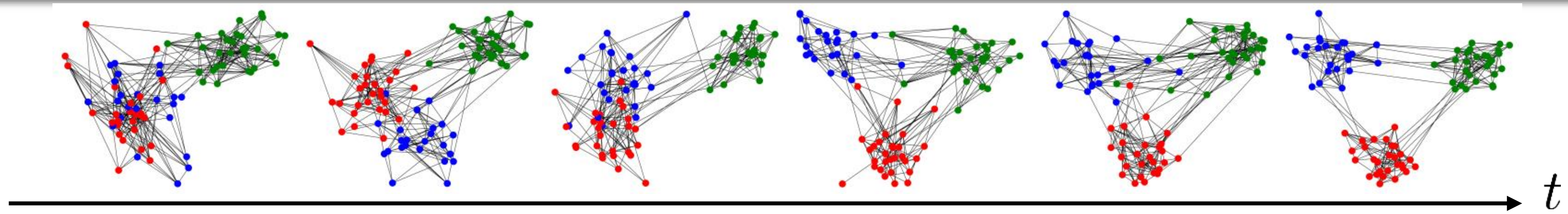
Result



$$\bar{A}_t = (1-\lambda)\bar{A}_{t-1} + \lambda A_t \longrightarrow$$ *Your favorite (approx. ?) SC algo.*

Theorem:

Result

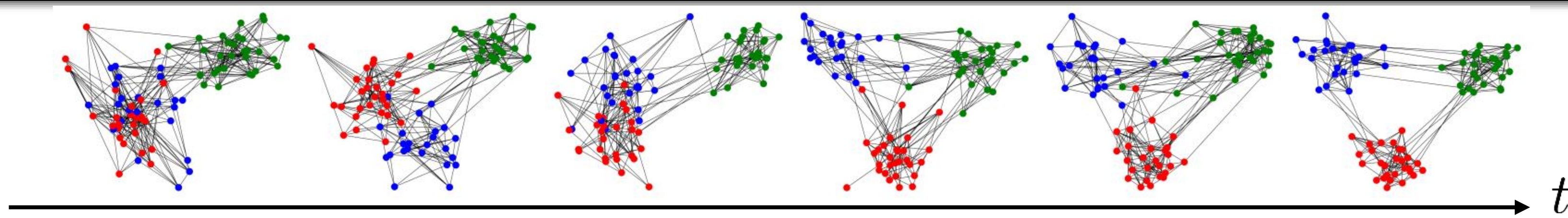


$\bar{A}_t = (1-\lambda)\bar{A}_{t-1} + \lambda A_t \longrightarrow$ *Your favorite (approx. ?) SC algo.*

Theorem:

Assume $t \geq t_{\min} = \frac{1}{2}\frac{\log(\alpha_n n/\lambda)}{\log(1/(1-\lambda))}$ (to reduce effect of initialization)

Result



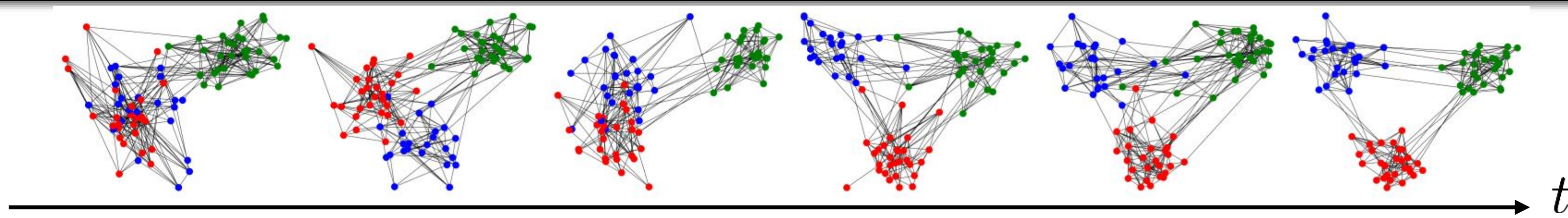
$$\bar{A}_t = (1-\lambda)\bar{A}_{t-1} + \lambda A_t \longrightarrow \boxed{\textit{Your favorite (approx. ?) SC algo.}}$$

Theorem:

Assume $t \geq t_{\min} = \frac{1}{2}\frac{\log(\alpha_n n/\lambda)}{\log(1/(1-\lambda))}$ (to reduce effect of initialization)

Choose $\lambda = \lambda^\star = \mathcal{O}(\min(1, \sqrt{n\alpha_n\varepsilon}))$

Result



$$\bar{A}_t = (1-\lambda)\bar{A}_{t-1} + \lambda A_t \longrightarrow$$ *Your favorite (approx. ?) SC algo.*

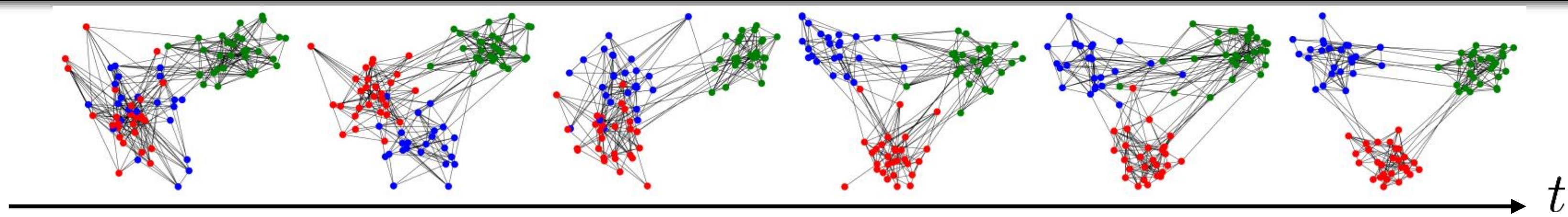
Theorem:

Assume $t \geq t_{\min} = \frac{1}{2}\frac{\log(\alpha_n n/\lambda)}{\log(1/(1-\lambda))}$ (to reduce effect of initialization)

Choose $\lambda = \lambda^\star = \mathcal{O}(\min(1, \sqrt{n\alpha_n\varepsilon}))$

If $\boxed{\frac{\alpha_n}{\lambda} \geq \frac{\log(n)}{n}}$, with probability $1 - n^{-r}$ on Θ, A :

Result



$\bar{A}_t = (1-\lambda)\bar{A}_{t-1} + \lambda A_t \longrightarrow$ *Your favorite (approx. ?) SC algo.*

Theorem:

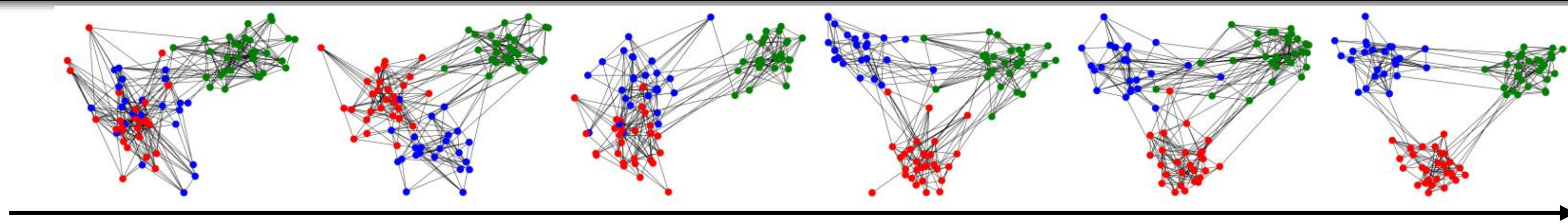
Assume $t \geq t_{\min} = \frac{1}{2}\frac{\log(\alpha_n n/\lambda)}{\log(1/(1-\lambda))}$ (to reduce effect of initialization)

Choose $\lambda = \lambda^\star = \mathcal{O}(\min(1, \sqrt{n\alpha_n\varepsilon}))$

If $\boxed{\frac{\alpha_n}{\lambda} \geq \frac{\log(n)}{n}}$, with probability $1 - n^{-r}$ on Θ, A :

$$\boxed{L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n\alpha_n}\min(1, \sqrt{n\alpha_n\varepsilon})}$$

Result



$$\bar{A}_t = (1-\lambda)\bar{A}_{t-1} + \lambda A_t \longrightarrow$$ *Your favorite (approx. ?) SC algo.*

Theorem:

Assume $t \geq t_{\min} = \frac{1}{2}\frac{\log(\alpha_n n/\lambda)}{\log(1/(1-\lambda))}$ (to reduce effect of initialization)

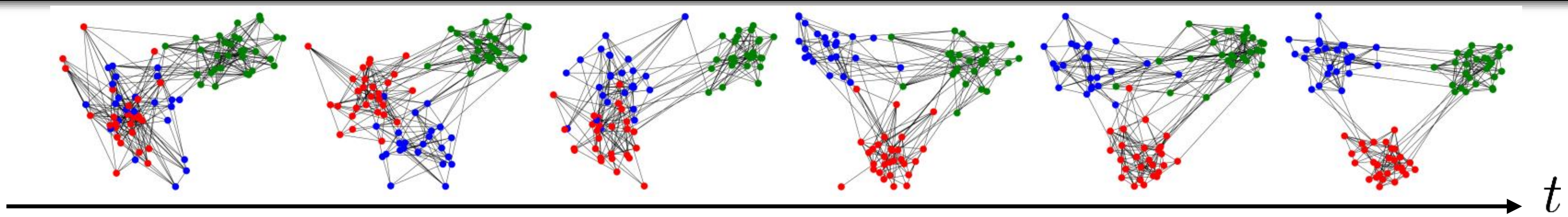
Choose $\lambda = \lambda^\star = \mathcal{O}(\min(1, \sqrt{n\alpha_n\varepsilon}))$

If $\boxed{\frac{\alpha_n}{\lambda} \geq \frac{\log(n)}{n}}$, with probability $1 - n^{-r}$ on Θ, A :

$$\boxed{L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n\alpha_n}\min(1, \sqrt{n\alpha_n\varepsilon})}$$

- Same rate as *[Pensky 2017]* with $\varepsilon = s/n$

Result



$\bar{A}_t = (1-\lambda)\bar{A}_{t-1} + \lambda A_t \longrightarrow$ *Your favorite (approx. ?) SC algo.*

Theorem:

Assume $t \geq t_{\min} = \frac{1}{2}\frac{\log(\alpha_n n/\lambda)}{\log(1/(1-\lambda))}$ (to reduce effect of initialization)

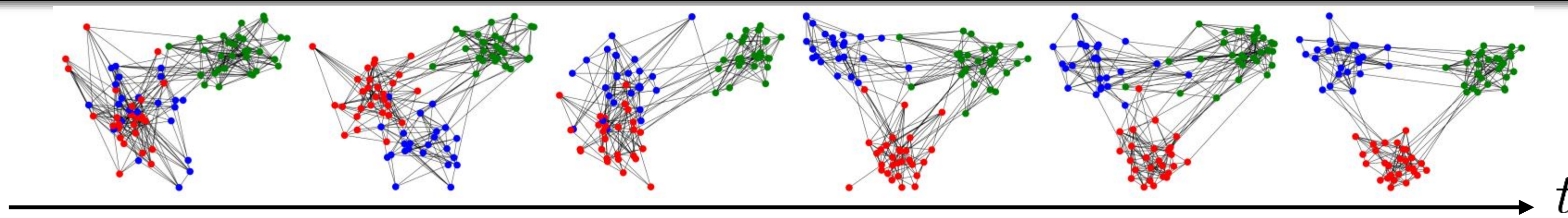
Choose $\lambda = \lambda^\star = \mathcal{O}(\min(1, \sqrt{n\alpha_n\varepsilon}))$

If $\boxed{\frac{\alpha_n}{\lambda} \geq \frac{\log(n)}{n}}$, with probability $1 - n^{-r}$ on Θ, A :

$$\boxed{L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n\alpha_n}\min(1, \sqrt{n\alpha_n\varepsilon})}$$

- Same rate as *[Pensky 2017]* with $\varepsilon = s/n$
- **Can handle the sparse case** if $\varepsilon = \mathcal{O}\left(\frac{1}{\log(n)^2}\right)$ *(which was already assumed in [Pensky2017] !)*

Result



$$\bar{A}_t = (1-\lambda)\bar{A}_{t-1} + \lambda A_t \longrightarrow$$ *Your favorite (approx. ?) SC algo.*

Theorem:

Assume $t \geq t_{\min} = \frac{1}{2}\frac{\log(\alpha_n n/\lambda)}{\log(1/(1-\lambda))}$ (to reduce effect of initialization)

Choose $\lambda = \lambda^\star = \mathcal{O}(\min(1, \sqrt{n\alpha_n\varepsilon}))$

If $\boxed{\frac{\alpha_n}{\lambda} \geq \frac{\log(n)}{n}}$, with probability $1 - n^{-r}$ on Θ, A :

$$\boxed{L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n\alpha_n}\min(1, \sqrt{n\alpha_n\varepsilon})}$$

- Same rate as *[Pensky 2017]* with $\varepsilon = s/n$
- **Can handle the sparse case** if $\varepsilon = \mathcal{O}\left(\frac{1}{\log(n)^2}\right)$ *(which was already assumed in [Pensky2017] !)*
- *Can « zero-out » the elements of* $\bar{A}_t$ *that are* $\leq (1-\lambda)^{t_{\min}}$ *, to keep it « sparse »*

Sketch of proof (1)

Step 1 *[Lei 2015]* :

Sketch of proof (1)

Step 1 *[Lei 2015]* :

- Define $P_t = \Theta_t B \Theta_t^\top$ (probability of connection between every two nodes)

Sketch of proof (1)

Step 1 *[Lei 2015]* :

- Define $P_t = \Theta_t B \Theta_t^\top$ (probability of connection between every two nodes)

- Apply perturbation theory (Davis-Kahan thm.) to show that for any matrix W used for spectral clustering, $L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n^2 \alpha_n^2} \|W - P_t\|^2$

Sketch of proof (1)

Step 1 *[Lei 2015]* :

- Define $P_t = \Theta_t B \Theta_t^\top$ (probability of connection between every two nodes)

- Apply perturbation theory (Davis-Kahan thm.) to show that for any matrix W used for spectral clustering, $L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n^2 \alpha_n^2} \|W - P_t\|^2$

Step 2 : bound $\|\bar{A}_t - P_t\|$

Sketch of proof (1)

Step 1 *[Lei 2015]* :

- Define $P_t = \Theta_t B \Theta_t^\top$ (probability of connection between every two nodes)

- Apply perturbation theory (Davis-Kahan thm.) to show that for any matrix W used for spectral clustering, $L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n^2 \alpha_n^2} \|W - P_t\|^2$

Step 2 : bound $\|\bar{A}_t - P_t\|$

- *[Lei 2015]* : $\|A_t - P_t\| \leq \mathcal{O}(\sqrt{n\alpha_n})$

Sketch of proof (1)

Step 1 *[Lei 2015]* :

- Define $P_t = \Theta_t B \Theta_t^\top$ (probability of connection between every two nodes)

- Apply perturbation theory (Davis-Kahan thm.) to show that for any matrix W used for spectral clustering, $L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n^2 \alpha_n^2} \|W - P_t\|^2$

Step 2 : bound $\|\bar{A}_t - P_t\|$

- *[Lei 2015]* : $\|A_t - P_t\| \leq \mathcal{O}(\sqrt{n\alpha_n})$

- *[Pensky 2017]*: $\|\bar{A}_t - P_t\| \leq \mathcal{O}(\sqrt{n\alpha_n} \min(1, (\alpha_n s)^{1/4}))$ *(for ideal window size)*

Sketch of proof (1)

Step 1 *[Lei 2015]* :

- Define $P_t = \Theta_t B \Theta_t^\top$ (probability of connection between every two nodes)

- Apply perturbation theory (Davis-Kahan thm.) to show that for any matrix W used for spectral clustering, $L(\hat{\Theta}^t, \Theta^t) \lesssim \frac{K^2}{n^2 \alpha_n^2} \|W - P_t\|^2$

Step 2 : bound $\|\bar{A}_t - P_t\|$

- *[Lei 2015]* : $\|A_t - P_t\| \leq \mathcal{O}(\sqrt{n\alpha_n})$

- *[Pensky 2017]*: $\|\bar{A}_t - P_t\| \leq \mathcal{O}(\sqrt{n\alpha_n}\min(1, (\alpha_n s)^{1/4}))$ *(for ideal window size)*

- Us: $\|\bar{A}_t - P_t\| \leq \mathcal{O}(\sqrt{n\alpha_n}\min(1, (n\alpha_n \varepsilon)^{1/4}))$ *(for ideal* λ *)*

Sketch of proof (2)

How to bound $\|\bar{A}_t - P_t\|$?

Sketch of proof (2)

How to bound $\|\bar{A}_t - P_t\|$?

- Decompose $\|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\|$ where $\bar{P}_t = (1-\lambda)\bar{P}_{t-1} + \lambda P_t = \mathbb{E}(\bar{A}_t)$

Sketch of proof (2)

How to bound $\|\bar{A}_t - P_t\|$?

- Decompose $\|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\|$ where $\bar{P}_t = (1-\lambda)\bar{P}_{t-1} + \lambda P_t = \mathbb{E}(\bar{A}_t)$

Advanced matrix concentration inequality, similar to *[Lei 2015]*

Sketch of proof (2)

How to bound $\|\bar{A}_t - P_t\|$?

- Decompose $\|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\|$ where $\bar{P}_t = (1-\lambda)\bar{P}_{t-1} + \lambda P_t = \mathbb{E}(\bar{A}_t)$

Advanced matrix concentration inequality, similar to *[Lei 2015]*

$$\|\bar{A}_t - \bar{P}_t\| \leq \delta_1(\lambda) := C_1\sqrt{n\alpha_n\lambda}$$

Sketch of proof (2)

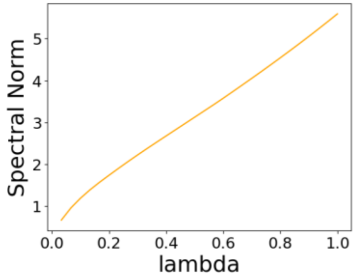
How to bound $\|\bar{A}_t - P_t\|$?

- Decompose $\|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\|$ where $\bar{P}_t = (1-\lambda)\bar{P}_{t-1} + \lambda P_t = \mathbb{E}(\bar{A}_t)$

Advanced matrix concentration inequality, similar to *[Lei 2015]*

$$\|\bar{A}_t - \bar{P}_t\| \leq \delta_1(\lambda) := C_1\sqrt{n\alpha_n\lambda}$$

$\boxed{\searrow \text{ when } \lambda \searrow}$



Sketch of proof (2)

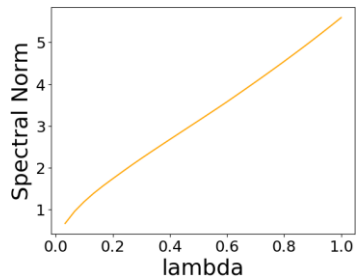
How to bound $\|\bar{A}_t - P_t\|$ **?**

- Decompose $\|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\|$ where $\bar{P}_t = (1-\lambda)\bar{P}_{t-1} + \lambda P_t = \mathbb{E}(\bar{A}_t)$

Advanced matrix concentration inequality, similar to *[Lei 2015]*

$$\|\bar{A}_t - \bar{P}_t\| \leq \delta_1(\lambda) := C_1\sqrt{n\alpha_n\lambda}$$

$\searrow$ when $\lambda \searrow$



Only depends on the Markov chain Θ

$$\|\bar{P}_t - P_t\| \leq \delta_2(\lambda) := C_2 n\alpha_n\sqrt{\frac{\varepsilon}{\lambda}}$$

Sketch of proof (2)

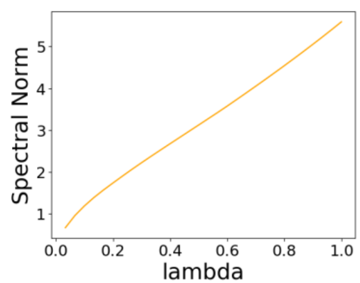
How to bound $\|\bar{A}_t - P_t\|$?

- Decompose $\|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\|$ where $\bar{P}_t = (1-\lambda)\bar{P}_{t-1} + \lambda P_t = \mathbb{E}(\bar{A}_t)$

Advanced matrix concentration inequality, similar to *[Lei 2015]*

$$\|\bar{A}_t - \bar{P}_t\| \leq \delta_1(\lambda) := C_1\sqrt{n\alpha_n\lambda}$$

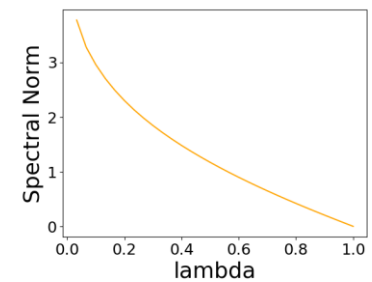
$$\boxed{\searrow \text{ when } \lambda \searrow}$$



Only depends on the Markov chain Θ

$$\|\bar{P}_t - P_t\| \leq \delta_2(\lambda) := C_2 n\alpha_n\sqrt{\frac{\varepsilon}{\lambda}}$$

$$\boxed{\nearrow \text{ when } \lambda \searrow}$$



Sketch of proof (2)

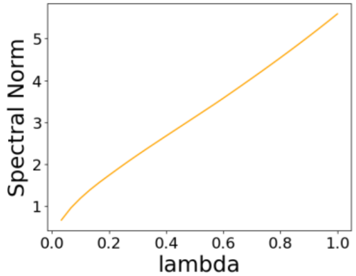
How to bound $\|\bar{A}_t - P_t\|$?

- Decompose $\|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\|$ where $\bar{P}_t = (1-\lambda)\bar{P}_{t-1} + \lambda P_t = \mathbb{E}(\bar{A}_t)$

Advanced matrix concentration inequality, similar to *[Lei 2015]*

$$\|\bar{A}_t - \bar{P}_t\| \leq \delta_1(\lambda) := C_1\sqrt{n\alpha_n\lambda}$$

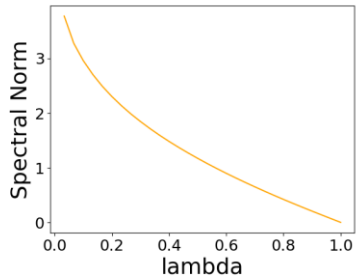
$$\boxed{\searrow \text{ when } \lambda \searrow}$$



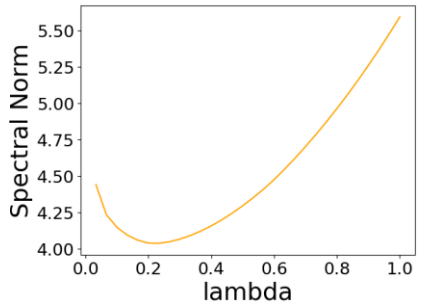
Only depends on the Markov chain Θ

$$\|\bar{P}_t - P_t\| \leq \delta_2(\lambda) := C_2 n\alpha_n\sqrt{\frac{\varepsilon}{\lambda}}$$

$$\boxed{\nearrow \text{ when } \lambda \searrow}$$



$$\boxed{\lambda^\star = \arg\min(\delta_1(\lambda) + \delta_2(\lambda)) = C_2 C_1^{-1}\sqrt{n\alpha_n\varepsilon}}$$



Sketch of proof (3)

Lei's concentration inequality for (sum of) Bernoulli matrices

$\mathcal{G}$

Sketch of proof (3)

Lei's concentration inequality for (sum of) Bernoulli matrices

- **Step 1**: write $\|A - P\| = \max_{x \in \mathcal{S}} x^\top (A - P) x$

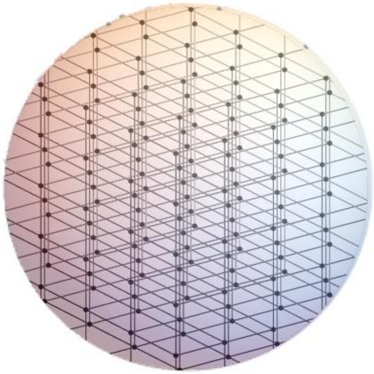
$\mathcal{G}$

Sketch of proof (3)

Lei's concerntration inequality for (sum of) Bernoulli matrices

- **Step 1**: write $\|A-P\| = \max_{x \in \mathcal{S}} x^\top (A-P)x \approx \max_{x,y \in \mathcal{G}} x^\top (A-P)y$

$\mathcal{G}$: *Appropriate **grid***

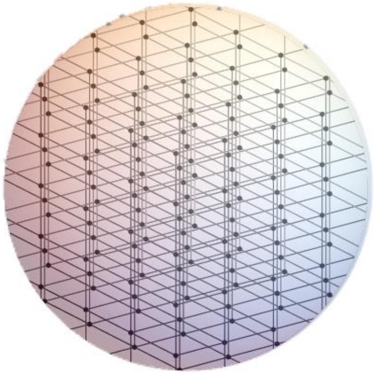


Sketch of proof (3)

Lei's concentration inequality for (sum of) Bernoulli matrices

- **Step 1**: write $\|A-P\| = \max_{x\in\mathcal{S}} x^\top (A-P)x \approx \max_{x,y\in\mathcal{G}} x^\top (A-P)y$

$\mathcal{G}$ *: Appropriate* ***grid***



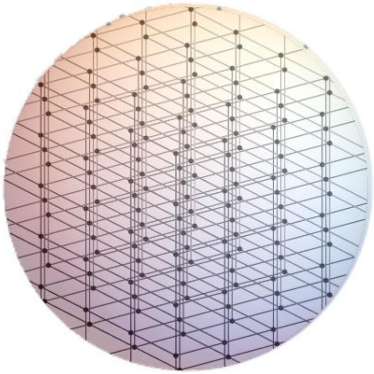
- **Observation:** $y^\top (A-P)x = \sum_{i,j} x_i y_j (a_{ij}-p_{ij})$ but directly applying Bernstein does not work !

Sketch of proof (3)

Lei's concentration inequality for (sum of) Bernoulli matrices

- **Step 1**: write $\|A - P\| = \max_{x \in \mathcal{S}} x^\top (A-P) x \approx \max_{x,y \in \mathcal{G}} x^\top (A-P) y$

$\mathcal{G}$: *Appropriate **grid***



  - **Observation:** $y^\top (A-P) x = \sum_{i,j} x_i y_j (a_{ij} - p_{ij})$ but directly applying Bernstein does not work !

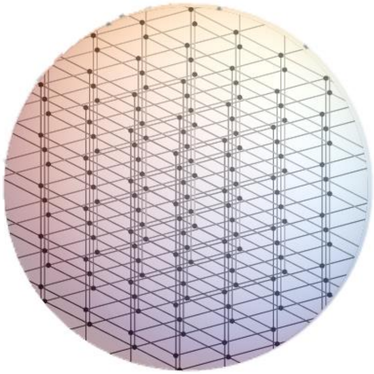
- **Step 2**: decompose $\sum_{i,j} x_i y_j (a_{ij} - p_{ij}) = \sum_{x_i y_j \text{ small}} x_i y_j (a_{ij} - p_{ij}) + \sum_{x_i y_j \text{ large}} x_i y_j (a_{ij} - p_{ij})$

Sketch of proof (3)

Lei's concentration inequality for (sum of) Bernoulli matrices

- **Step 1**: write $\|A-P\| = \max_{x\in\mathcal{S}} x^\top (A-P)x \approx \max_{x,y\in\mathcal{G}} x^\top (A-P)y$

$\mathcal{G}$: *Appropriate* ***grid***



  - **Observation:** $y^\top (A-P)x = \sum_{i,j} x_i y_j (a_{ij}-p_{ij})$ but directly applying Bernstein does not work !

- **Step 2**: decompose $\sum_{i,j} x_i y_j (a_{ij}-p_{ij}) = \sum_{x_i y_j \text{ small}} x_i y_j (a_{ij}-p_{ij}) + \sum_{x_i y_j \text{ large}} x_i y_j (a_{ij}-p_{ij})$

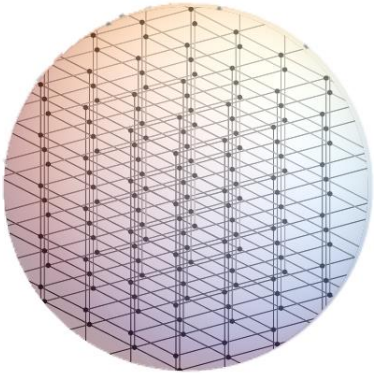
Bernstein ok

Sketch of proof (3)

Lei's concentration inequality for (sum of) Bernoulli matrices

- **Step 1**: write $\|A-P\| = \max_{x\in\mathcal{S}} x^\top(A-P)x \approx \max_{x,y\in\mathcal{G}} x^\top(A-P)y$

$\mathcal{G}$: *Appropriate* ***grid***



- **Observation:** $y^\top(A-P)x = \sum_{i,j} x_i y_j (a_{ij}-p_{ij})$ but directly applying Bernstein does not work !

- **Step 2**: decompose $\sum_{i,j} x_i y_j (a_{ij}-p_{ij}) = \sum_{x_i y_j \text{ small}} x_i y_j (a_{ij}-p_{ij}) + \sum_{x_i y_j \text{ large}} x_i y_j (a_{ij}-p_{ij})$

Bernstein ok

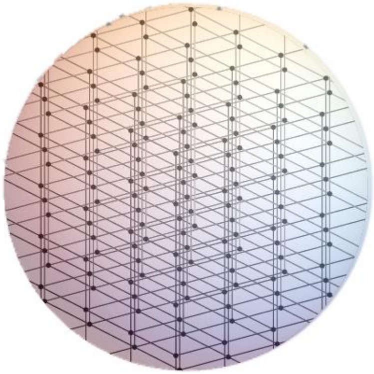
- Can be bounded *uniformly* for all $x, y \in \mathcal{G}$!

Sketch of proof (3)

Lei's concentration inequality for (sum of) Bernoulli matrices

- **Step 1**: write $\|A-P\| = \max_{x\in\mathcal{S}} x^\top (A-P)x \approx \max_{x,y\in\mathcal{G}} x^\top (A-P)y$

$\mathcal{G}$ *: Appropriate **grid***



  - **Observation:** $y^\top (A-P)x = \sum_{i,j} x_i y_j (a_{ij}-p_{ij})$ but directly applying Bernstein does not work !

- **Step 2**: decompose $\sum_{i,j} x_i y_j (a_{ij}-p_{ij}) = \sum_{x_i y_j \text{ small}} x_i y_j (a_{ij}-p_{ij}) + \sum_{x_i y_j \text{ large}} x_i y_j (a_{ij}-p_{ij})$

Bernstein ok

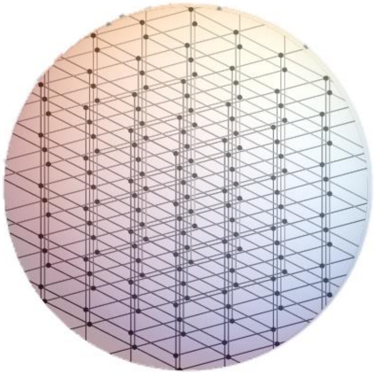
- Can be bounded *uniformly* for all $x, y \in \mathcal{G}$!
- Proof « specific » to Bernoulli matrices, not SBM
  - *Good chance that it could be further generalized*

Sketch of proof (3)

Lei's concentration inequality for (sum of) Bernoulli matrices

- **Step 1**: write $\|A-P\| = \max_{x\in\mathcal{S}} x^\top (A-P)x \approx \max_{x,y\in\mathcal{G}} x^\top (A-P)y$

$\mathcal{G}$: *Appropriate **grid***



  - **Observation:** $y^\top (A-P)x = \sum_{i,j} x_i y_j (a_{ij}-p_{ij})$ but directly applying Bernstein does not work !

- **Step 2**: decompose $\sum_{i,j} x_i y_j (a_{ij}-p_{ij}) = \sum_{x_i y_j \text{ small}} x_i y_j (a_{ij}-p_{ij}) + \sum_{x_i y_j \text{ large}} x_i y_j (a_{ij}-p_{ij})$

Bernstein ok

- Can be bounded *uniformly* for all $x, y \in \mathcal{G}$!
- Proof « specific » to Bernoulli matrices, not SBM
  - *Good chance that it could be further generalized*
- *Future work: other applications ?*

Outline

1. Main result
2. Choosing the forgetting factor
3. Experiments
4. Conclusion *(Bonus : GNN ?)*

Grid of forgetting factors

How to choose λ ?

Grid of forgetting factors

How to choose λ ?

- *[Pensky 2017]: how to choose window size ?*

Grid of forgetting factors

How to choose λ ?

- *[Pensky 2017]: how to choose window size ?*
  - keep all data in memory (offline)

Grid of forgetting factors

How to choose λ ?

- *[Pensky 2017]: how to choose window size ?*
  - keep all data in memory (offline)
  - choose window size *a posteriori* using **Lepski's method**
    - ***Does not work in practice !*** *(no numerics in [Pensky 2017])*

Grid of forgetting factors

How to choose λ ?

- *[Pensky 2017]: how to choose window size ?*
  - keep all data in memory (offline)
  - choose window size *a posteriori* using **Lepski's method**
    - ***Does not work in practice !*** *(no numerics in [Pensky 2017])*

- *[Xu 2010]:* perform a spectral clustering at each time step, choose an adaptive λ_t
    - *In our case, we do not want to perform SC at each time step !*

Grid of forgetting factors

How to choose λ ?

- *[Pensky 2017]: how to choose window size ?*
  - keep all data in memory (offline)
  - choose window size *a posteriori* using **Lepski's method**
    - ***Does not work in practice !*** *(no numerics in [Pensky 2017])*

- *[Xu 2010]:* perform a spectral clustering at each time step, choose an adaptive λ_t
    - *In our case, we do not want to perform SC at each time step !*

Proposed : maintain online several $\bar{A}_t(\lambda_1), \ldots, \bar{A}_t(\lambda_N)$, choose only when SC desired.

Grid of forgetting factors

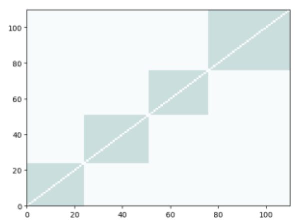
How to choose λ ?

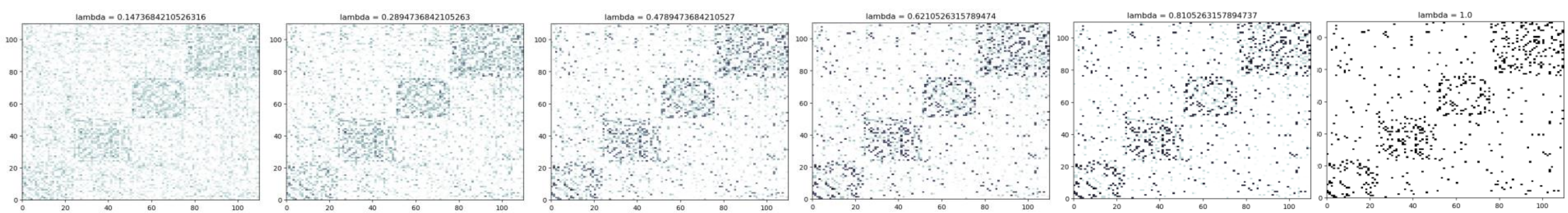
- *[Pensky 2017]: how to choose window size ?*
  - keep all data in memory (offline)
  - choose window size *a posteriori* using **Lepski's method**
    - ***Does not work in practice !*** *(no numerics in [Pensky 2017])*

- *[Xu 2010]:* perform a spectral clustering at each time step, choose an adaptive λ_t
  - *In our case, we do not want to perform SC at each time step !*

Proposed : maintain online several $\bar{A}_t(\lambda_1), \ldots, \bar{A}_t(\lambda_N)$, choose only when SC desired.

P_t





$\bar{A}_t(\lambda_1)$ $\bar{A}_t(\lambda_2)$ $\bar{A}_t(\lambda_N = 1)$

Grid of forgetting factors

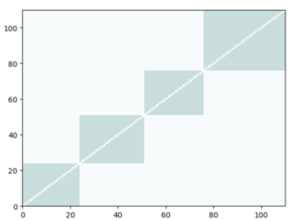
How to choose λ ?

- *[Pensky 2017]: how to choose window size ?*
    - keep all data in memory (offline)
    - choose window size *a posteriori* using **Lepski's method**
        - ***Does not work in practice !*** *(no numerics in [Pensky 2017])*

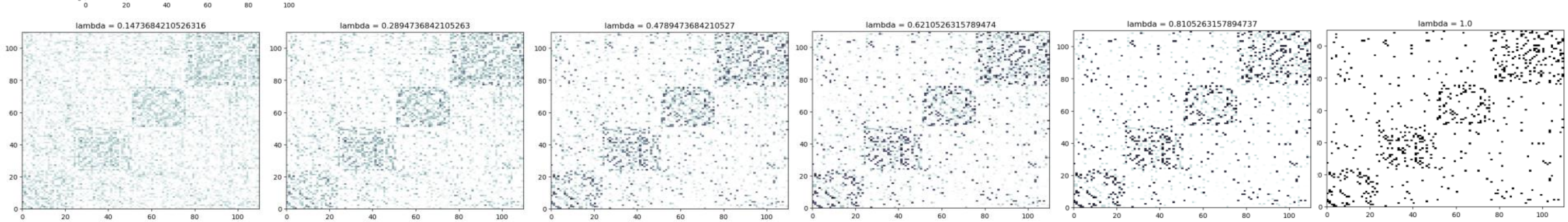
- *[Xu 2010]:* perform a spectral clustering at each time step, choose an adaptive λ_t
    - *In our case, we do not want to perform SC at each time step !*

Proposed : maintain online several $\bar{A}_t(\lambda_1), \ldots, \bar{A}_t(\lambda_N)$, choose only when SC desired.

P_t



- Possible to maintain strong smoothing (small forgetting factors) without additional computational load
- Does not necessitate access to raw past data



$\bar{A}_t(\lambda_1)$ $\bar{A}_t(\lambda_2)$ $\bar{A}_t(\lambda_N = 1)$

Method 1 : Lepski

Method 1 : Adaptation of Lepski's method

Method 1 : Lepski

Method 1 : Adaptation of Lepski's method

Lemma

Assume that $\sqrt{\lambda_i} - \sqrt{\lambda_{i-1}} \leq \gamma$ and α_n is known. Choose $\lambda_{\tilde{i}}$ such that

$$\lambda_{\tilde{i}} = \arg\min_{\lambda_i} \{\lambda_i : \forall i < j \leq N, \|\bar{A}_t(\lambda_i) - \bar{A}_t(\lambda_j)\| \leq 4\delta_1(\lambda_j)\}$$

Then with probability at least $1 - Nn^{-r}$,

$$\|\bar{A}_t(\lambda_{\tilde{i}}) - P_t\| \leq 6\delta^\star + 5\gamma\sqrt{n\alpha_n}$$

Method 1 : Lepski

Method 1 : Adaptation of Lepski's method

Lemma

Assume that $\sqrt{\lambda_i} - \sqrt{\lambda_{i-1}} \leq \gamma$ and α_n is known. Choose $\lambda_{\tilde{i}}$ such that

$$\lambda_{\tilde{i}} = \arg\min_{\lambda_i} \{\lambda_i : \forall i < j \leq N, \|\bar{A}_t(\lambda_i) - \bar{A}_t(\lambda_j)\| \leq 4\delta_1(\lambda_j)\}$$

Then with probability at least $1 - Nn^{-r}$,

$$\|\bar{A}_t(\lambda_{\tilde{i}}) - P_t\| \leq 6\delta^\star + 5\gamma\sqrt{n\alpha_n}$$

$\mathcal{O}(\delta^\star)$ if $\gamma = \mathcal{O}((n\alpha_n\varepsilon)^{1/4})$

$N = \mathcal{O}((n\alpha_n\varepsilon)^{-1/4})$

Method 1 : Lepski

Method 1 : Adaptation of Lepski's method

Lemma

Assume that $\sqrt{\lambda_i} - \sqrt{\lambda_{i-1}} \leq \gamma$ and α_n is known. Choose $\lambda_{\tilde{i}}$ such that

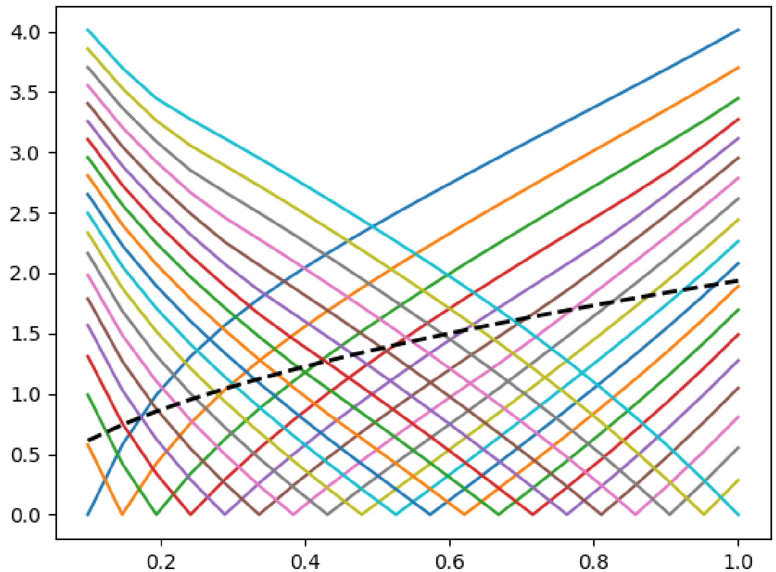
$$\lambda_{\tilde{i}} = \arg\min_{\lambda_i} \{\lambda_i : \forall i < j \leq N, \|\bar{A}_t(\lambda_i) - \bar{A}_t(\lambda_j)\| \leq 4\delta_1(\lambda_j)\}$$

Then with probability at least $1 - Nn^{-r}$,

$$\|\bar{A}_t(\lambda_{\tilde{i}}) - P_t\| \leq 6\delta^\star + 5\gamma\sqrt{n\alpha_n}$$

$\mathcal{O}(\delta^\star)$ if $\gamma = \mathcal{O}((n\alpha_n\varepsilon)^{1/4})$

$N = \mathcal{O}((n\alpha_n\varepsilon)^{-1/4})$



Method 1 : Lepski

Method 1 : Adaptation of Lepski's method

Lemma

Assume that $\sqrt{\lambda_i} - \sqrt{\lambda_{i-1}} \leq \gamma$ and α_n is known. Choose $\lambda_{\tilde{i}}$ such that

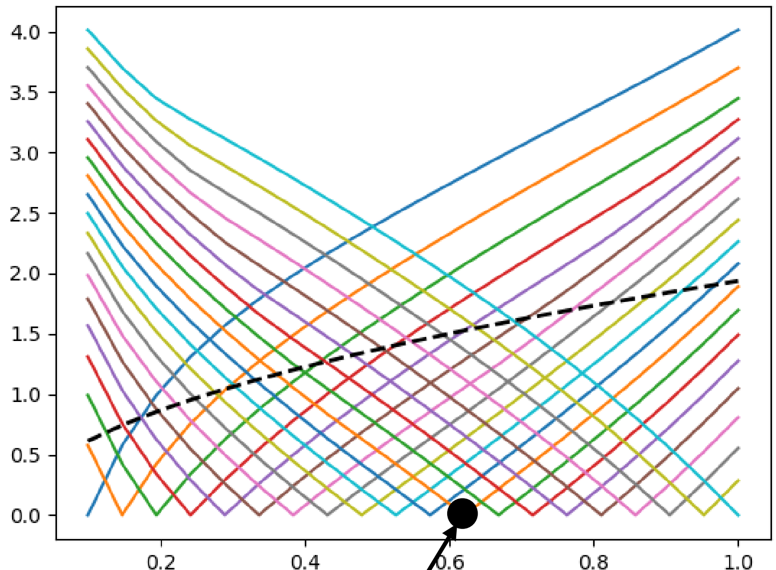
$$\lambda_{\tilde{i}} = \arg\min_{\lambda_i} \{\lambda_i : \forall i < j \leq N, \|\bar{A}_t(\lambda_i) - \bar{A}_t(\lambda_j)\| \leq 4\delta_1(\lambda_j)\}$$

Then with probability at least $1 - Nn^{-r}$,

$$\|\bar{A}_t(\lambda_{\tilde{i}}) - P_t\| \leq 6\delta^\star + 5\gamma\sqrt{n\alpha_n}$$

$\mathcal{O}(\delta^\star)$ if $\gamma = \mathcal{O}((n\alpha_n\varepsilon)^{1/4})$

$N = \mathcal{O}((n\alpha_n\varepsilon)^{-1/4})$



$\lambda_{\tilde{i}} \approx 0.6$

Method 1 : Lepski

Method 1 : Adaptation of Lepski's method

Lemma

Assume that $\sqrt{\lambda_i} - \sqrt{\lambda_{i-1}} \leq \gamma$ and α_n is known. Choose $\lambda_{\tilde{i}}$ such that

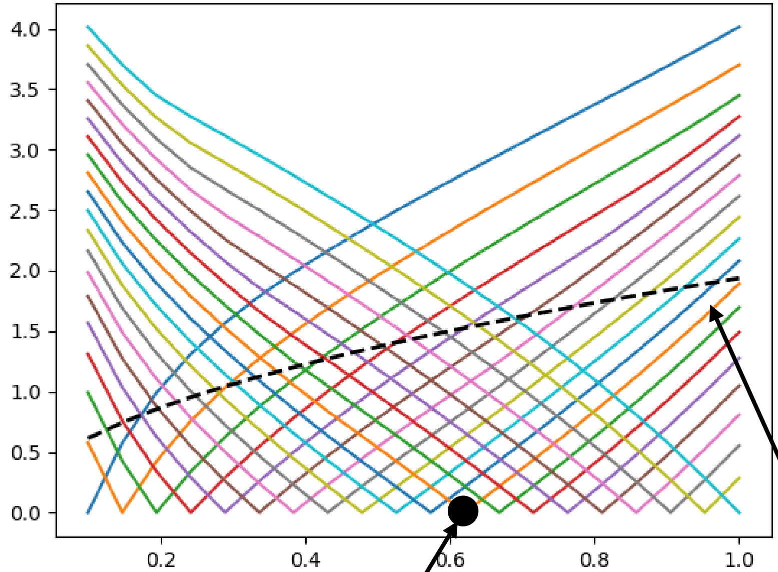
$$\lambda_{\tilde{i}} = \arg\min_{\lambda_i} \{\lambda_i : \forall i < j \leq N, \|\bar{A}_t(\lambda_i) - \bar{A}_t(\lambda_j)\| \leq 4\delta_1(\lambda_j)\}$$

Then with probability at least $1 - Nn^{-r}$,

$$\|\bar{A}_t(\lambda_{\tilde{i}}) - P_t\| \leq 6\delta^\star + 5\gamma\sqrt{n\alpha_n}$$

$\mathcal{O}(\delta^\star)$ if $\gamma = \mathcal{O}((n\alpha_n\varepsilon)^{1/4})$

$N = \mathcal{O}((n\alpha_n\varepsilon)^{-1/4})$



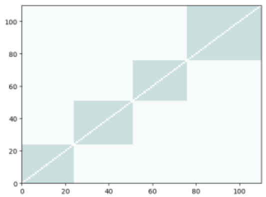
$\lambda_{\tilde{i}} \approx 0.6$

Problem :
The theoretical expression for $\delta_1(\lambda) = C_1\sqrt{n\alpha_n\lambda}$ is not tight ! Unusable in practice….

Here $C_1 = 0.5$ for illustrative purpose, in theory $C_1 \geq 289^2$! Many proof artifacts…

Method 2 : proxy for P

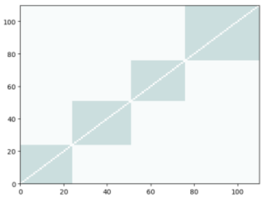
Method 2 : Proxy for P_t



Goal: minimize $\|\bar{A}_t(\lambda_i) - P_t\|$, but P_t unknown.

Method 2 : proxy for P

Method 2 : Proxy for P_t

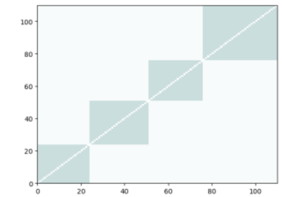


Goal: minimize $\|\bar{A}_t(\lambda_i) - P_t\|$, but P_t unknown.

Idea: Replace it with a proxy.

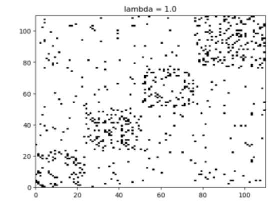
Method 2 : proxy for P

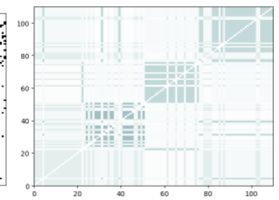
Method 2 : Proxy for P_t



Goal: minimize $\|\bar{A}_t(\lambda_i) - P_t\|$, but P_t unknown.

Idea: Replace it with a proxy.

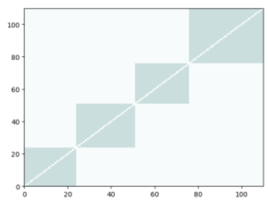




- Estimate $\hat{P}_t$ from A_t
  - ***(Spectral Clustering + maximum likelihood)***

Method 2 : proxy for P

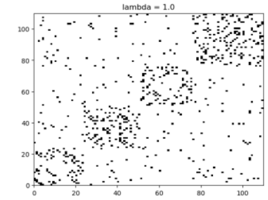
Method 2 : Proxy for P_t

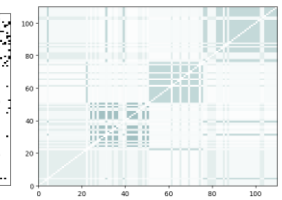


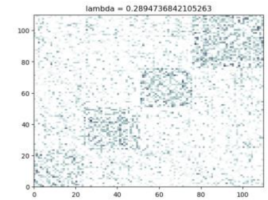
Goal: minimize $\|\bar{A}_t(\lambda_i) - P_t\|$, but P_t unknown.

Idea: Replace it with a proxy.

- Estimate $\hat{P}_t$ from A_t
    - ***(Spectral Clustering + maximum likelihood)***
- Minimize $\|\bar{A}_t(\lambda_i) - \hat{P}_t\|$

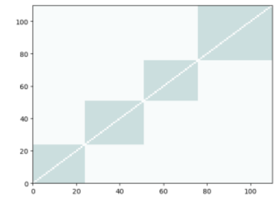






Method 2 : proxy for P

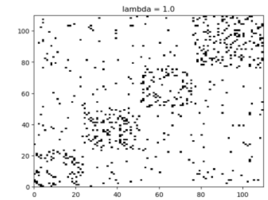
Method 2 : Proxy for P_t

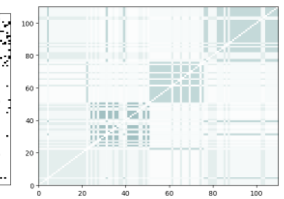


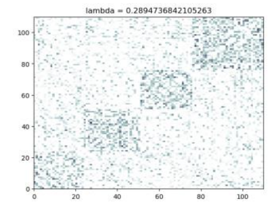
Goal: minimize $\|\bar{A}_t(\lambda_i) - P_t\|$, but P_t unknown.

Idea: Replace it with a proxy.

- Estimate $\hat{P}_t$ from A_t
    - ***(Spectral Clustering + maximum likelihood)***
- Minimize $\|\bar{A}_t(\lambda_i) - \hat{P}_t\|$
- Repeat with the new $\bar{A}_t(\lambda_i)$ and iterate ?

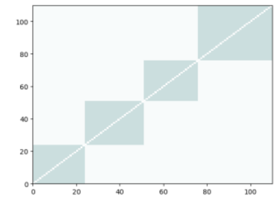






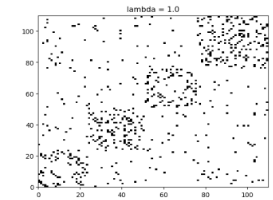
Method 2 : proxy for P

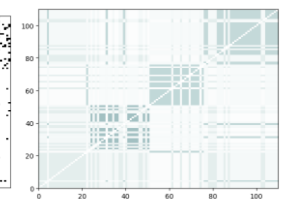
Method 2 : Proxy for P_t



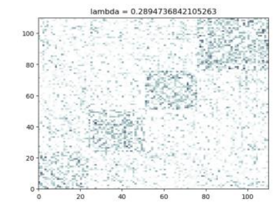
Goal: minimize $\|\bar{A}_t(\lambda_i) - P_t\|$, but P_t unknown.

Idea: Replace it with a proxy.

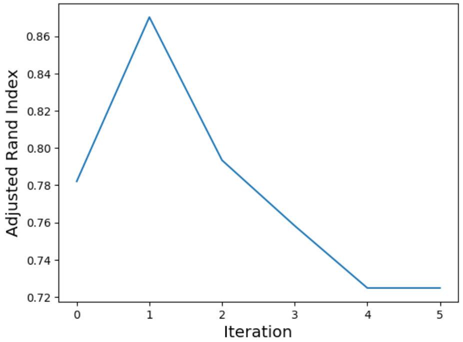




- Estimate $\hat{P}_t$ from A_t
    - ***(Spectral Clustering + maximum likelihood)***
- Minimize $\|\bar{A}_t(\lambda_i) - \hat{P}_t\|$
- Repeat with the new $\bar{A}_t(\lambda_i)$ and iterate ?

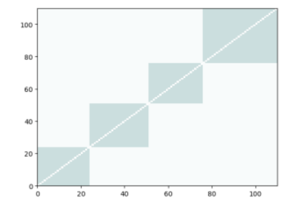


Usually diverge ! ☹



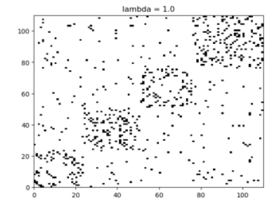
Method 2 : proxy for P

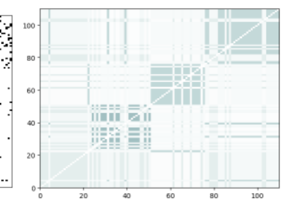
Method 2 : Proxy for P_t



Goal: minimize $\|\bar{A}_t(\lambda_i) - P_t\|$, but P_t unknown.

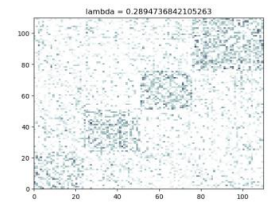
Idea: Replace it with a proxy.





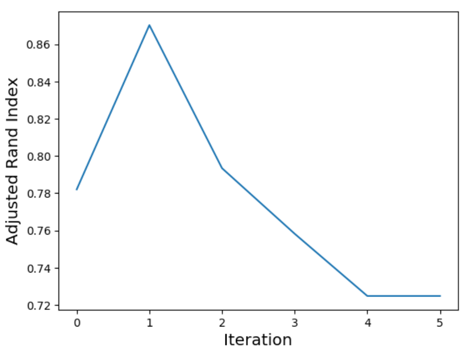
- Estimate $\hat{P}_t$ from A_t
  - ***(Spectral Clustering + maximum likelihood)***

- Minimize $\|\bar{A}_t(\lambda_i) - \hat{P}_t\|$

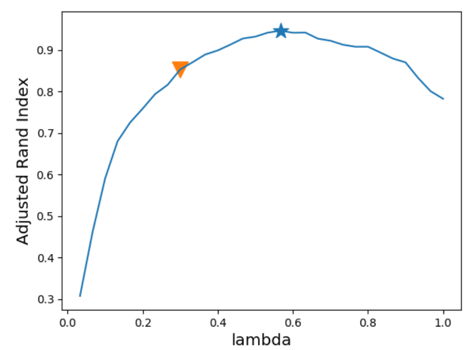


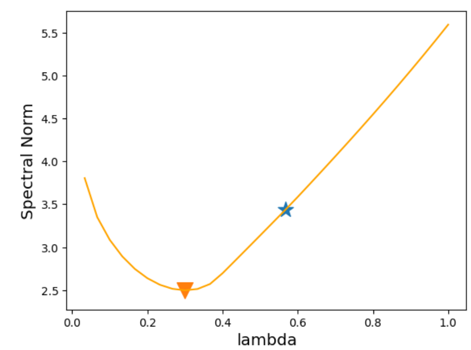
- Repeat with the new $\bar{A}_t(\lambda_i)$ and iterate ?

Usually diverge ! ☹



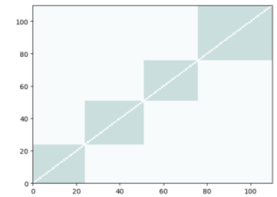
Explanation ? The spectral norm may not be the best criterion !





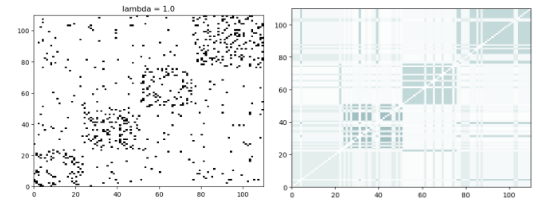
Method 2 : proxy for P

Method 2 : Proxy for P_t

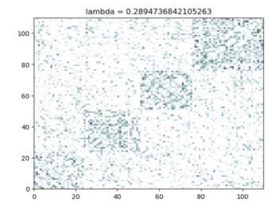


Goal: minimize $\|\bar{A}_t(\lambda_i) - P_t\|$, but P_t unknown.

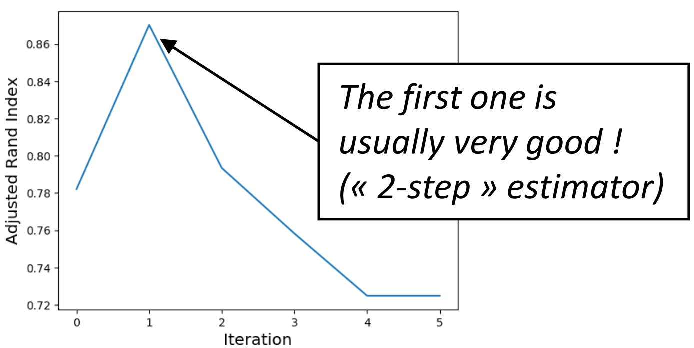
Idea: Replace it with a proxy.



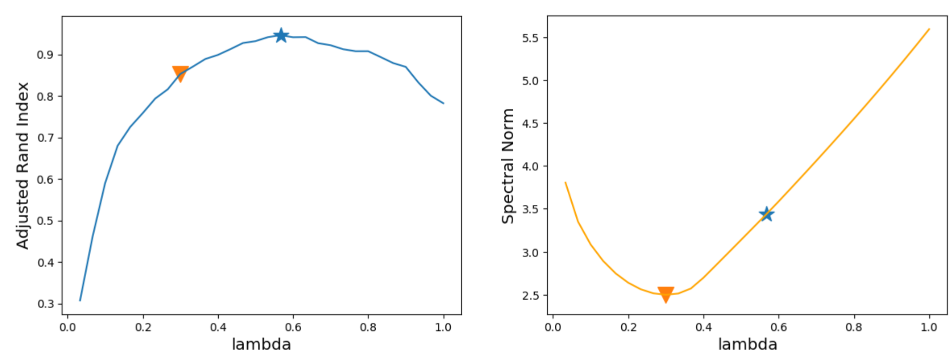
- Estimate $\hat{P}_t$ from A_t
  - ***(Spectral Clustering + maximum likelihood)***
- Minimize $\|\bar{A}_t(\lambda_i) - \hat{P}_t\|$
- Repeat with the new $\bar{A}_t(\lambda_i)$ and iterate ?



Usually diverge ! ☹

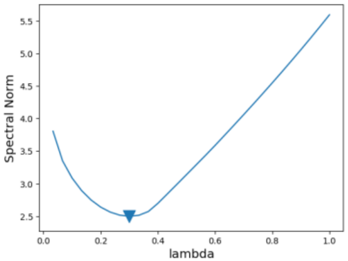


Explanation ? The spectral norm may not be the best criterion !



Method 3 : finite differences (in progress…)

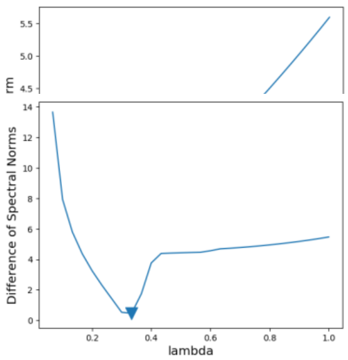
Method 3 : Derivative and finite differences



Observation : the function $f(\lambda_i) = \|\bar{A}_t(\lambda_i) - P_t\|$ that we are trying to minimize looks convex…

Method 3 : finite differences (in progress…)

Method 3 : Derivative and finite differences



Observation : the function $f(\lambda_i) = \|\bar{A}_t(\lambda_i) - P_t\|$ that we are trying to minimize looks convex…

Idea : minimize the derivative ? (still unknown !) $g(\lambda_i) = \frac{|f(\lambda_i) - f(\lambda_{i-1})|}{\lambda_i - \lambda_{i-1}}$

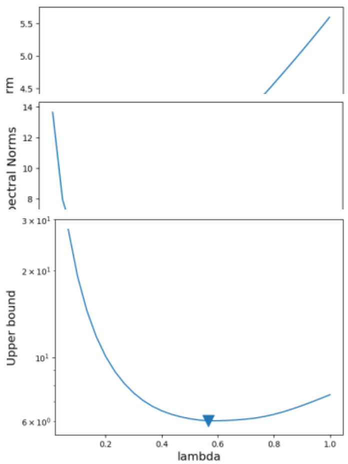
Method 3 : finite differences (in progress…)

Method 3 : Derivative and finite differences

Observation : the function $f(\lambda_i) = \|\bar{A}_t(\lambda_i) - P_t\|$ that we are trying to minimize looks convex…

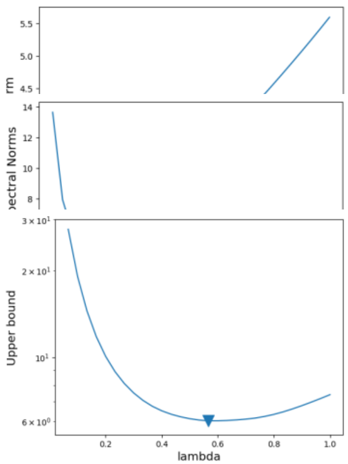
Idea : minimize the derivative ? (still unknown !) $g(\lambda_i) = \frac{|f(\lambda_i) - f(\lambda_{i-1})|}{\lambda_i - \lambda_{i-1}}$

Final procedure : minimize an upper bound $\boxed{h(\lambda_i) = \frac{\|\bar{A}_t(\lambda_i) - \bar{A}_t(\lambda_{i-1})\|}{\lambda_i - \lambda_{i-1}}}$



Method 3 : finite differences (in progress…)

Method 3 : Derivative and finite differences



Observation : the function $f(\lambda_i) = \|\bar{A}_t(\lambda_i) - P_t\|$ that we are trying to minimize looks convex…

Idea : minimize the derivative ? (still unknown !) $g(\lambda_i) = \frac{|f(\lambda_i) - f(\lambda_{i-1})|}{\lambda_i - \lambda_{i-1}}$

Final procedure : minimize an upper bound $\boxed{h(\lambda_i) = \frac{\|\bar{A}_t(\lambda_i) - \bar{A}_t(\lambda_{i-1})\|}{\lambda_i - \lambda_{i-1}}}$

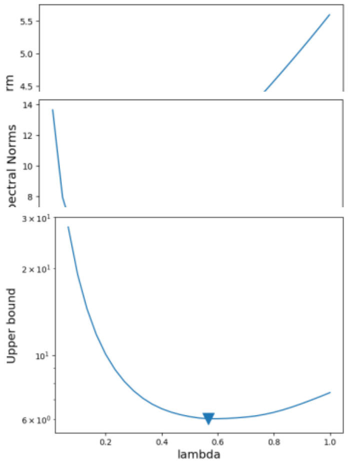
Lemma

Assume that whp f is strongly convex and $0 < c \leq f'' \leq C$ on an interval $[\underline{\lambda}, \overline{\lambda}]$, and $\lambda^\star, \lambda_{\hat{i}} \in [\underline{\lambda}, \overline{\lambda}]$, $\lambda_i - \lambda_{i-1} = \gamma$
Then, whp, we have

$$\|\bar{A}_t(\lambda_{\hat{i}}) - P_t\| \leq 2\delta^\star + C\left(\gamma + \frac{3C\gamma + 4\delta^\star/\gamma}{c}\right)^2$$

Method 3 : finite differences (in progress…)

Method 3 : Derivative and finite differences



Observation : the function $f(\lambda_i) = \|\bar{A}_t(\lambda_i) - P_t\|$ that we are trying to minimize looks convex…

Idea : minimize the derivative ? (still unknown !) $g(\lambda_i) = \frac{|f(\lambda_i) - f(\lambda_{i-1})|}{\lambda_i - \lambda_{i-1}}$

Final procedure : minimize an upper bound $\boxed{h(\lambda_i) = \frac{\|\bar{A}_t(\lambda_i) - \bar{A}_t(\lambda_{i-1})\|}{\lambda_i - \lambda_{i-1}}}$

Lemma

Assume that whp f is strongly convex and $0 < c \leq f'' \leq C$ on an interval $[\underline{\lambda}, \overline{\lambda}]$, and $\lambda^\star, \lambda_{\hat{i}} \in [\underline{\lambda}, \overline{\lambda}]$, $\lambda_i - \lambda_{i-1} = \gamma$

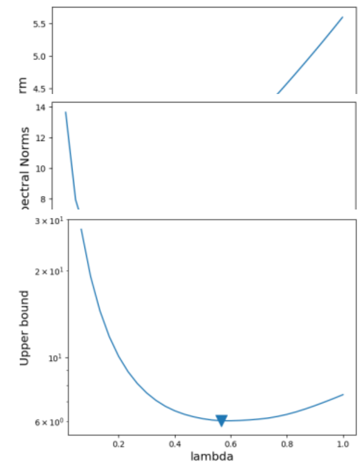
Then, whp, we have

$$\|\bar{A}_t(\lambda_{\hat{i}}) - P_t\| \leq 2\delta^\star + C\left(\gamma + \frac{3C\gamma + 4\delta^\star/\gamma}{c}\right)^2$$

Gives the right rate if : $\gamma = \mathcal{O}(\sqrt{\alpha_n n \varepsilon})$ $\quad c, C = \mathcal{O}((\alpha_n n)^{-1/4} \varepsilon^{-3/4})$

Method 3 : finite differences (in progress…)

Method 3 : Derivative and finite differences



Observation : the function $f(\lambda_i) = \|\bar{A}_t(\lambda_i) - P_t\|$ that we are trying to minimize looks convex…

Idea : minimize the derivative ? (still unknown !) $g(\lambda_i) = \frac{|f(\lambda_i) - f(\lambda_{i-1})|}{\lambda_i - \lambda_{i-1}}$

Final procedure : minimize an upper bound $\boxed{h(\lambda_i) = \frac{\|\bar{A}_t(\lambda_i) - \bar{A}_t(\lambda_{i-1})\|}{\lambda_i - \lambda_{i-1}}}$

Lemma

Assume that whp f is strongly convex and $0 < c \leq f'' \leq C$ on an interval $[\underline{\lambda}, \overline{\lambda}]$, and $\lambda^\star, \lambda_{\hat{i}} \in [\underline{\lambda}, \overline{\lambda}]$, $\lambda_i - \lambda_{i-1} = \gamma$
Then, whp, we have

$$\|\bar{A}_t(\lambda_{\hat{i}}) - P_t\| \leq 2\delta^\star + C\left(\gamma + \frac{3C\gamma + 4\delta^\star/\gamma}{c}\right)^2$$

Gives the right rate if : $\gamma = \mathcal{O}(\sqrt{\alpha_n n \varepsilon})$ $\quad c, C = \mathcal{O}((\alpha_n n)^{-1/4} \varepsilon^{-3/4})$

Future work: actually proving the convexity ?

For now: the upper bound $\delta_1(\lambda) + \delta_2(\lambda)$ has **the right strong convexity** on $[a\lambda^\star, b\lambda^\star]$ …

Outline

1. Main result
2. Choosing the forgetting factor
3. Experiments
4. Conclusion *(Bonus : GNN ?)*

Illustration on synthetic data

Quality of clustering $\longleftrightarrow$ $\|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\| \leq \delta_1(\lambda) + \delta_2(\lambda)$

$\lambda^\star = \mathcal{O}(\sqrt{n\alpha_n\varepsilon})$

Illustration on synthetic data

Quality of clustering $\longleftrightarrow \|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\| \leq \delta_1(\lambda) + \delta_2(\lambda)$

?

$\lambda^\star = \mathcal{O}(\sqrt{n\alpha_n\varepsilon})$

Illustration on synthetic data

Quality of clustering $\longleftrightarrow \|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\| \leq \delta_1(\lambda) + \delta_2(\lambda)$

?

$\boxed{\lambda^\star = \mathcal{O}(\sqrt{n\alpha_n\varepsilon})}$

$n = 500,$

$k = 5,$

$\alpha = \mathcal{O}(\frac{\log(n)}{n}),$

$\varepsilon = \mathcal{O}(\frac{1}{\log^2(n)})$

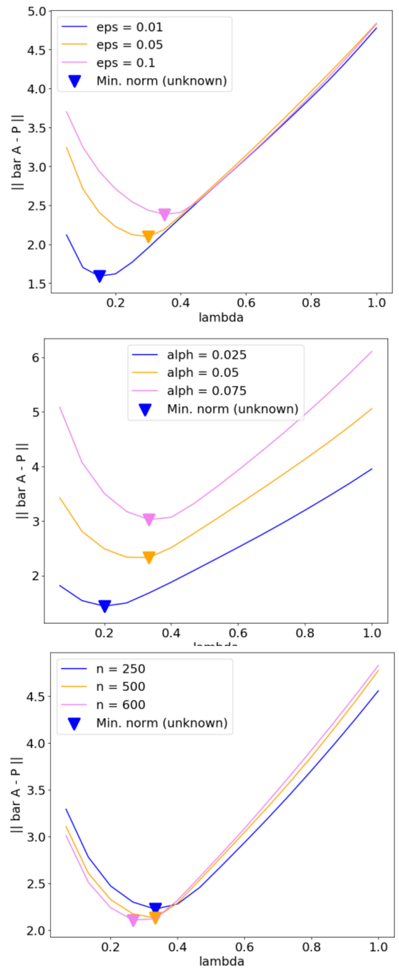


Illustration on synthetic data

Quality of clustering $\longleftrightarrow$ $\|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\| \leq \delta_1(\lambda) + \delta_2(\lambda)$

?

$\lambda^\star = \mathcal{O}(\sqrt{n\alpha_n \varepsilon})$

$n = 500,$

$k = 5,$

$\alpha = \mathcal{O}(\frac{\log(n)}{n}),$

$\varepsilon = \mathcal{O}(\frac{1}{\log^2(n)})$

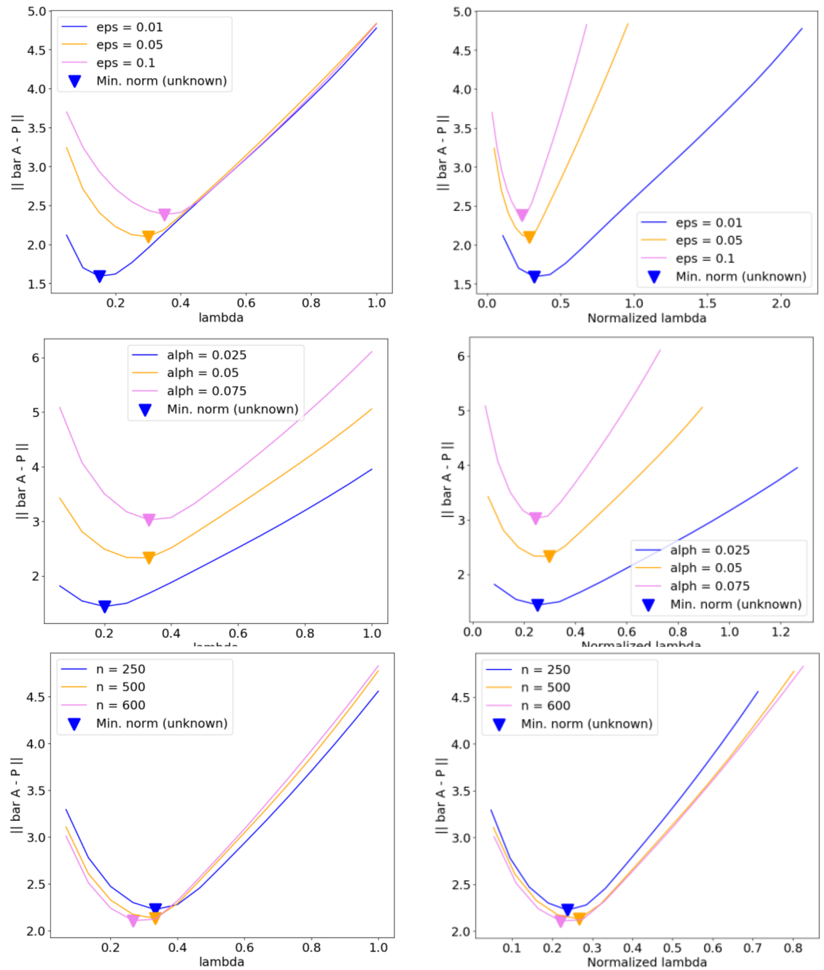


Illustration on synthetic data

Quality of clustering $\longleftrightarrow$ $$\|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\| \leq \delta_1(\lambda) + \delta_2(\lambda)$$

?

$$\lambda^\star = \mathcal{O}(\sqrt{n\alpha_n\varepsilon})$$

$n = 500,$

$k = 5,$

$\alpha = \mathcal{O}(\frac{\log(n)}{n}),$

$\varepsilon = \mathcal{O}(\frac{1}{\log^2(n)})$

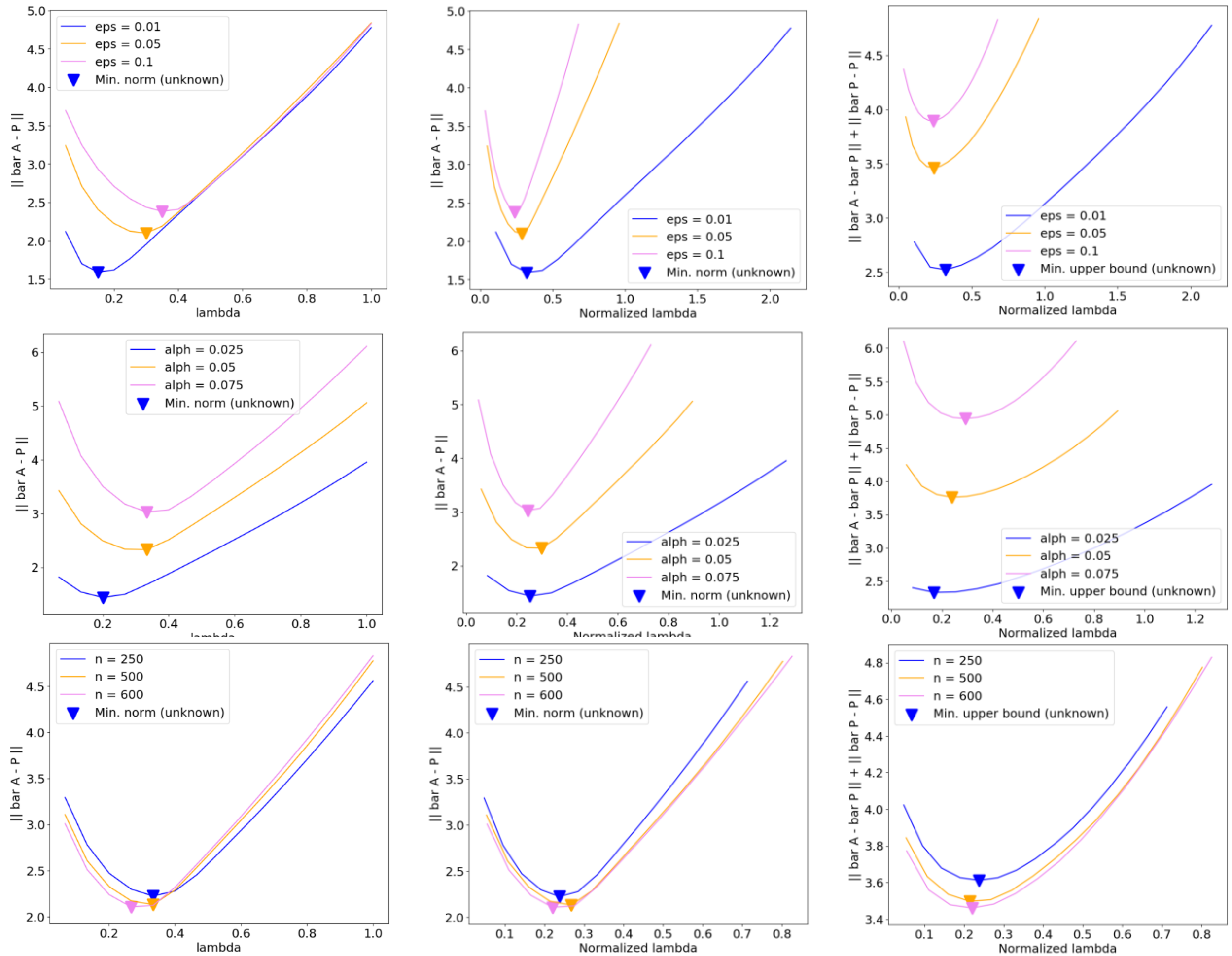


Illustration on synthetic data

$\boxed{\text{Quality of clustering}}$ $\leftrightarrow$ $\boxed{?}$ $\|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\| \leq \delta_1(\lambda) + \delta_2(\lambda)$

Illustration on synthetic data

Quality of clustering $\longleftrightarrow$? $\|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\| \leq \delta_1(\lambda) + \delta_2(\lambda)$

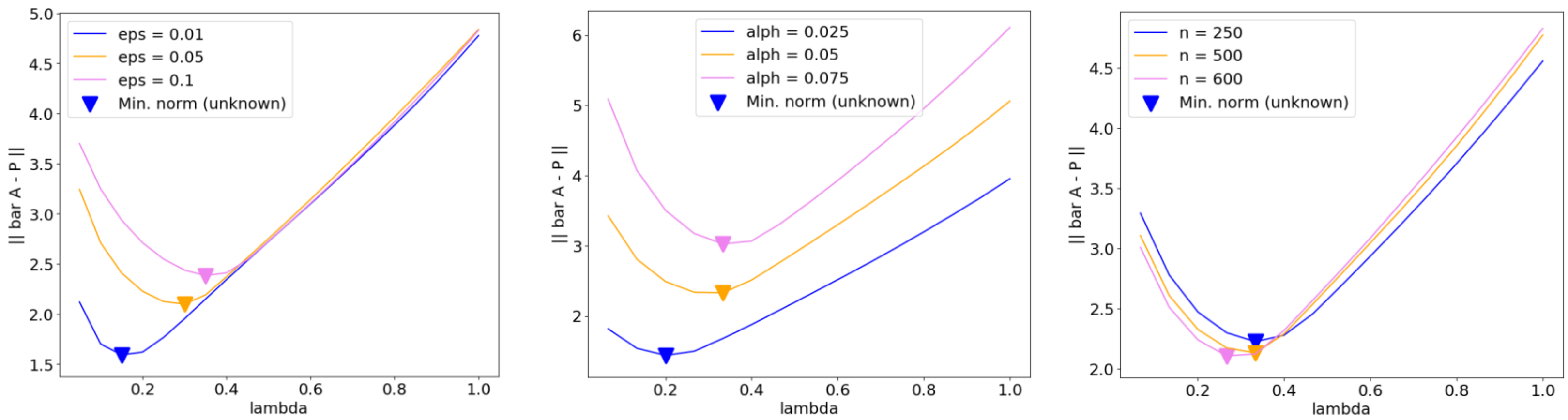


Illustration on synthetic data

Quality of clustering ?

$$\|\bar{A}_t - P_t\| \leq \|\bar{A}_t - \bar{P}_t\| + \|\bar{P}_t - P_t\| \leq \delta_1(\lambda) + \delta_2(\lambda)$$

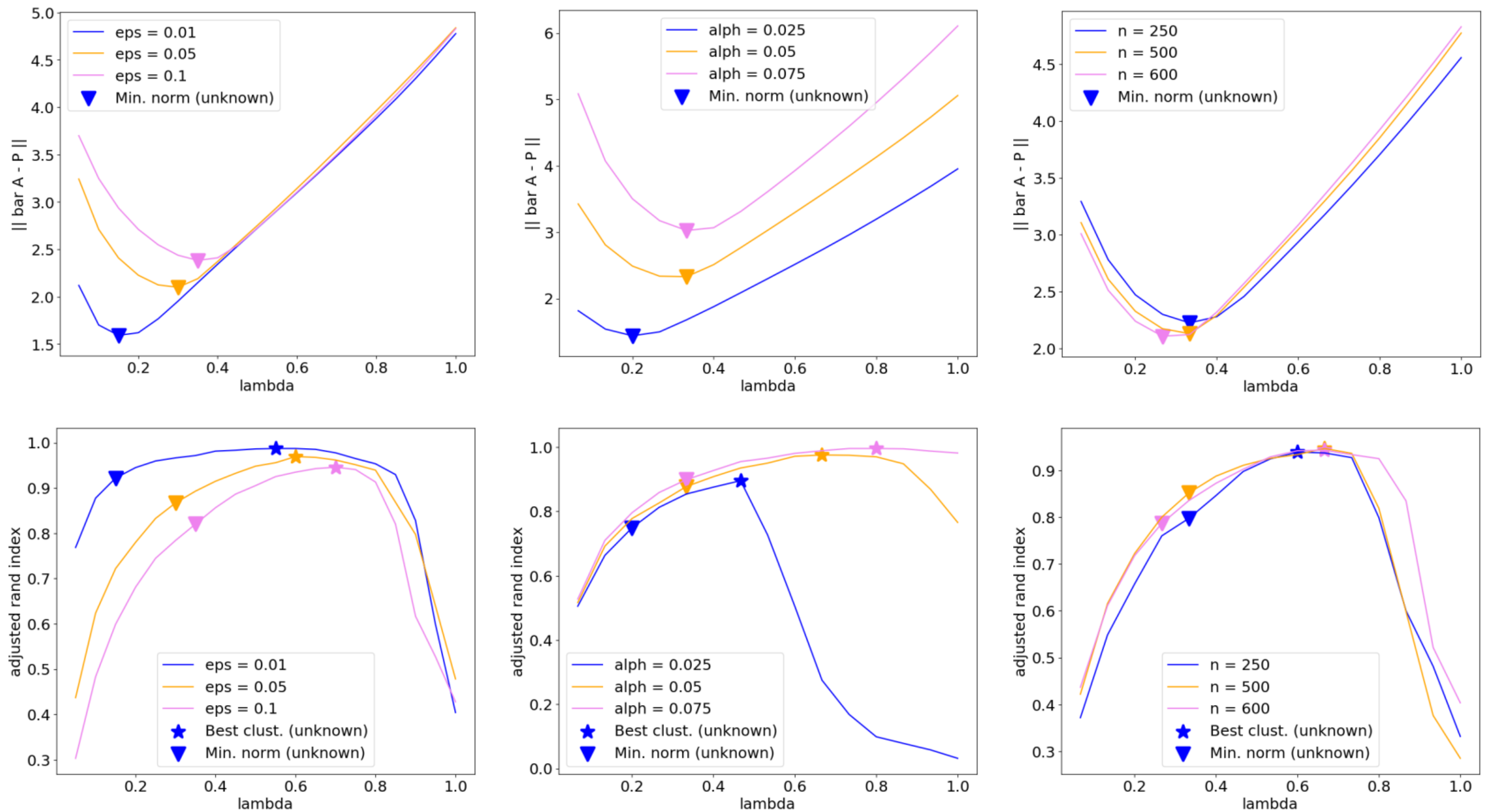
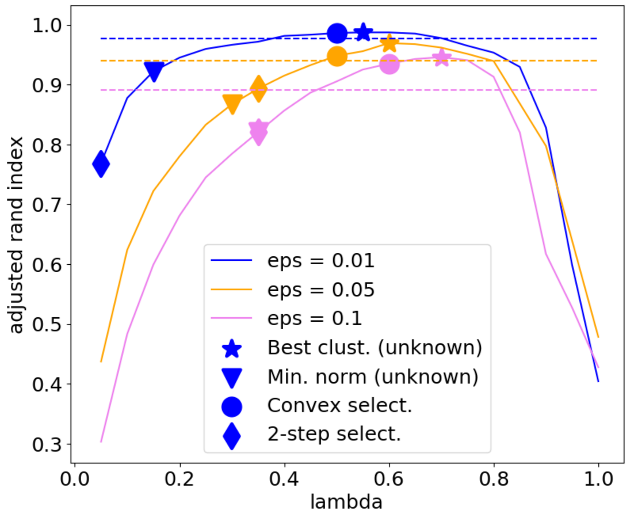
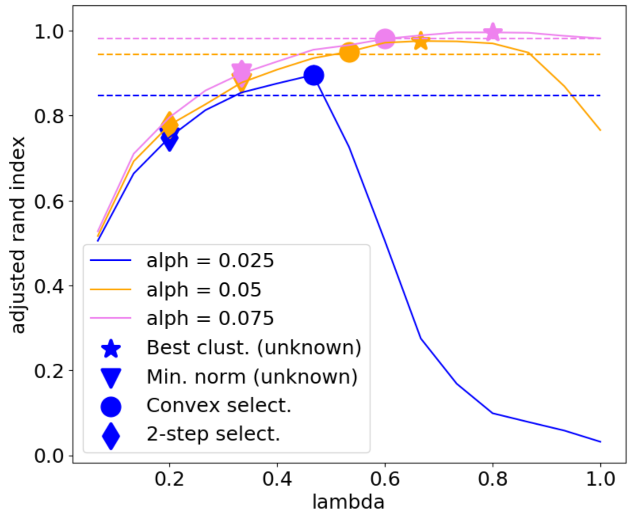


Illustration on synthetic data

Choice of forgetting factor, comparison with uniform average:





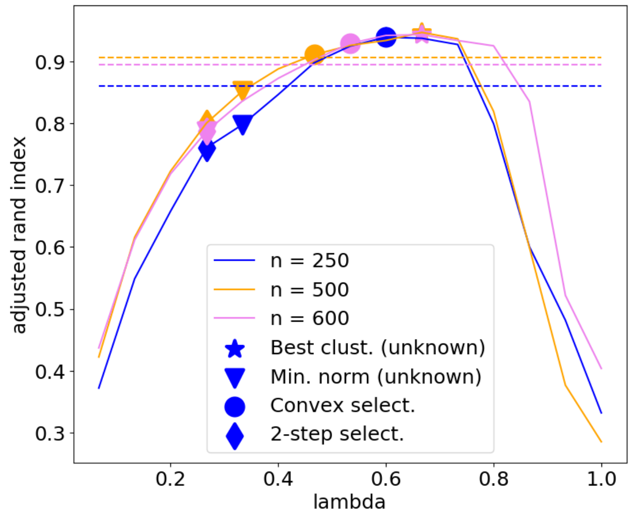
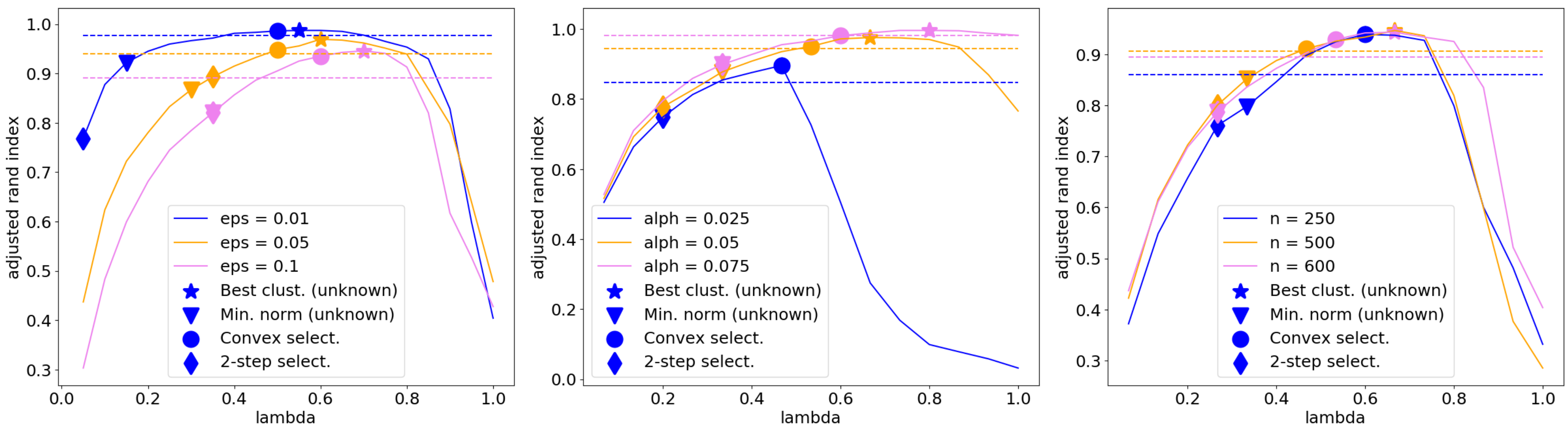


Illustration on synthetic data

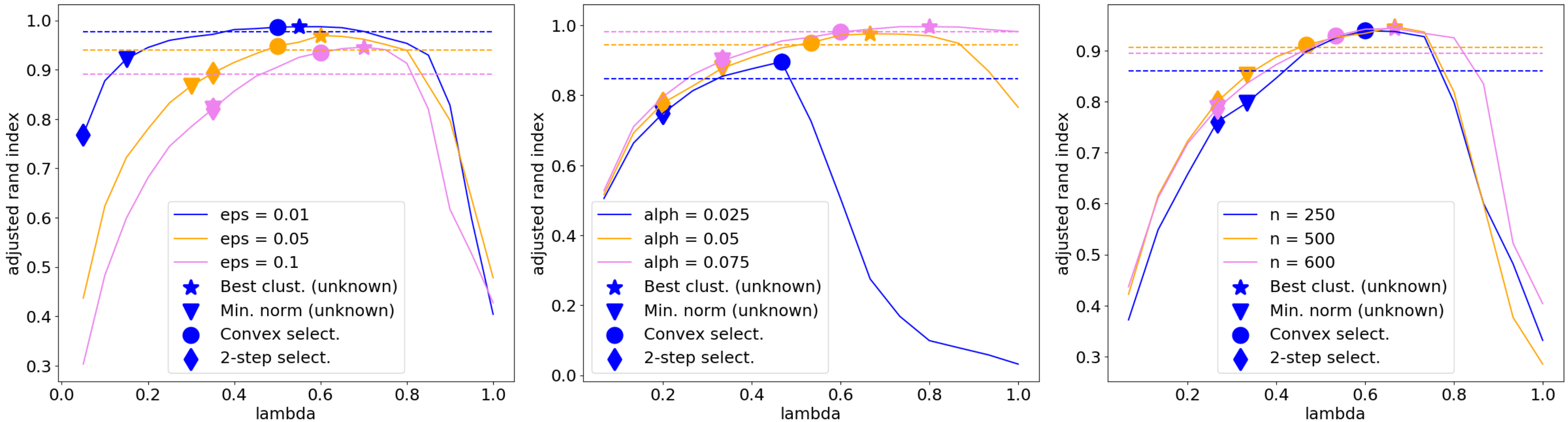
Choice of forgetting factor, comparison with uniform average:



- Choice by « proxy » of P_t often does not work… (tends to privilege low λ)

Illustration on synthetic data

Choice of forgetting factor, comparison with uniform average:



- Choice by « proxy » of P_t often does not work… (tends to privilege low λ)
- Choice by « finite differences » is even better than the (unknown) best uniform average

Outline

1. Main result
2. Choosing the forgetting factor
3. Experiments
4. Conclusion *(Bonus : GNN ?)*

Conclusion, outlooks

Conclusion

- **Improved Non-asymptotic guarantees** for smoothed Spectral Clustering for Dynamic Stochastic Block Model
  - ***Non-asymptotic guarantees for the sparse case !!***

- Efficient practical choice of forgetting factor
- Experiments show that it outperform uniform average

Conclusion, outlooks

Conclusion

- **Improved Non-asymptotic guarantees** for smoothed Spectral Clustering for Dynamic Stochastic Block Model
  - ***Non-asymptotic guarantees for the sparse case !!***
- Efficient practical choice of forgetting factor
- Experiments show that it outperform uniform average

Outlooks (many !)

- More justification for finite difference ?
- Laplacian ? (normalized, unnormalized…)
- Other use of Lei's modified Bernstein inequality for Bernoulli matrices ?

Conclusion, outlooks

Conclusion

- **Improved Non-asymptotic guarantees** for smoothed Spectral Clustering for Dynamic Stochastic Block Model
  - ***Non-asymptotic guarantees for the sparse case !!***
- Efficient practical choice of forgetting factor
- Experiments show that it outperform uniform average

Outlooks (many !)

- More justification for finite difference ?
- Laplacian ? (normalized, unnormalized…)
- Other use of Lei's modified Bernstein inequality for Bernoulli matrices ?
- **Detectability threshold** *à la* statistical physic of the difficult model !!

Done $\begin{cases} \alpha = \mathcal{O}(1/n) \\ \varepsilon = \mathcal{O}(1/\log(n)^2) \end{cases}$

Todo: $\begin{cases} \alpha = \mathcal{O}(1/n) \\ \varepsilon = cte \end{cases}$

Bonus : Universal invariant and equivariant Graph Neural Networks

N. Keriven[1], Gabriel Peyré[1]

[1]Ecole Normale Supérieure, Paris

Universal Invariant and Equivariant GNN

Graph

$$W \in \mathbb{R}^{n \times n}, P \star W := P^\top W P$$

Universal Invariant and Equivariant GNN

Graph	**Higher-order** tensor
$W \in \mathbb{R}^{n \times n}, P \star W := P^\top W P$	$W \in \mathbb{R}^{n^k}, P \star W := \ldots$

Universal Invariant and Equivariant GNN

Graph

$$W \in \mathbb{R}^{n \times n}, P \star W := P^\top W P$$

Higher-order tensor

$$W \in \mathbb{R}^{n^k}, P \star W := \ldots$$

Invariant functions (graph -> scalar)

$$f : \mathbb{R}^{n^k} \to \mathbb{R}, f(P \star W) = f(W)$$

Universal Invariant and Equivariant GNN

Graph

$$W \in \mathbb{R}^{n \times n}, P \star W := P^\top W P$$

Higher-order tensor

$$W \in \mathbb{R}^{n^k}, P \star W := \dots$$

Invariant functions (graph -> scalar)

$$f : \mathbb{R}^{n^k} \to \mathbb{R}, f(P \star W) = f(W)$$

Equivariant functions (graph -> graph)

$$f : \mathbb{R}^{n^k} \to \mathbb{R}^{n^\ell}, f(P \star W) = P \star f(W)$$

Universal Invariant and Equivariant GNN

Graph
$W \in \mathbb{R}^{n \times n}, P \star W := P^\top W P$

Higher-order tensor
$W \in \mathbb{R}^{n^k}, P \star W := \ldots$

Invariant functions (graph -> scalar)
$f : \mathbb{R}^{n^k} \to \mathbb{R}, f(P \star W) = f(W)$

Equivariant functions (graph -> graph)
$f : \mathbb{R}^{n^k} \to \mathbb{R}^{n^\ell}, f(P \star W) = P \star f(W)$

1-layer GNN: $f(W) = \sum_{s=1}^{S} I_s \rho\left(E_s(W) + B_s\right)$

Universal Invariant and Equivariant GNN

Graph

$W \in \mathbb{R}^{n\times n}, P \star W := P^\top W P$

Higher-order tensor

$W \in \mathbb{R}^{n^k}, P \star W := \dots$

Invariant functions (graph -> scalar)

$f : \mathbb{R}^{n^k} \to \mathbb{R}, f(P \star W) = f(W)$

Equivariant functions (graph -> graph)

$f : \mathbb{R}^{n^k} \to \mathbb{R}^{n^\ell}, f(P \star W) = P \star f(W)$

1-layer GNN: $$f(W) = \sum_{s=1}^{S} I_s \rho\left(E_s(W) + B_s\right)$$

Linear equivariant

$\mathbb{R}^{n\times n} \to \mathbb{R}^{n^k}$

Universal Invariant and Equivariant GNN

Graph

$W \in \mathbb{R}^{n \times n}, P \star W := P^\top W P$

Higher-order tensor

$W \in \mathbb{R}^{n^k}, P \star W := \dots$

Invariant functions (graph -> scalar)

$f : \mathbb{R}^{n^k} \to \mathbb{R}, f(P \star W) = f(W)$

Equivariant functions (graph -> graph)

$f : \mathbb{R}^{n^k} \to \mathbb{R}^{n^\ell}, f(P \star W) = P \star f(W)$

1-layer GNN: $f(W) = \sum_{s=1}^{S} I_s \rho \left(E_s(W) + B_s \right)$

Linear equivariant

$\mathbb{R}^{n \times n} \to \mathbb{R}^{n^k}$

Equivariant bias

$\mathbb{R}^{n^k}$

Universal Invariant and Equivariant GNN

Graph

$W \in \mathbb{R}^{n\times n}, P \star W := P^\top W P$

Higher-order tensor

$W \in \mathbb{R}^{n^k}, P \star W := \dots$

Invariant functions (graph -> scalar)

$f : \mathbb{R}^{n^k} \to \mathbb{R}, f(P \star W) = f(W)$

Equivariant functions (graph -> graph)

$f : \mathbb{R}^{n^k} \to \mathbb{R}^{n^\ell}, f(P \star W) = P \star f(W)$

1-layer GNN: $$f(W) = \sum_{s=1}^{S} I_s \rho\left(E_s(W) + B_s\right)$$

Linear invariant or equivariant

$\mathbb{R}^{n^k} \to \mathbb{R}$ $\quad$ $\mathbb{R}^{n^k} \to \mathbb{R}^n$

Linear equivariant

$\mathbb{R}^{n\times n} \to \mathbb{R}^{n^k}$

Equivariant bias

$\mathbb{R}^{n^k}$

Universal Invariant and Equivariant GNN

Graph

$W \in \mathbb{R}^{n\times n}, P \star W := P^\top W P$

Higher-order tensor

$W \in \mathbb{R}^{n^k}, P \star W := \ldots$

Invariant functions (graph -> scalar)

$f : \mathbb{R}^{n^k} \to \mathbb{R}, f(P \star W) = f(W)$

Equivariant functions (graph -> graph)

$f : \mathbb{R}^{n^k} \to \mathbb{R}^{n^\ell}, f(P \star W) = P \star f(W)$

1-layer GNN: $$f(W) = \sum_{s=1}^{S} I_s \rho\left(E_s(W) + B_s\right)$$

Linear invariant or equivariant

$\mathbb{R}^{n^k} \to \mathbb{R}$ $\quad$ $\mathbb{R}^{n^k} \to \mathbb{R}^n$

Linear equivariant

$\mathbb{R}^{n\times n} \to \mathbb{R}^{n^k}$

Equivariant bias

$\mathbb{R}^{n^k}$

Characterized by [Maron et al 2018]: basis does not depend on n !!

Universal Invariant and Equivariant GNN

Graph
$W \in \mathbb{R}^{n \times n}, P \star W := P^\top W P$

Higher-order tensor
$W \in \mathbb{R}^{n^k}, P \star W := \ldots$

Invariant functions (graph -> scalar)
$f : \mathbb{R}^{n^k} \to \mathbb{R}, f(P \star W) = f(W)$

Equivariant functions (graph -> graph)
$f : \mathbb{R}^{n^k} \to \mathbb{R}^{n^\ell}, f(P \star W) = P \star f(W)$

1-layer GNN: $f(W) = \sum_{s=1}^{S} I_s \rho \left(E_s(W) + B_s \right)$

Linear invariant or equivariant
$\mathbb{R}^{n^k} \to \mathbb{R}$ $\quad$ $\mathbb{R}^{n^k} \to \mathbb{R}^n$

Linear equivariant
$\mathbb{R}^{n \times n} \to \mathbb{R}^{n^k}$

Equivariant bias
$\mathbb{R}^{n^k}$

Characterized by [Maron et al 2018]: basis does not depend on n !!

Thm: GNNs are **universal approximators** of invariant (resp. equivariant) continuous functions.

Universal Invariant and Equivariant GNN

Graph

$W \in \mathbb{R}^{n \times n}, P \star W := P^\top W P$

Higher-order tensor

$W \in \mathbb{R}^{n^k}, P \star W := \dots$

Invariant functions (graph -> scalar)

$f : \mathbb{R}^{n^k} \to \mathbb{R}, f(P \star W) = f(W)$

Equivariant functions (graph -> graph)

$f : \mathbb{R}^{n^k} \to \mathbb{R}^{n^\ell}, f(P \star W) = P \star f(W)$

1-layer GNN: $$f(W) = \sum_{s=1}^{S} I_s \rho\left(E_s(W) + B_s\right)$$

Linear invariant or equivariant

$\mathbb{R}^{n^k} \to \mathbb{R}$ $\quad \mathbb{R}^{n^k} \to \mathbb{R}^n$

Linear equivariant

$\mathbb{R}^{n \times n} \to \mathbb{R}^{n^k}$

Equivariant bias

$\mathbb{R}^{n^k}$

Characterized by [Maron et al 2018]: basis does not depend on n !!

Thm: GNNs are **universal approximators** of invariant (resp. equivariant) continuous functions.

- *Invariant case*: Stone-Weierstrass *[Hornik1989]*, the **separation of points** is hard to prove !

Universal Invariant and Equivariant GNN

Graph

$W \in \mathbb{R}^{n \times n}, P \star W := P^\top W P$

Higher-order tensor

$W \in \mathbb{R}^{n^k}, P \star W := \dots$

Invariant functions (graph -> scalar)

$f : \mathbb{R}^{n^k} \to \mathbb{R}, f(P \star W) = f(W)$

Equivariant functions (graph -> graph)

$f : \mathbb{R}^{n^k} \to \mathbb{R}^{n^\ell}, f(P \star W) = P \star f(W)$

1-layer GNN: $$f(W) = \sum_{s=1}^{S} I_s \rho\left(E_s(W) + B_s\right)$$

Linear invariant or equivariant

$\mathbb{R}^{n^k} \to \mathbb{R}$ $\quad$ $\mathbb{R}^{n^k} \to \mathbb{R}^n$

Linear equivariant

$\mathbb{R}^{n \times n} \to \mathbb{R}^{n^k}$

Equivariant bias

$\mathbb{R}^{n^k}$

Characterized by [Maron et al 2018]: basis does not depend on n !!

Thm: GNNs are **universal approximators** of invariant (resp. equivariant) continuous functions.

- *Invariant case*: Stone-Weierstrass *[Hornik1989]*, the **separation of points** is hard to prove !
- *Equivariant case*: **new S-W theorem** ! *(non-trivial adaptation of [Brosowski 1981])*

Universal Invariant and Equivariant GNN

Graph
$W \in \mathbb{R}^{n \times n}, P \star W := P^\top W P$

Higher-order tensor
$W \in \mathbb{R}^{n^k}, P \star W := \dots$

Invariant functions (graph -> scalar)
$f : \mathbb{R}^{n^k} \to \mathbb{R}, f(P \star W) = f(W)$

Equivariant functions (graph -> graph)
$f : \mathbb{R}^{n^k} \to \mathbb{R}^{n^\ell}, f(P \star W) = P \star f(W)$

1-layer GNN: $$f(W) = \sum_{s=1}^{S} I_s \rho\left(E_s(W) + B_s\right)$$

Linear invariant or equivariant
$\mathbb{R}^{n^k} \to \mathbb{R}$ $\qquad$ $\mathbb{R}^{n^k} \to \mathbb{R}^n$

Linear equivariant
$\mathbb{R}^{n \times n} \to \mathbb{R}^{n^k}$

Equivariant bias
$\mathbb{R}^{n^k}$

Characterized by [Maron et al 2018]: basis does not depend on n !!

Thm: GNNs are **universal approximators** of invariant (resp. equivariant) continuous functions.

- *Invariant case*: Stone-Weierstrass *[Hornik1989]*, the **separation of points** is hard to prove !
- *Equivariant case*: **new S-W theorem** ! *(non-trivial adaptation of [Brosowski 1981])*

- Case **invariant** already known *[Maron 2019]*, high k is necessary !

Thank you !

Preprints are coming soon !

data-ens.github.io

Enter the data challenges!
Come to the colloquium!
Come to the Laplace seminars!