gipsa-lab

Grenoble | images | parole | signal | automatique | laboratoire

30th EUSIPCO EUROPEAN SIGNAL PROCESSING CONFERENCE
BELGRADE/SERBIA
29 AUG – 2 SEP 2022

Variance Reduction in Stochastic Methods For Large-Scale Regularized Least-Squares Problems

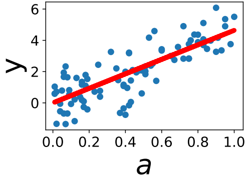
Yusuf Yiğit Pilavcı*
Pierre-Olivier Amblard
Simon Barthelmé
Nicolas Tremblay

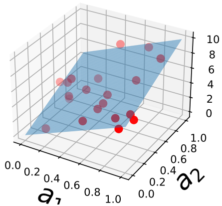
29/07/2022

cnrs Grenoble INP UNIVERSITÉ Grenoble Alpes UMR 5216

Regularized Least-Squares Problem

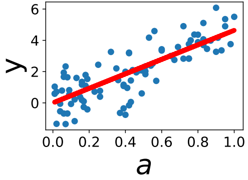
- Given the n data-measurement pairs $(a_{i,1}, \ldots, a_{i,p}, y_i)$'s, we seek for the best hyperplane that interprets the relation between the data and the measurements.

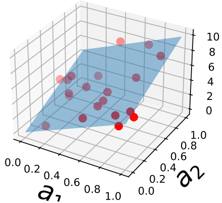




Regularized Least-Squares Problem

- Given the n data-measurement pairs $(a_{i,1}, \ldots, a_{i,p}, y_i)$'s, we seek for the best hyperplane that interprets the relation between the data and the measurements.





- This problem often takes the following form:

$$\hat{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^p} ||\mathsf{A}\mathbf{x} - \mathbf{y}||_2^2 + \lambda \mathbf{x}^\top \mathsf{P}\mathbf{x},$$

where $\lambda \mathbf{x}^\top \mathsf{P}\mathbf{x}$ is the regularization term.

Regularized Least-Squares Problem

- The closed-form solution can be exactly calculated at the cost of $\mathcal{O}(np^2)$.

Regularized Least-Squares Problem

- The closed-form solution can be exactly calculated at the cost of $\mathcal{O}(np^2)$.
- This is impractical when n and p are large.

Regularized Least-Squares Problem

- The closed-form solution can be exactly calculated at the cost of $\mathcal{O}(np^2)$.
- This is impractical when n and p are large.
- The approximate methods are often used:

Regularized Least-Squares Problem

- The closed-form solution can be exactly calculated at the cost of $\mathcal{O}(np^2)$.
- This is impractical when n and p are large.
- The approximate methods are often used:
  - Deterministic: Gradient descent algorithms.

Regularized Least-Squares Problem

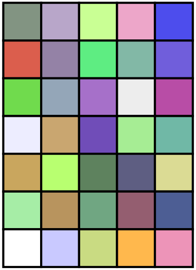
- The closed-form solution can be exactly calculated at the cost of $\mathcal{O}(np^2)$.
- This is impractical when n and p are large.
- The approximate methods are often used:
  - Deterministic: Gradient descent algorithms.
  - Randomized: Stochastic gradient descent.

Regularized Least-Squares Problem

- The closed-form solution can be exactly calculated at the cost of $\mathcal{O}(np^2)$.
- This is impractical when n and p are large.
- The approximate methods are often used:
  - Deterministic: Gradient descent algorithms.
  - Randomized: Stochastic gradient descent.
- Interesting alternatives are the algorithms based on determinantal point processes [DM21].

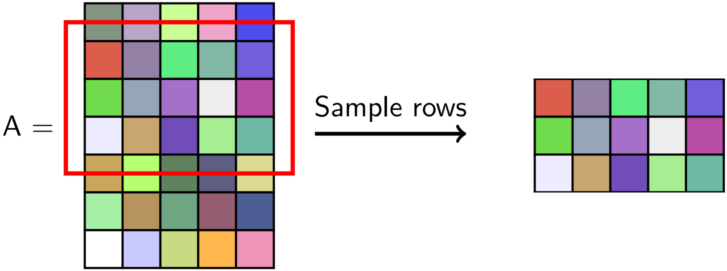
DPP-based Randomized Methods

- Assume $P = I$ for the simplicity,

$A =$ 

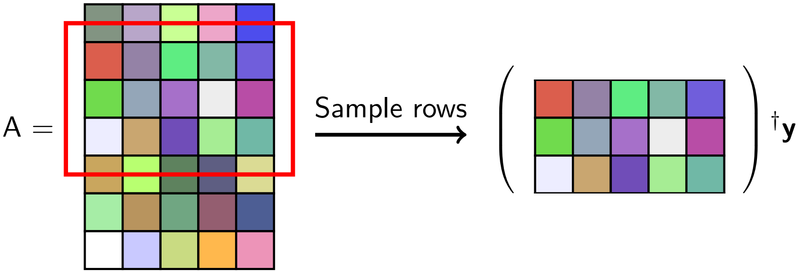
DPP-based Randomized Methods

- Assume $\mathsf{P} = \mathsf{I}$ for the simplicity,



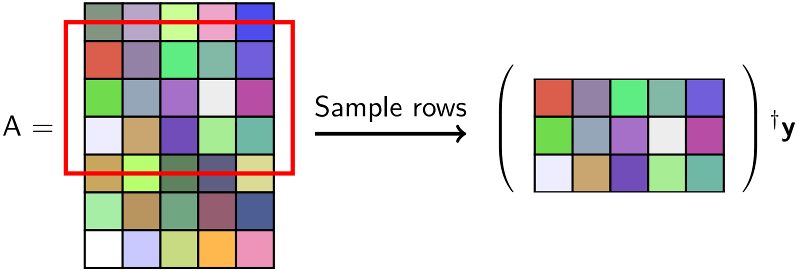
DPP-based Randomized Methods

- Assume $P = I$ for the simplicity,



DPP-based Randomized Methods

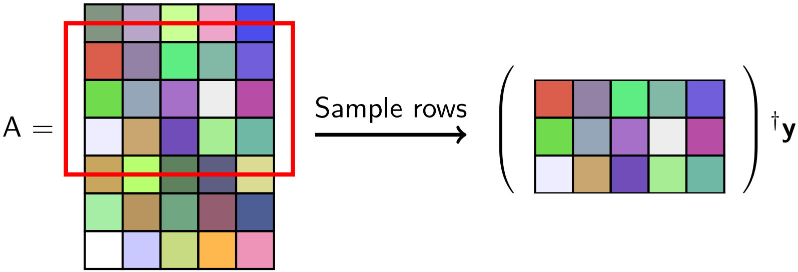
- Assume $\mathrm{P} = \mathrm{I}$ for the simplicity,



- They give unbiased estimates with tractable variance calculation.

DPP-based Randomized Methods

- Assume $\mathsf{P} = \mathsf{I}$ for the simplicity,



- They give unbiased estimates with tractable variance calculation.
- However, they have a slow convergence rate *i.e.* Monte Carlo rate $\mathcal{O}(N^{-1/2})$.

Main Idea

- Solving the optimization problem is equivalent to minimizing the following quadratic form:

$$F(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top Q\mathbf{x} - \mathbf{x}^\top \mathbf{r}.$$

Main Idea

- Solving the optimization problem is equivalent to minimizing the following quadratic form:

$$F(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top Q\mathbf{x} - \mathbf{x}^\top \mathbf{r}.$$

- The gradient descent algorithm draws the following iteration scheme:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla F(\mathbf{x}_k)$$

where $\alpha \in \mathbb{R}$ and $\nabla F(\mathbf{x}_k) = Q\mathbf{x}_k - \mathbf{r}$.

Main Idea

- Solving the optimization problem is equivalent to minimizing the following quadratic form:

$$F(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top Q\mathbf{x} - \mathbf{x}^\top \mathbf{r}.$$

- The gradient descent algorithm draws the following iteration scheme:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla F(\mathbf{x}_k)$$

  where $\alpha \in \mathbb{R}$ and $\nabla F(\mathbf{x}_k) = Q\mathbf{x}_k - \mathbf{r}$.
- Let $\tilde{\mathbf{x}}$ be the DPP estimator. A new estimator by applying a single step is:

$$\tilde{\mathbf{z}} := \tilde{\mathbf{x}} - \alpha(Q\tilde{\mathbf{x}} - \mathbf{r})$$

Main Idea

- If $\tilde{\mathbf{x}}$ is unbiased *i.e.* $\mathbb{E}[\tilde{\mathbf{x}}] = \mathrm{Q}^{-1}\mathbf{r}$, then $\tilde{\mathbf{z}}$ is also unbiased since:

$$\mathbb{E}[\tilde{\mathbf{z}}] = \mathbb{E}[\tilde{\mathbf{x}}] - \alpha(\mathrm{Q}\mathbb{E}[\tilde{\mathbf{x}}] - \mathbf{r}) = \mathrm{Q}^{-1}\mathbf{r}.$$

Main Idea

- If $\tilde{\mathbf{x}}$ is unbiased *i.e.* $\mathbb{E}[\tilde{\mathbf{x}}] = \mathrm{Q}^{-1}\mathbf{r}$, then $\tilde{\mathbf{z}}$ is also unbiased since:

$$\mathbb{E}[\tilde{\mathbf{z}}] = \mathbb{E}[\tilde{\mathbf{x}}] - \alpha(\mathrm{Q}\mathbb{E}[\tilde{\mathbf{x}}] - \mathbf{r}) = \mathrm{Q}^{-1}\mathbf{r}.$$

- For some values of α, one can guarantee that $\mathrm{Var}(\tilde{\mathbf{z}}) \leq \mathrm{Var}(\tilde{\mathbf{x}})$.

Main Idea

- If $\tilde{\mathbf{x}}$ is unbiased *i.e.* $\mathbb{E}[\tilde{\mathbf{x}}] = \mathrm{Q}^{-1}\mathbf{r}$, then $\tilde{\mathbf{z}}$ is also unbiased since:

$$\mathbb{E}[\tilde{\mathbf{z}}] = \mathbb{E}[\tilde{\mathbf{x}}] - \alpha(\mathrm{Q}\mathbb{E}[\tilde{\mathbf{x}}] - \mathbf{r}) = \mathrm{Q}^{-1}\mathbf{r}.$$

- For some values of α, one can guarantee that $\mathrm{Var}(\tilde{\mathbf{z}}) \leq \mathrm{Var}(\tilde{\mathbf{x}})$.
- Moreover, $\mathrm{Var}(\tilde{\mathbf{z}})$ is a quadratic function of α which is minimized at:

$$\alpha^\star = \frac{\mathrm{tr}(\mathrm{Cov}(\mathrm{Q}\tilde{\mathbf{x}}, \tilde{\mathbf{x}}))}{\mathrm{tr}(\mathrm{Cov}(\mathrm{Q}\tilde{\mathbf{x}}))}.$$

Main Idea

- If $\tilde{\mathbf{x}}$ is unbiased *i.e.* $\mathbb{E}[\tilde{\mathbf{x}}] = \mathsf{Q}^{-1}\mathbf{r}$, then $\tilde{\mathbf{z}}$ is also unbiased since:

$$\mathbb{E}[\tilde{\mathbf{z}}] = \mathbb{E}[\tilde{\mathbf{x}}] - \alpha(\mathsf{Q}\mathbb{E}[\tilde{\mathbf{x}}] - \mathbf{r}) = \mathsf{Q}^{-1}\mathbf{r}.$$

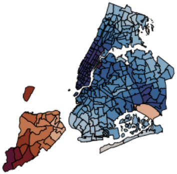
- For some values of α, one can guarantee that $\mathrm{Var}(\tilde{\mathbf{z}}) \leq \mathrm{Var}(\tilde{\mathbf{x}})$.
- Moreover, $\mathrm{Var}(\tilde{\mathbf{z}})$ is a quadratic function of α which is minimized at:

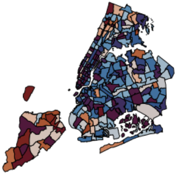
$$\alpha^\star = \frac{\mathrm{tr}(\mathrm{Cov}(\mathsf{Q}\tilde{\mathbf{x}}, \tilde{\mathbf{x}}))}{\mathrm{tr}(\mathrm{Cov}(\mathsf{Q}\tilde{\mathbf{x}}))}.$$

- In Monte Carlo literature, this way of reducing the variance is called control variate method.

Graph Tikhonov Regularization: A Use Case

Original Signal: $\mathbf{y}$:





Graph Tikhonov Regularization: A Use Case

Original Signal: $\mathbf{y}$: $\hat{\mathbf{x}}$:

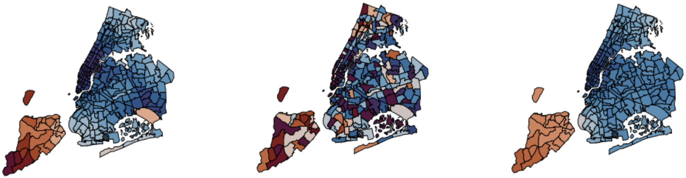


Figure: Median taxi fees paid in drop-off locations in NYC

Graph Tikhonov Regularization: A Use Case

Original Signal: $\mathbf{y}$: $\hat{\mathbf{x}}$:

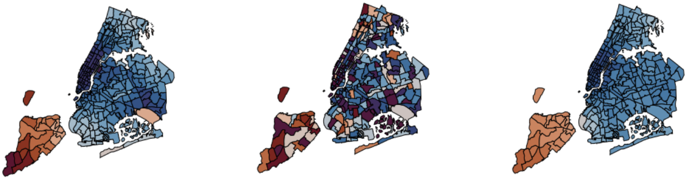


Figure: Median taxi fees paid in drop-off locations in NYC

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

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

where L is the graph Laplacian and $\mathbf{x}^T \mathrm{L} \mathbf{x} = \sum_{(i,j)\in\mathcal{E}} w(i,j)(x_i - x_j)^2$.

Graph Tikhonov Regularization: A Use Case

- The explicit solution to this problem is:

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

Graph Tikhonov Regularization: A Use Case

- The explicit solution to this problem is:

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

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

Graph Tikhonov Regularization: A Use Case

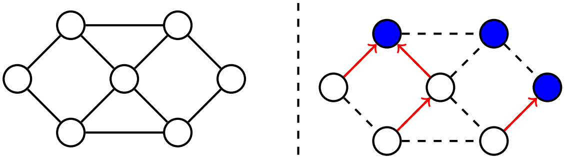
- The explicit solution to this problem is:

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

- Direct computation of K requires $\mathcal{O}(n^3)$ elementary operations due to the inverse.
- For large n, iterative methods and polynomial approximations are state-of-the-art. Both compute $\hat{\mathbf{x}}$ in linear time in the number of edges $|\mathcal{E}|$.
- In [Pil+21], we also propose a Monte Carlo algorithm for estimating $\hat{\mathbf{x}}$.

Random Spanning Forests

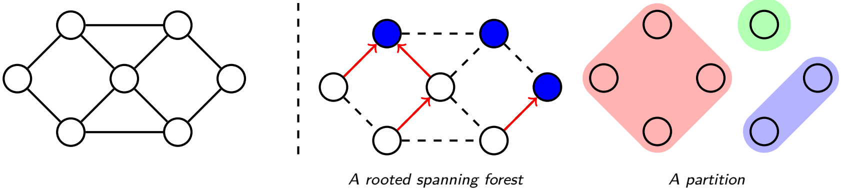
- A rooted spanning forest on a graph and its partition:



A rooted spanning forest

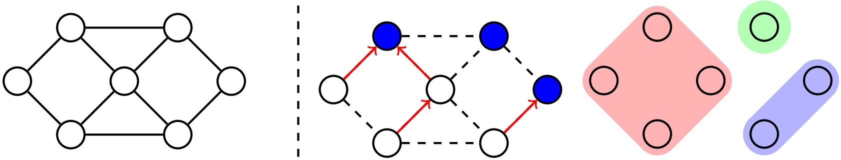
Random Spanning Forests

- A rooted spanning forest on a graph and its partition:



Random Spanning Forests

- A rooted spanning forest on a graph and its partition:

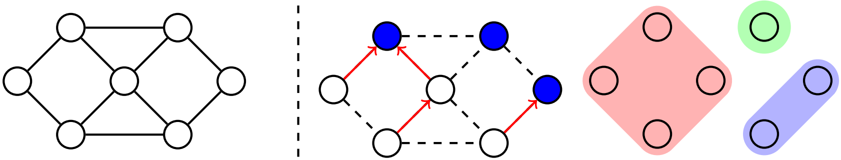


A rooted spanning forest *A partition*

- Random spanning forests is the process of randomly selecting a spanning forest over all possible forests.

Random Spanning Forests

- A rooted spanning forest on a graph and its partition:

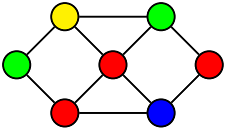


A rooted spanning forest

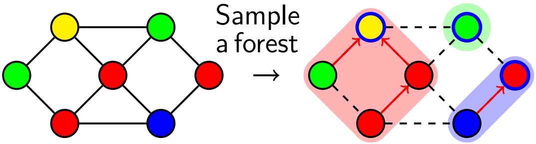
A partition

- Random spanning forests is the process of randomly selecting a spanning forest over all possible forests.
- For a particular distribution [AG13], we have useful links with graph-related algebra.

Forest-based Estimator

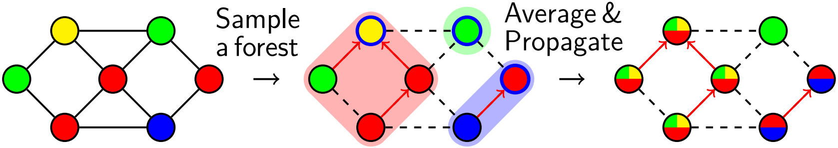


Forest-based Estimator



- Random partitions are sampled via random spanning forests.

Forest-based Estimator



- Random partitions are sampled via random spanning forests.
- This yields an unbiased estimator $\bar{\mathbf{x}}$.

Variance Reduction on the Forest Estimator

- Adapting the variance reduction idea, one has:

$$\bar{\mathbf{z}} := \bar{\mathbf{x}} - \alpha(\mathsf{K}^{-1}\bar{\mathbf{x}} - \mathbf{y}).$$

Variance Reduction on the Forest Estimator

- Adapting the variance reduction idea, one has:

$$\bar{\mathbf{z}} := \bar{\mathbf{x}} - \alpha(\mathsf{K}^{-1}\bar{\mathbf{x}} - \mathbf{y}).$$

- $\bar{\mathbf{z}}$ is unbiased.

Variance Reduction on the Forest Estimator

- Adapting the variance reduction idea, one has:

$$\bar{\mathbf{z}} := \bar{\mathbf{x}} - \alpha(\mathsf{K}^{-1}\bar{\mathbf{x}} - \mathbf{y}).$$

- $\bar{\mathbf{z}}$ is unbiased.
- A matrix-vector product with L is needed only once.

Variance Reduction on the Forest Estimator

- Adapting the variance reduction idea, one has:

$$\bar{\mathbf{z}} := \bar{\mathbf{x}} - \alpha(\mathsf{K}^{-1}\bar{\mathbf{x}} - \mathbf{y}).$$

- $\bar{\mathbf{z}}$ is unbiased.
- A matrix-vector product with L is needed only once.
- The optimal value for α is:

$$\alpha^\star = \frac{\mathrm{tr}(\mathrm{Cov}(\mathsf{K}^{-1}\bar{\mathbf{x}}, \bar{\mathbf{x}}))}{\mathrm{tr}(\mathrm{Cov}(\mathsf{K}^{-1}\bar{\mathbf{x}}))}.$$

Variance Reduction on the Forest Estimator

- Adapting the variance reduction idea, one has:

$$\bar{\mathbf{z}} := \bar{\mathbf{x}} - \alpha(\mathsf{K}^{-1}\bar{\mathbf{x}} - \mathbf{y}).$$

- $\bar{\mathbf{z}}$ is unbiased.
- A matrix-vector product with L is needed only once.
- The optimal value for α is:

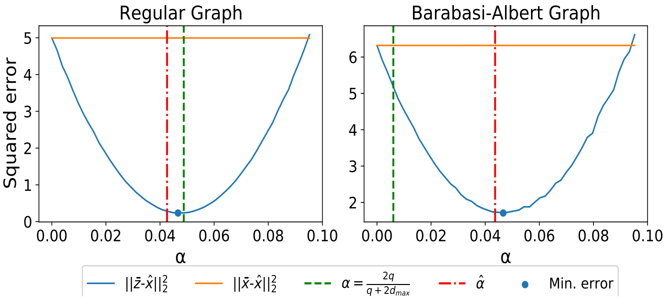
$$\alpha^\star = \frac{\mathrm{tr}(\mathrm{Cov}(\mathsf{K}^{-1}\bar{\mathbf{x}}, \bar{\mathbf{x}}))}{\mathrm{tr}(\mathrm{Cov}(\mathsf{K}^{-1}\bar{\mathbf{x}}))}.$$

- One can either choose a value for α from the safe range (*e.g.* $\alpha = \frac{2q}{q+2d_{max}}$) or estimate from the samples:

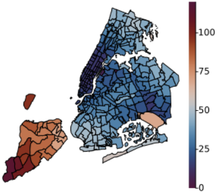
$$\hat{\alpha} = \frac{\mathrm{tr}(\widehat{\mathrm{Cov}}(\mathsf{K}^{-1}\bar{\mathbf{x}}, \bar{\mathbf{x}}))}{\mathrm{tr}(\widehat{\mathrm{Cov}}(\mathsf{K}^{-1}\bar{\mathbf{x}}))}.$$

Two choices of α

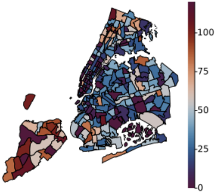
- We empirically compare these options of α over a regular and irregular graph:



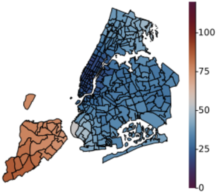
More Illustrations



Original Signal

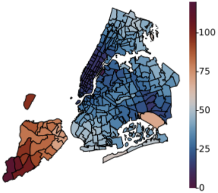


Noisy Measurements $\mathbf{y}$

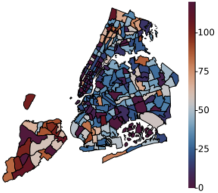


Exact solution $\hat{\mathbf{x}}$

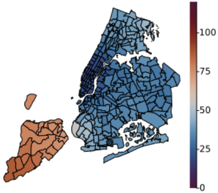
More Illustrations



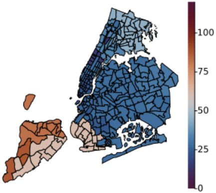
Original Signal



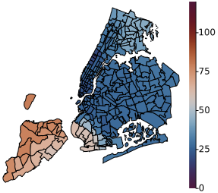
Noisy Measurements $\mathbf{y}$



Exact solution $\hat{\mathbf{x}}$



$\bar{\mathbf{x}}$



$\bar{\mathbf{z}}$

More Illustrations

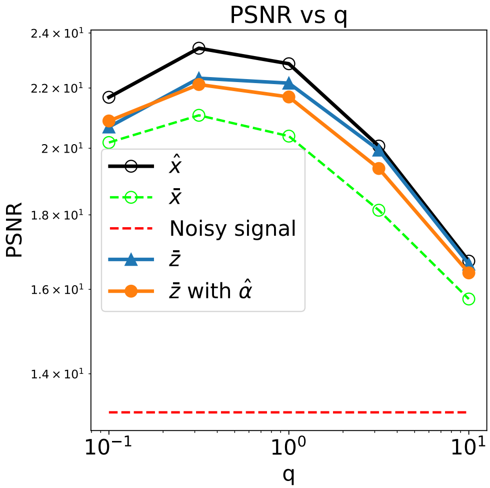


Figure: *PSNR vs q, N=2*

Future Work

- We propose a variance reduction technique for the DPP-based estimators to solve the regularized least squares problem

Future Work

- We propose a variance reduction technique for the DPP-based estimators to solve the regularized least squares problem
- We adapt this technique for a particular DPP-estimator for solving graph Tikhonov regularization problem.

Future Work

- We propose a variance reduction technique for the DPP-based estimators to solve the regularized least squares problem
- We adapt this technique for a particular DPP-estimator for solving graph Tikhonov regularization problem.
- There are several avenues to improve $\bar{\mathbf{z}} = \mathsf{T}\mathbf{y}$:
  - Using $\frac{1}{2}(\mathsf{T} + \mathsf{T}^\top)\mathbf{y}$,
  - Preconditioning with $\mathrm{diag}(K^{-1})$.

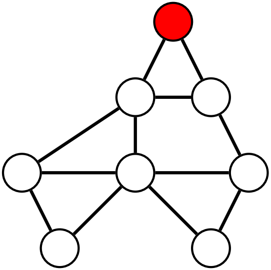
Random Spanning Forests

Definition (RSF)

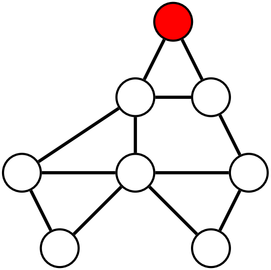
A random spanning forest Φ_q on a graph $\mathcal{G}$ is spanning forest selected over all spanning forests of $\mathcal{G}$ according to the following distribution:

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

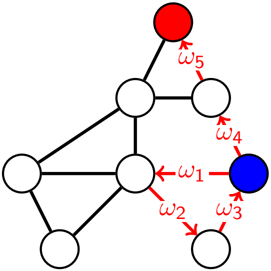
Wilson's Algorithm



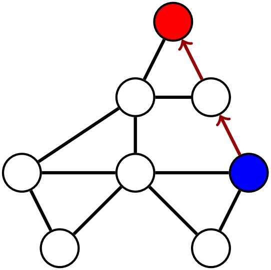
Wilson's Algorithm



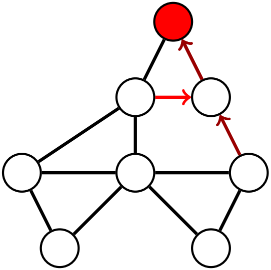
Wilson's Algorithm



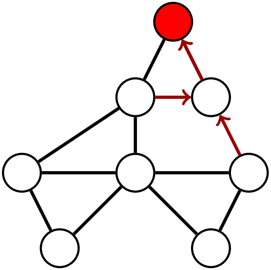
Wilson's Algorithm



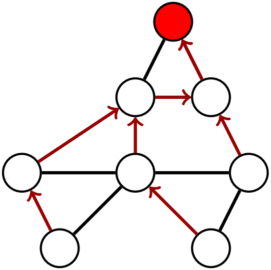
Wilson's Algorithm



Wilson's Algorithm



Wilson's Algorithm



Wilson's Algorithm