gipsa-lab

Grenoble | images | parole | signal | automatique | laboratoire

Graph Signal Smoothing via Random Spanning Forests

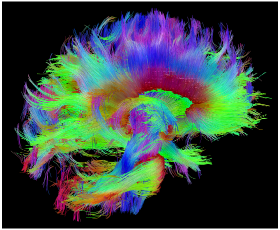
Yusuf Yigit Pilavci*
Pierre-Olivier Amblard
Simon Barthelmé
Nicolas Tremblay

17/04/2020

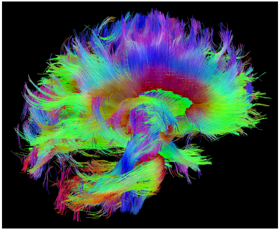
cnrs Grenoble INP UNIVERSITÉ Grenoble Alpes UMR 5216

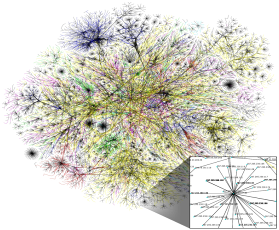
Graphs and Graph Signals

Graphs and Graph Signals

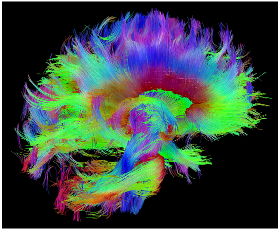


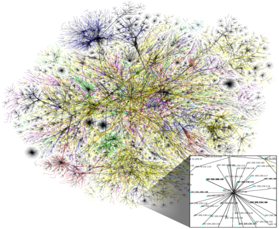
Graphs and Graph Signals



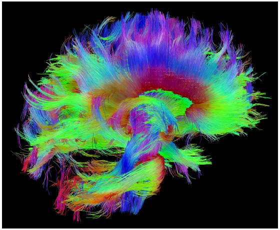


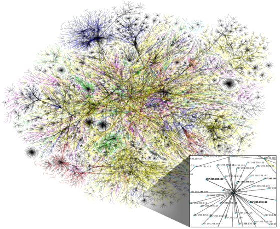
Graphs and Graph Signals

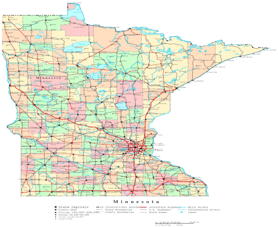




Graphs and Graph Signals







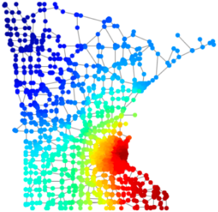
Signal Denoising via Smoothing

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, a graph signal is $\mathbf{x} \in \mathbb{R}^{|\mathcal{V}|}$.

Signal Denoising via Smoothing

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, a graph signal is $\mathbf{x} \in \mathbb{R}^{|\mathcal{V}|}$.

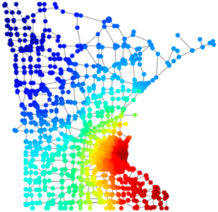
$\mathbf{x}$:



Signal Denoising via Smoothing

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, a graph signal is $\mathbf{x} \in \mathbb{R}^{|\mathcal{V}|}$.

x:

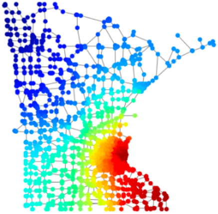


y:

Signal Denoising via Smoothing

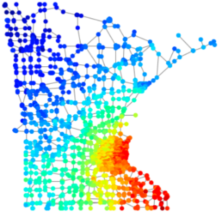
Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, a graph signal is $\mathbf{x} \in \mathbb{R}^{|\mathcal{V}|}$.

$\mathbf{x}$:



$\mathbf{y}$:

Smoothed $\mathbf{y}$:



Tikhonov Regularization

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{z} \in \mathbb{R}^n} q$$

Tikhonov Regularization

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{z} \in \mathbb{R}^n} q \underbrace{||\mathbf{y} - \mathbf{z}||^2}_{\text{Fidelity}} +$$

Tikhonov Regularization

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{z}\in\mathbb{R}^n} q \underbrace{||\mathbf{y} - \mathbf{z}||^2}_{\text{Fidelity}} + \underbrace{\mathbf{z}^T \mathsf{L}\mathbf{z}}_{\text{Regularization}} \quad , \quad q > 0$$

where L is the graph Laplacian

Tikhonov Regularization

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{z}\in\mathbb{R}^n} q \underbrace{||\mathbf{y} - \mathbf{z}||^2}_{\text{Fidelity}} + \underbrace{\mathbf{z}^T \mathsf{L} \mathbf{z}}_{\text{Regularization}} \quad , \quad q > 0$$

where L is the graph Laplacian and $\mathbf{z}^T \mathsf{L} \mathbf{z} = \sum_{(i,j)\in\mathcal{E}} w(i,j)(z_i - z_j)^2$.

Tikhonov Regularization

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{z}\in\mathbb{R}^n} q\underbrace{||\mathbf{y}-\mathbf{z}||^2}_{\text{Fidelity}} + \underbrace{\mathbf{z}^T\mathsf{L}\mathbf{z}}_{\text{Regularization}} \quad , \quad q > 0$$

where L is the graph Laplacian and $\mathbf{z}^T\mathsf{L}\mathbf{z} = \sum_{(i,j)\in\mathcal{E}} w(i,j)(z_i - z_j)^2$.

- The explicit solution to this problem is:

$$\hat{\mathbf{x}} = \mathsf{K}\mathbf{y} \text{ with } \mathsf{K} = (\mathsf{L} + q\mathsf{I})^{-1}q\mathsf{I}$$

Tikhonov Regularization

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{z}\in\mathbb{R}^n} q \underbrace{||\mathbf{y} - \mathbf{z}||^2}_{\text{Fidelity}} + \underbrace{\mathbf{z}^T \mathsf{L} \mathbf{z}}_{\text{Regularization}} \quad , \quad q > 0$$

where L is the graph Laplacian and $\mathbf{z}^T \mathsf{L} \mathbf{z} = \sum_{(i,j)\in\mathcal{E}} w(i,j)(z_i - z_j)^2$.

- The explicit solution to this problem is:

$$\hat{\mathbf{x}} = \mathsf{K}\mathbf{y} \text{ with } \mathsf{K} = (\mathsf{L} + q\mathbf{I})^{-1} q\mathbf{I}$$

- Direct computation of K requires $\mathcal{O}(n^3)$ elementary operations due to the inverse.

Tikhonov Regularization

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{z}\in\mathbb{R}^n} q \underbrace{||\mathbf{y} - \mathbf{z}||^2}_{\text{Fidelity}} + \underbrace{\mathbf{z}^T \mathsf{L}\mathbf{z}}_{\text{Regularization}} \quad , \quad q > 0$$

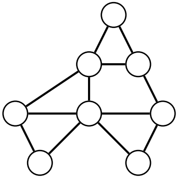
where L is the graph Laplacian and $\mathbf{z}^T \mathsf{L}\mathbf{z} = \sum_{(i,j)\in\mathcal{E}} w(i,j)(z_i - z_j)^2$.

- The explicit solution to this problem is:

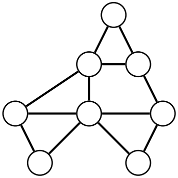
$$\hat{\mathbf{x}} = \mathsf{K}\mathbf{y} \text{ with } \mathsf{K} = (\mathsf{L} + q\mathsf{I})^{-1} q\mathsf{I}$$

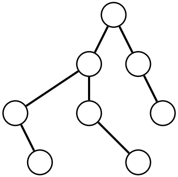
- Direct computation of K requires $\mathcal{O}(n^3)$ elementary operations due to the inverse.
- For large n, iterative methods and polynomial approximations are the state-of-the-art. Both compute $\hat{\mathbf{x}}$ in linear time in the number of edges $|\mathcal{E}|$.

Spanning Forests, Roots and Partitions



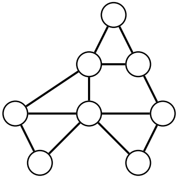
Spanning Forests, Roots and Partitions

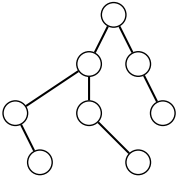




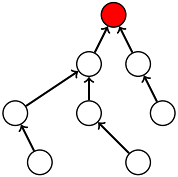
Spanning Tree

Spanning Forests, Roots and Partitions



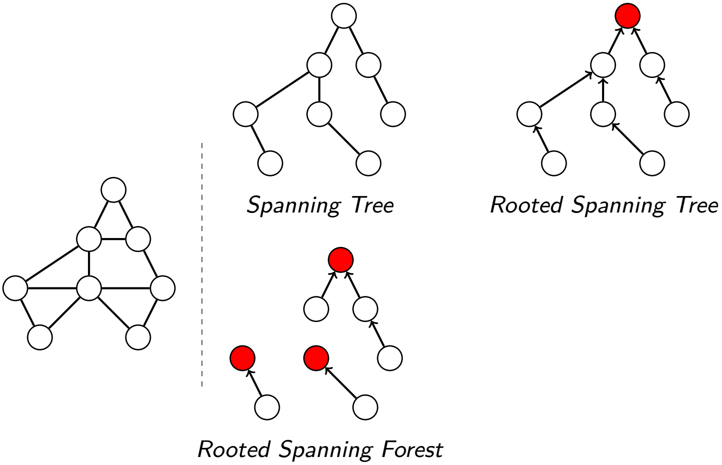


Spanning Tree

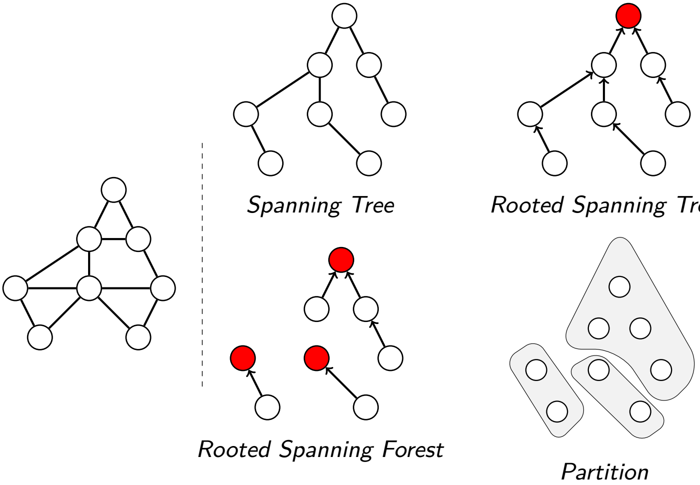


Rooted Spanning Tree

Spanning Forests, Roots and Partitions

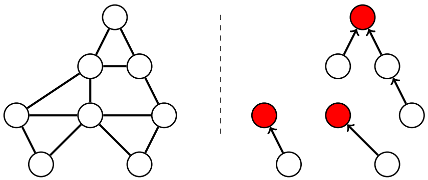


Spanning Forests, Roots and Partitions



A few notation

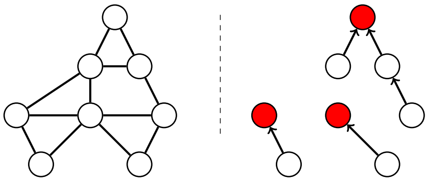
- Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, we denote:



  - a spanning forest as ϕ

A few notation

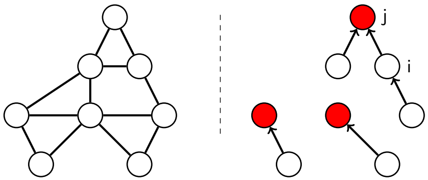
- Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, we denote:



  - a spanning forest as ϕ and its root set as $\rho(\phi)$,

A few notation

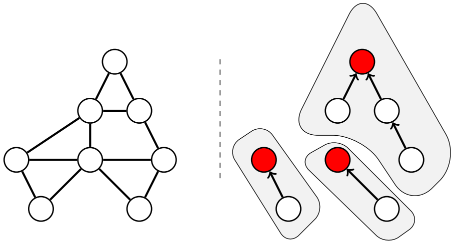
- Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, we denote:



- a spanning forest as ϕ and its root set as $\rho(\phi)$,
- the root of vertex i in ϕ as $r_\phi(i) = j$

A few notation

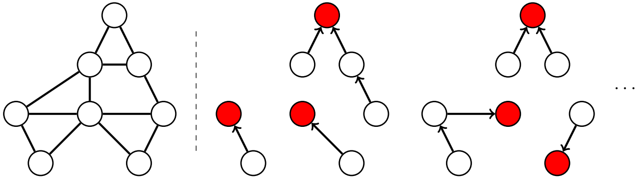
- Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, we denote:



  - a spanning forest as ϕ and its root set as $\rho(\phi)$,
  - the root of vertex i in ϕ as $r_\phi(i) = j$
  - the partition associated to ϕ as $\pi(\phi) = \{\mathcal{V}_1, \dots, \mathcal{V}_{|\rho(\phi)|}\}$.

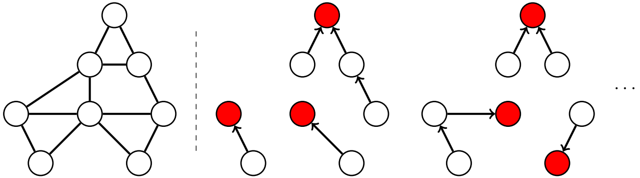
Random Spanning Forests

- More than one spanning forest is generally possible:



Random Spanning Forests

- More than one spanning forest is generally possible:

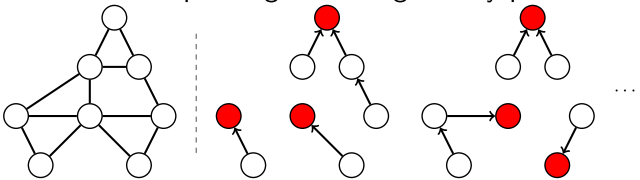


- Thus, we model them statistically. The probability distribution of random spanning forests:

$$P(\Phi_q = \phi) \propto q^{|\rho(\phi)|} \prod_{(i,j)\in\mathcal{E}_\phi} w(i,j) \tag{1}$$

Random Spanning Forests

- More than one spanning forest is generally possible:



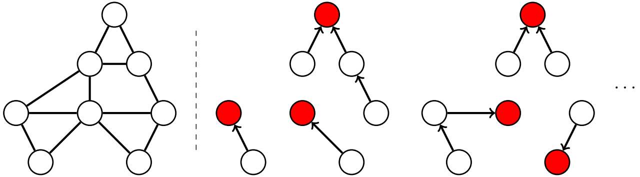
- Thus, we model them statistically. The probability distribution of random spanning forests:

$$P(\Phi_q = \phi) \propto q^{|\rho(\phi)|} \prod_{(i,j)\in\mathcal{E}_\phi} w(i,j) \quad (1)$$

- The computational cost to sample from Φ_q is $\mathcal{O}(\frac{|\mathcal{E}|}{q})$

Random Spanning Forests

- More than one spanning forest is generally possible:



- Thus, we model them statistically. The probability distribution of random spanning forests:

$$P(\Phi_q = \phi) \propto q^{|\rho(\phi)|} \prod_{(i,j) \in \mathcal{E}_\phi} w(i,j) \quad (1)$$

- The computational cost to sample from Φ_q is $\mathcal{O}(\frac{|\mathcal{E}|}{q})$
- Importantly, the probability of node i rooted at j in Φ_q reads:

$$P(r_{\Phi_q}(i) = j) = \mathsf{K}_{i,j} \text{ with } \mathsf{K} = (\mathsf{L} + q\mathsf{I})^{-1} q\mathsf{I}$$

RSF based Estimators

The first estimator $\tilde{\mathbf{x}}$

- Our first estimator for computing $\hat{\mathbf{x}} = \mathsf{K}\mathbf{y}$ is :

$$\tilde{x}(i) = y\left(r_{\Phi_q}(i)\right)$$

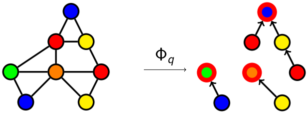
RSF based Estimators

The first estimator $\tilde{\mathbf{x}}$

- Our first estimator for computing $\hat{\mathbf{x}} = \mathsf{K}\mathbf{y}$ is :

$$\tilde{x}(i) = y\left(r_{\Phi_q}(i)\right)$$

- In practice, we propagate the measurement of the root in each tree



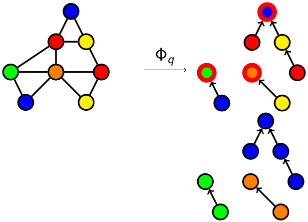
RSF based Estimators

The first estimator $\tilde{\mathbf{x}}$

- Our first estimator for computing $\hat{\mathbf{x}} = \mathsf{K}\mathbf{y}$ is :

$$\tilde{x}(i) = y\left(r_{\Phi_q}(i)\right)$$

- In practice, we propagate the measurement of the root in each tree



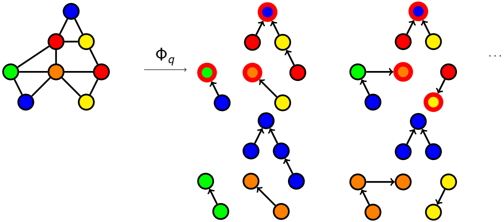
RSF based Estimators

The first estimator $\tilde{\mathbf{x}}$

- Our first estimator for computing $\hat{\mathbf{x}} = \mathsf{K}\mathbf{y}$ is :

$$\tilde{x}(i) = y\left(r_{\Phi_q}(i)\right)$$

- In practice, we propagate the measurement of the root in each tree



RSF based Estimators

An improved estimator $\bar{\mathbf{x}}$

- An improved estimator :

$$\bar{x}(i) = \frac{1}{|\mathcal{V}_{t(i)}|} \sum_{j \in \mathcal{V}_{t(i)}} y(j)$$

where $\mathcal{V}_{t(i)}$ gives the vertex set of the tree that includes i in $\pi(\Phi_q)$.

RSF based Estimators

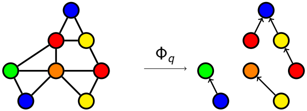
An improved estimator $\bar{\mathbf{x}}$

- An improved estimator :

$$\bar{x}(i) = \frac{1}{|\mathcal{V}_{t(i)}|} \sum_{j \in \mathcal{V}_{t(i)}} y(j)$$

where $\mathcal{V}_{t(i)}$ gives the vertex set of the tree that includes i in $\pi(\Phi_q)$.

- In practice,



RSF based Estimators

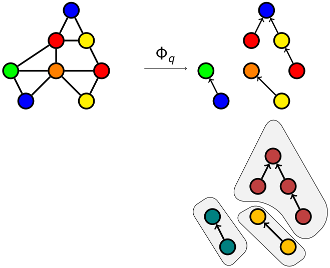
An improved estimator $\bar{\mathbf{x}}$

- An improved estimator :

$$\bar{x}(i) = \frac{1}{|\mathcal{V}_{t(i)}|} \sum_{j \in \mathcal{V}_{t(i)}} y(j)$$

where $\mathcal{V}_{t(i)}$ gives the vertex set of the tree that includes i in $\pi(\Phi_q)$.

- In practice,



RSF based Estimators

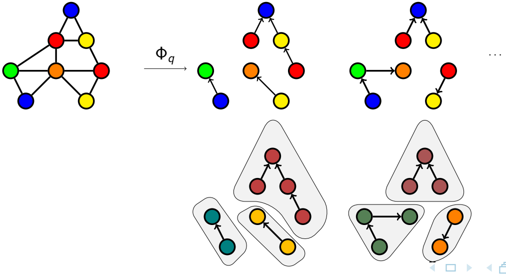
An improved estimator $\bar{\mathbf{x}}$

- An improved estimator :

$$\bar{x}(i) = \frac{1}{|\mathcal{V}_{t(i)}|} \sum_{j \in \mathcal{V}_{t(i)}} y(j)$$

where $\mathcal{V}_{t(i)}$ gives the vertex set of the tree that includes i in $\pi(\Phi_q)$.

- In practice,



Comparison of Estimators

- Both estimators are unbiased: $\mathbb{E}[\tilde{x}(i)] = \mathbb{E}[\bar{x}(i)] = \mathsf{K}\mathbf{y}(i)$

Comparison of Estimators

- Both estimators are unbiased: $\mathbb{E}\left[\tilde{x}(i)\right] = \mathbb{E}\left[\bar{x}(i)\right] = \mathsf{K}\mathbf{y}(i)$
- Moreover, the expected error for $\tilde{x}(i)$:

$$\mathbb{E}\left[||\hat{\mathbf{x}} - \tilde{\mathbf{x}}||^2\right] = \mathbf{y}^T(\mathbf{I} - \mathsf{K}^2)\mathbf{y}$$

Comparison of Estimators

- Both estimators are unbiased: $\mathbb{E}\left[\tilde{x}(i)\right] = \mathbb{E}\left[\bar{x}(i)\right] = \mathsf{K}\mathbf{y}(i)$
- Moreover, the expected error for $\tilde{x}(i)$:

$$\mathbb{E}\left[||\hat{\mathbf{x}} - \tilde{\mathbf{x}}||^2\right] = \mathbf{y}^T(\mathbf{I} - \mathsf{K}^2)\mathbf{y}$$

- The expected error for $\bar{x}(i)$:

$$\mathbb{E}\left[||\hat{\mathbf{x}} - \tilde{\mathbf{x}}||^2\right] = \mathbf{y}^T(\mathsf{K} - \mathsf{K}^2)\mathbf{y}$$

Comparison of Estimators

- Both estimators are unbiased: $\mathbb{E}\left[\tilde{x}(i)\right] = \mathbb{E}\left[\bar{x}(i)\right] = \mathsf{K}\mathbf{y}(i)$
- Moreover, the expected error for $\tilde{x}(i)$:

$$\mathbb{E}\left[||\hat{\mathbf{x}} - \tilde{\mathbf{x}}||^2\right] = \mathbf{y}^T(\mathbf{I} - \mathsf{K}^2)\mathbf{y}$$

- The expected error for $\bar{x}(i)$:

$$\mathbb{E}\left[||\hat{\mathbf{x}} - \tilde{\mathbf{x}}||^2\right] = \mathbf{y}^T(\mathsf{K} - \mathsf{K}^2)\mathbf{y}$$

- Recalling $\mathsf{K} = (\mathsf{L} + q\mathbf{I})^{-1}q\mathbf{I} \preceq 1$, we have

$$\mathbb{E}\left[||\hat{\mathbf{x}} - \bar{\mathbf{x}}||^2\right] \leq \mathbb{E}\left[||\hat{\mathbf{x}} - \tilde{\mathbf{x}}||^2\right]$$

Comparison of Estimators

- Both estimators are unbiased: $\mathbb{E}\left[\tilde{x}(i)\right] = \mathbb{E}\left[\bar{x}(i)\right] = \mathsf{K}\mathbf{y}(i)$
- Moreover, the expected error for $\tilde{x}(i)$:

$$\mathbb{E}\left[||\hat{\mathbf{x}} - \tilde{\mathbf{x}}||^2\right] = \mathbf{y}^T(\mathbf{I} - \mathsf{K}^2)\mathbf{y}$$

- The expected error for $\bar{x}(i)$:

$$\mathbb{E}\left[||\hat{\mathbf{x}} - \tilde{\mathbf{x}}||^2\right] = \mathbf{y}^T(\mathsf{K} - \mathsf{K}^2)\mathbf{y}$$

- Recalling $\mathsf{K} = (\mathsf{L} + q\mathbf{I})^{-1}q\mathbf{I} \preceq 1$, we have

$$\mathbb{E}\left[||\hat{\mathbf{x}} - \bar{\mathbf{x}}||^2\right] \leq \mathbb{E}\left[||\hat{\mathbf{x}} - \tilde{\mathbf{x}}||^2\right]$$

- The complexity for both is $\mathcal{O}(\frac{N|\mathcal{E}|}{q})$.

Comparison of Estimators

- Both estimators are unbiased: $\mathbb{E}[\tilde{x}(i)] = \mathbb{E}[\bar{x}(i)] = \mathsf{K}\mathbf{y}(i)$
- Moreover, the expected error for $\tilde{x}(i)$:

$$\mathbb{E}\left[||\hat{\mathbf{x}} - \tilde{\mathbf{x}}||^2\right] = \mathbf{y}^T(\mathbf{I} - \mathsf{K}^2)\mathbf{y}$$

- The expected error for $\bar{x}(i)$:

$$\mathbb{E}\left[||\hat{\mathbf{x}} - \tilde{\mathbf{x}}||^2\right] = \mathbf{y}^T(\mathsf{K} - \mathsf{K}^2)\mathbf{y}$$

- Recalling $\mathsf{K} = (\mathsf{L} + q\mathbf{I})^{-1}q\mathbf{I} \preceq 1$, we have

$$\mathbb{E}\left[||\hat{\mathbf{x}} - \bar{\mathbf{x}}||^2\right] \leq \mathbb{E}\left[||\hat{\mathbf{x}} - \tilde{\mathbf{x}}||^2\right]$$

- The complexity for both is $\mathcal{O}(\frac{N|\mathcal{E}|}{q})$.
- Both can be used for computing a more generalized form

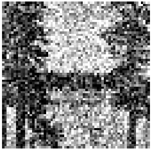
$$\hat{\mathbf{x}} = (\mathsf{Q} + \mathsf{L})^{-1}\mathsf{Q}\mathbf{y} \text{ with } \mathsf{Q} = \text{diag}(q_1, \ldots, q_n)$$

Experiments

Image Denoising

x:

y:

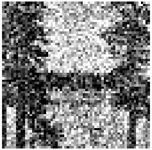


Experiments

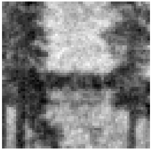
Image Denoising

$\mathbf{x}$:

$\mathbf{y}$:



$\hat{\mathbf{x}}$:

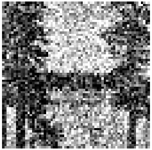


Experiments

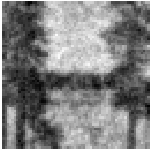
Image Denoising

$\mathbf{x}$:

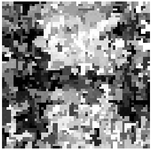
$\mathbf{y}$:



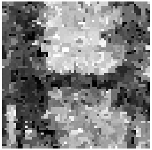
$\hat{\mathbf{x}}$:



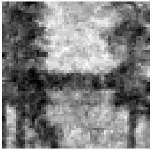
$\tilde{\mathbf{x}}(N = 1)$:



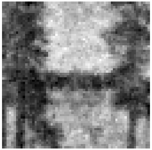
$\bar{\mathbf{x}}(N = 1)$:



$\tilde{\mathbf{x}}(N = 20)$:

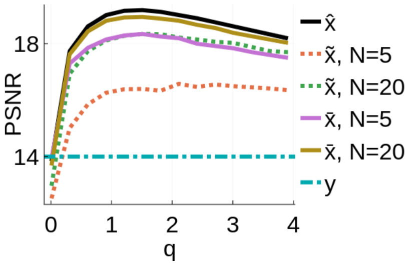


$\bar{\mathbf{x}}(N = 20)$:



Experiments

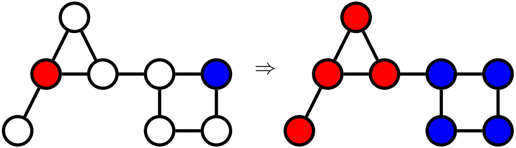
Image Denoising



Experiments

Semi-Supervised Learning for Node Classification

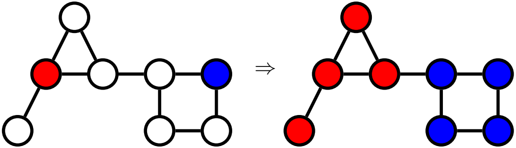
- Problem: Given a few labels over the nodes, infer the others



Experiments

Semi-Supervised Learning for Node Classification

- Problem: Given a few labels over the nodes, infer the others



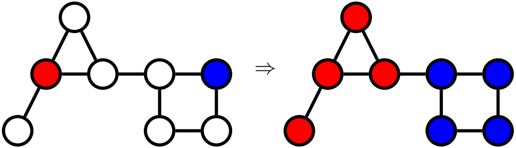
- Priori knowledge is encoded for l-th class as follows:

$$y_l(i) = \begin{cases} 1 & \text{if node } i \text{ belongs to } l\text{-th class} \\ 0 & \text{if not} \end{cases}$$

Experiments

Semi-Supervised Learning for Node Classification

- Problem: Given a few labels over the nodes, infer the others



- Priori knowledge is encoded for l-th class as follows:

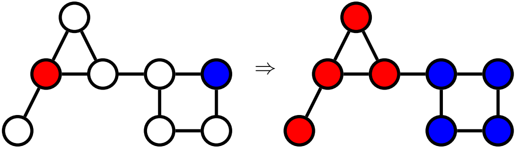
$$y_l(i) = \begin{cases} 1 & \text{if node } i \text{ belongs to } l\text{-th class} \\ 0 & \text{if not} \end{cases}$$

- The classification function f_l is assumed to be:

Experiments

Semi-Supervised Learning for Node Classification

- Problem: Given a few labels over the nodes, infer the others



- Priori knowledge is encoded for l-th class as follows:

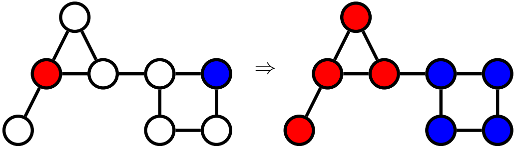
$$y_l(i) = \begin{cases} 1 & \text{if node } i \text{ belongs to } l\text{-th class} \\ 0 & \text{if not} \end{cases}$$

- The classification function f_l is assumed to be:
  - smooth on the graph

Experiments

Semi-Supervised Learning for Node Classification

- Problem: Given a few labels over the nodes, infer the others



- Priori knowledge is encoded for l-th class as follows:

$$y_l(i) = \begin{cases} 1 & \text{if node } i \text{ belongs to } l\text{-th class} \\ 0 & \text{if not} \end{cases}$$

- The classification function f_l is assumed to be:
  - smooth on the graph
  - close to y_l

Experiments

Semi-Supervised Learning for Node Classification

- One solution writes:

$$\mathbf{f}_l = \mathsf{D}^{1-\sigma}\mathsf{K}\mathsf{D}^{\sigma-1}\mathbf{y}_l \text{ with } \mathsf{K} = (\mathsf{Q}+\mathsf{L})^{-1}\mathsf{Q} \text{ and } \mathsf{Q} = \frac{\mu}{2}\mathsf{D}$$

Experiments

Semi-Supervised Learning for Node Classification

- One solution writes:

$$\mathbf{f}_l = \mathsf{D}^{1-\sigma}\mathsf{K}\mathsf{D}^{\sigma-1}\mathbf{y}_l \text{ with } \mathsf{K} = (\mathsf{Q}+\mathsf{L})^{-1}\mathsf{Q} \text{ and } \mathsf{Q} = \frac{\mu}{2}\mathsf{D}$$

- We can run our estimators to compute $\mathbf{f}_l$.

Experiments

Semi-Supervised Learning for Node Classification

- One solution writes:

$$\mathbf{f}_l = \mathsf{D}^{1-\sigma}\mathsf{K}\mathsf{D}^{\sigma-1}\mathbf{y}_l \text{ with } \mathsf{K} = (\mathsf{Q}+\mathsf{L})^{-1}\mathsf{Q} \text{ and } \mathsf{Q} = \frac{\mu}{2}\mathsf{D}$$

- We can run our estimators to compute $\mathbf{f}_l$.
- In the experiments, we generate a SBM with 3000 nodes and two equal-size communities:

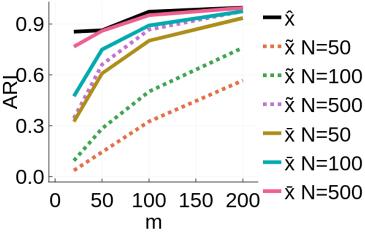
Experiments

Semi-Supervised Learning for Node Classification

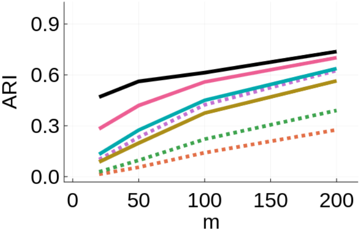
- One solution writes:

$$\mathbf{f}_l = \mathsf{D}^{1-\sigma}\mathsf{K}\mathsf{D}^{\sigma-1}\mathbf{y}_l \text{ with } \mathsf{K} = (\mathsf{Q}+\mathsf{L})^{-1}\mathsf{Q} \text{ and } \mathsf{Q} = \frac{\mu}{2}\mathsf{D}$$

- We can run our estimators to compute $\mathbf{f}_l$.
- In the experiments, we generate a SBM with 3000 nodes and two equal-size communities:



Strong Connections



Weak Connections

Conclusion and Future Works

Conclusion and Future Works

- We propose two Monte Carlo methods for graph signal smoothing.

Conclusion and Future Works

- We propose two Monte Carlo methods for graph signal smoothing.
- They scale linearly with the number of edges but also depend on q.

Conclusion and Future Works

- We propose two Monte Carlo methods for graph signal smoothing.
- They scale linearly with the number of edges but also depend on q.
- The links between RSFs and Laplacian-based numerical linear algebra are promising.