Stochastic Iterative Hard Thresholding for Graph-Structured Sparsity Optimization

Baojian Zhou[1,2], Feng Chen[1], and Yiming Ying[2]

[1]Department of Computer Science, [2]Department of Mathematics and Statistics, University at Albany, NY, USA

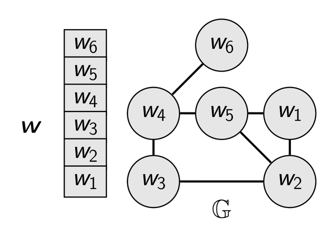
UNIVERSITY AT ALBANY
State University of New York

Structured Sparse Learning

Given $\mathcal{M}(\mathbb{M}) = \{\boldsymbol{w} : \text{supp}(\boldsymbol{w}) \in \mathbb{M}\}$, the structured sparse learning problems can be formulated as

$$\min_{\boldsymbol{w}\in\mathcal{M}(\mathbb{M})} F(\boldsymbol{w}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{w}), \text{ where}$$

- $F(\boldsymbol{w})$ is a convex loss such as least square, logistic loss, ...
- $\mathcal{M}(\mathbb{M})$ models structured sparsity such as connected subgraphs, dense subgraphs, and subgraphs isomophic to a query graph, ...



Previous Work

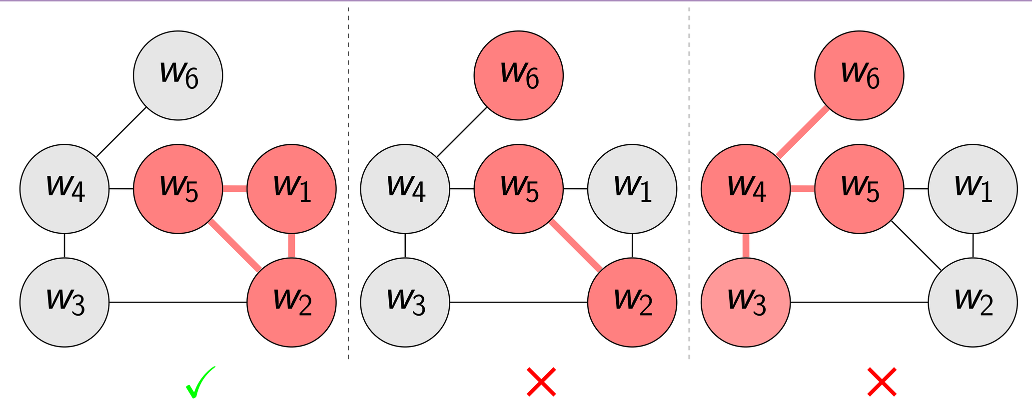


Figure: Weighted Graph Model $\mathbb{M} = \{S : |S| \le 3, S \text{ is connected }\}$ Hegde et al. (2015a).

To solve above problem under sparsity constraint, Nguyen et al. (2017) proposed Stochastic Iterative Hard Thresholding (StoIHT). At time t, StoIHT choose ξ_t from $[n]$ with probability p_{ξ_t} and project $\boldsymbol{w}^t$ onto a subspace

$$\boldsymbol{w}^{t+1} = \mathrm{P}(\boldsymbol{w}^t - \eta_t \nabla f_{\xi_t}(\boldsymbol{w}^t), \Gamma^t),$$

where the orthogonal projection $\mathrm{P}(\cdot, \Gamma)$ is defined as

$$\mathrm{P}(\boldsymbol{w}, \Gamma) := \arg\min_{\boldsymbol{w}'\in\mathcal{R}(\Gamma)} \|\boldsymbol{w} - \boldsymbol{w}'\|_2^2.$$

Why stochastic?

- More steady
- Less computation per-iteration

Two issues of StoIHT

- Cannot handle graph-structured constraint
- Ideally, $\nabla f_{\xi_t}(\boldsymbol{w}^t)$ also needs to be in a subspace

Our Algorithm

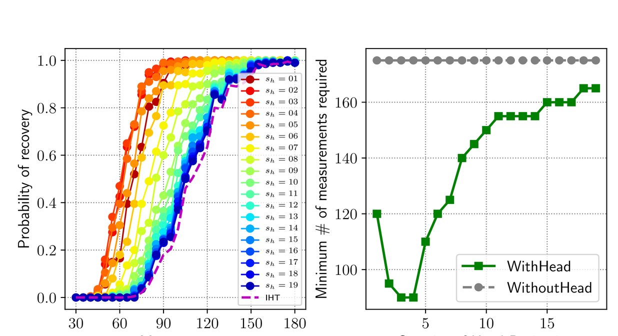
The hybird of Nguyen et al. (2017) and Hegde et al. (2016).

Algorithm 1 GraphStoIHT

1: **Input**: $\eta_t, F(\cdot), \mathbb{M}_{\mathcal{H}}, \mathbb{M}_{\mathcal{T}}$
2: **Initialize**: $\boldsymbol{w}^0$ and $t = 0$
3: **for** $t = 0, 1, 2, \ldots$ **do**
4: Choose ξ_t from $[n]$ with prob. p_{ξ_t}
5: $\boldsymbol{b}^t = \mathrm{P}(\nabla f_{\xi_t}(\boldsymbol{w}^t), \mathbb{M}_{\mathcal{H}})$
6: $\boldsymbol{w}^{t+1} = \mathrm{P}(\boldsymbol{w}^t - \eta_t \boldsymbol{b}^t, \mathbb{M}_{\mathcal{T}})$
7: **end for**
8: **Return** $\boldsymbol{w}^{t+1}$

Why projection $\boldsymbol{b}^t = \mathrm{P}(\nabla f_{\xi_t}(\boldsymbol{w}^t), \mathbb{M}_{\mathcal{H}})$?

- Both of them solve the same projection problem
- Sparsity is both in primal space and dual space
- Remove some noisy directions at the first stage



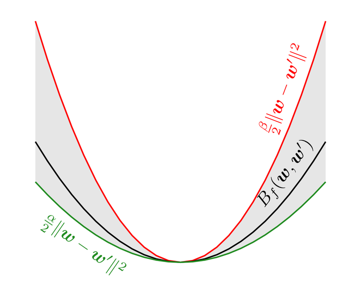
Convergence Analysis

Define the Bregman divergence of f as

$$B_f(\boldsymbol{w}, \boldsymbol{w}') = f(\boldsymbol{w}) - f(\boldsymbol{w}') - \langle \nabla f(\boldsymbol{w}'), \boldsymbol{w} - \boldsymbol{w}' \rangle$$

Two assumptions in $\mathcal{M}(\mathbb{M})$:

- $f_i(\boldsymbol{w})$: β-Restricted Strong Smoothness
  $F(\boldsymbol{w})$: α-Restricted Strong Convexity
- Efficient Approximated projections:
  - $\mathrm{P}(\cdot, \mathbb{M}_{\mathcal{H}})$ with approximation factor $c_{\mathcal{H}}$
  - $\mathrm{P}(\cdot, \mathbb{M}_{\mathcal{T}})$ with approximation factor $c_{\mathcal{T}}$



Theorem 1 (Linear Convergence) Let $\boldsymbol{w}^0$ be the start point and choose $\eta_t = \eta$, then $\boldsymbol{w}^{t+1}$ of Algorithm 1 satisfies

$$\mathbb{E}_{\xi_{[t]}}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^*\| \le \kappa^{t+1}\|\boldsymbol{w}^0 - \boldsymbol{w}^*\| + \frac{\sigma}{1-\kappa},$$

where $\eta, \tau \in (0, 2/\beta)$ and

$$\kappa = (1 + c_{\mathcal{T}})\left(\sqrt{\alpha\beta\eta^2 - 2\alpha\eta + 1} + \sqrt{1 - \alpha_0^2}\right),$$

$$\alpha_0 = c_{\mathcal{H}}\alpha\tau - \sqrt{\alpha\beta\tau^2 - 2\alpha\tau + 1}, \quad \beta_0 = (1 + c_{\mathcal{H}})\tau,$$

$$\sigma = \left(\frac{\beta_0}{\alpha_0} + \frac{\alpha_0\beta_0}{\sqrt{1-\alpha_0^2}}\right)\mathbb{E}_{\xi_t}\|\nabla_I f_{\xi_t}(\boldsymbol{w}^*)\| + \eta\mathbb{E}_{\xi_t}\|\nabla_I f_{\xi_t}(\boldsymbol{w}^*)\|.$$

Graph Sparse Linear Regression

Given a design matrix $\boldsymbol{X} \in \mathbb{R}^{m\times p}$ and corresponding observed noisy vector $\boldsymbol{y} \in \mathbb{R}^m$ that are linked via the linear relationship

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_m)$. To estimate $\boldsymbol{w}^*$, consider the least square loss and formulate it as

$$\arg\min_{\text{supp}(\boldsymbol{w})\in\mathcal{M}(\mathbb{M})} F(\boldsymbol{w}) := \frac{1}{n}\sum_{i=1}^{n} \frac{n}{2m}\|\boldsymbol{X}_{B_i}\boldsymbol{w} - \boldsymbol{y}_{B_i}\|^2,$$

where m observations have been partitioned into n blocks, $B_1, B_2, \ldots, B_n$. Let $\alpha = 1 - \delta, \beta = 1 + \delta$.

Algorithm	κ
GraphIHT	$(1 + c_{\mathcal{T}})\left(\sqrt{\delta} + 2\sqrt{1-\delta}\right)\sqrt{\delta}$
GraphStoIHT	$(1 + c_{\mathcal{T}})\left(\sqrt{\frac{2}{1+\delta}} + \frac{2\sqrt{2(1-\delta)}}{1+\delta}\right)\sqrt{\delta}$

κ of GraphIHT is controlled by $\mathcal{O}(\sqrt{\delta}\cdot 2(1 + c_{\mathcal{T}}))$ while for GraphStoIHT, κ is controlled by $\mathcal{O}(\sqrt{\delta}\cdot 3\sqrt{2}(1 + c_{\mathcal{T}}))$. To obtain $\kappa < 1$, $\delta \le 0.0527$ for GraphIHT while $\delta \le 0.0142$ for GraphStoIHT. The gap between the two κ is mainly due to the randomness introduced in our algorithm.

Graph Sparse Logistic Regression

Given a dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^m$, the graph logistic regression is formulated as the following problem

$$\arg\min_{\text{supp}(\boldsymbol{w})\in\mathcal{M}(\mathbb{M})} F(\boldsymbol{w}) := \frac{1}{n}\sum_{i=1}^{n}\frac{n}{m}\sum_{j=1}^{m/n} h(\boldsymbol{w}, i_j) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2,$$

where $h(\boldsymbol{w}, i_j) = \log(1 + \exp(-y_{i_j}\cdot\langle \boldsymbol{x}_{i_j}, \boldsymbol{w}\rangle))$. Problem above has an important application on gene pathway analysis. If each sample $\boldsymbol{a}_i$ is normalized, then $F(\boldsymbol{x})$ satisfies λ-RSC and each $f_i(\boldsymbol{x})$ satisfies $(\alpha + (1+\nu)\theta_{max})$-RSS. The condition of $\kappa < 1$ is

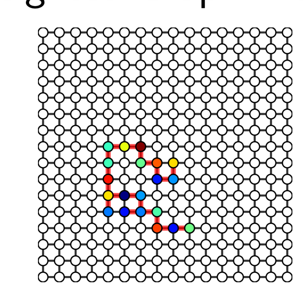
$$\frac{\lambda}{\lambda + n(1+\nu)\theta_{max}/4m} \ge \frac{243}{250},$$

with probability $1 - p\exp(-\theta_{max}\nu/4)$, where $\theta_{max} = \lambda_{max}(\sum_{j=1}^{m/n}\mathbb{E}[\boldsymbol{a}_{i_j}\boldsymbol{a}_{i_j}^{\mathsf{T}}])$ and $\nu \ge 1$.

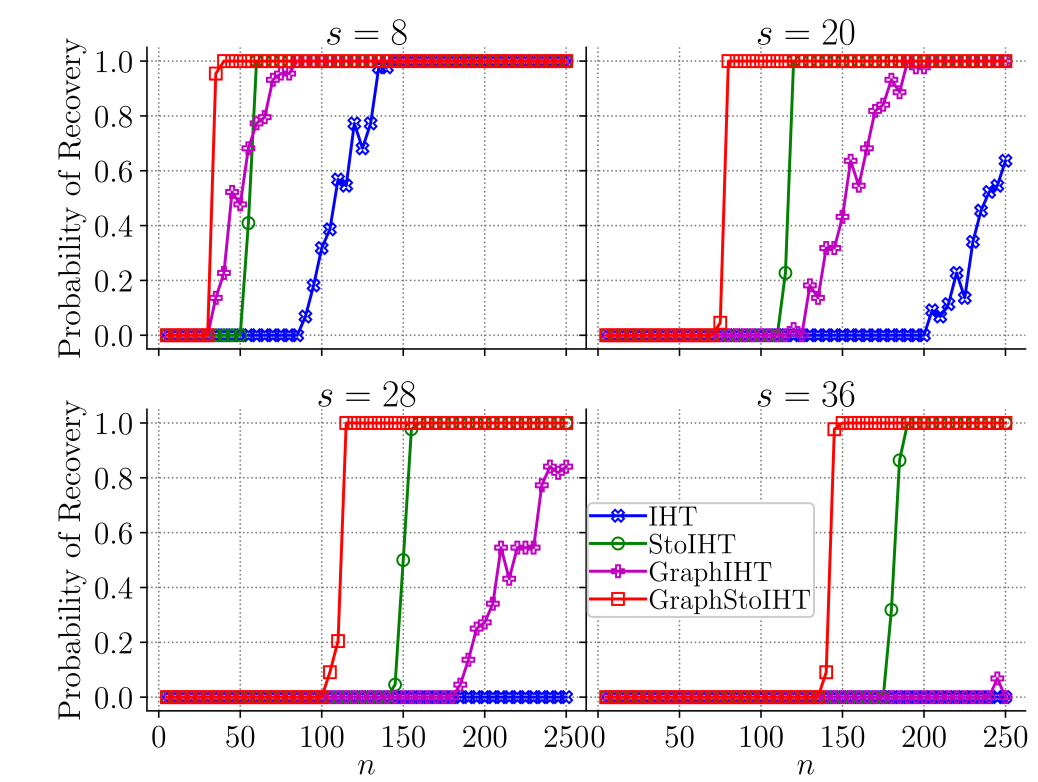
Experiments

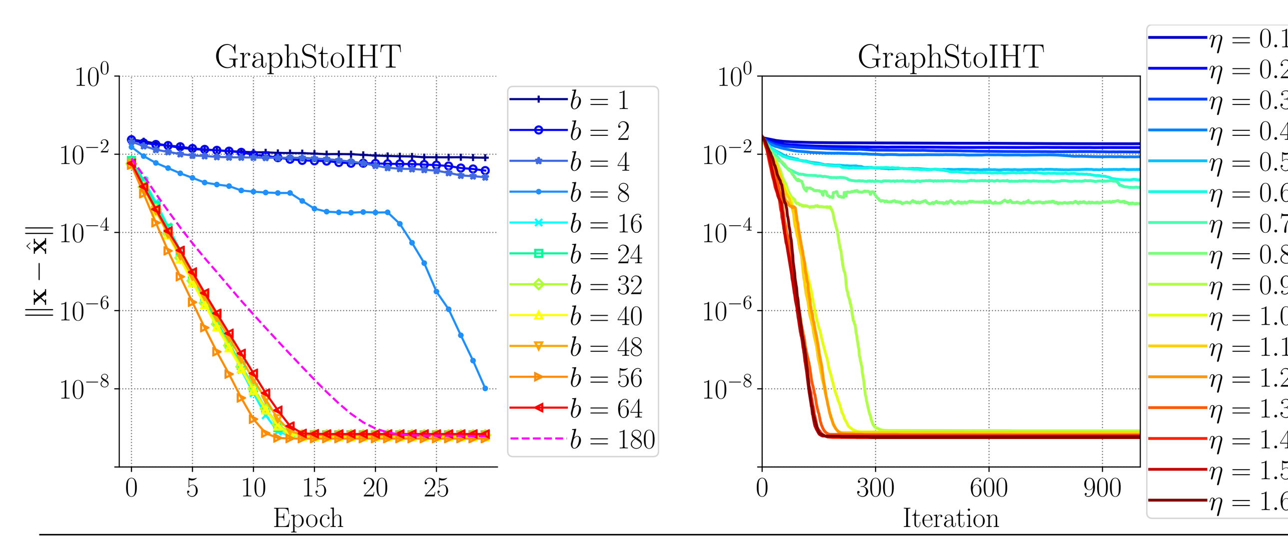
Simulation Dataset:

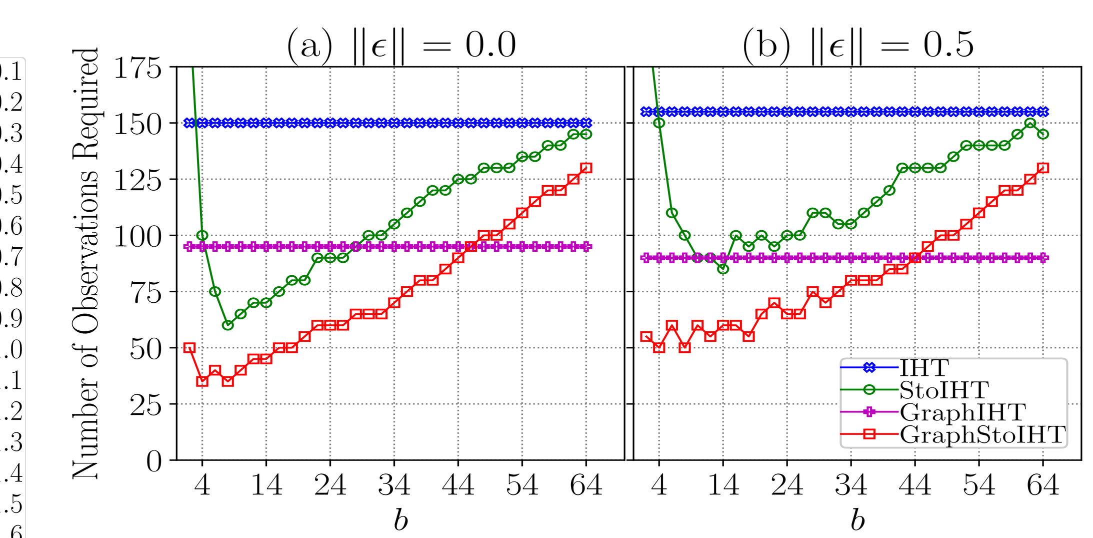
- Each entry $\sqrt{m}\boldsymbol{X}_{ij} \sim \mathcal{N}(0, 1)$
- $\text{Supp}(\boldsymbol{w}^*)$ is generated by random walk
- Entries of $\boldsymbol{w}^*$ from $\mathcal{N}(0, 1)$
- Weighted Graph Model



- **Upper Right:** Probability of recovery on synthetic dataset. The probability of recovery is a function of number of observations m.
- **Lower Left:** The left part illustrates the *estimation error* as a function of epochs for different choices of b. When $b = 180$, it degenerates to GraphIHT (the dashed line). The right part shows the *estimation error* as a function of iterations for different choices of η.
- **Lower Right:** Robustness to noise ϵ. The number of observations required is a function of different block sizes.







Real image dataset:

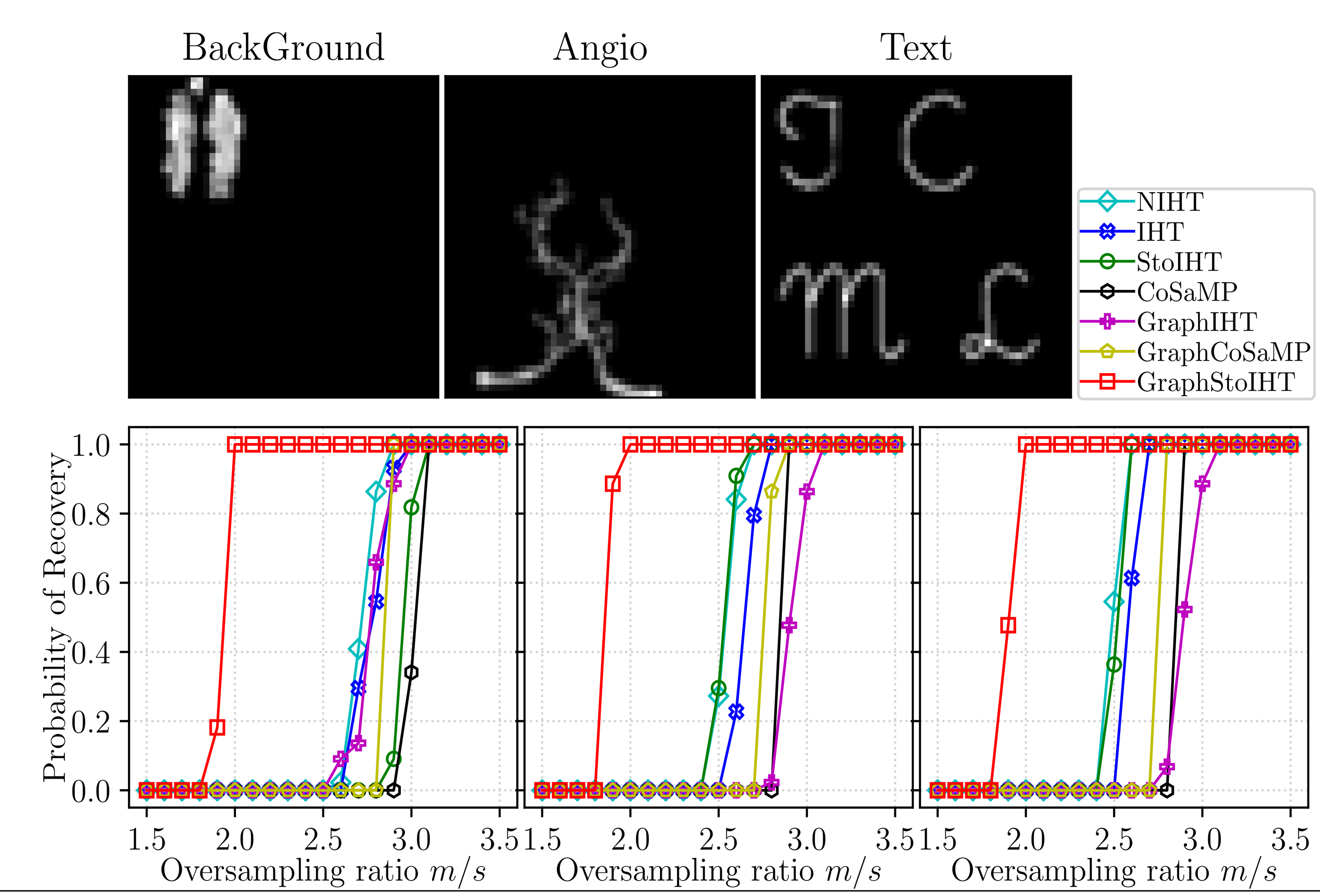
- IHT (Blumensath and Davies, 2009)
- StoIHT (Nguyen et al., 2017)
- NIHT (Blumensath and Davies, 2010)
- CoSaMP (Needell and Tropp, 2009)
- GraphIHT (Hegde et al., 2016) + WGM
- GraphCoSaMP (Hegde et al., 2015b)

Experimental settings:

- Resized real images (Hegde et al., 2015b)
- η of IHT-based in $\{0.2, 0.4, 0.6, 0.8\}$
- b of StoIHT-based in $\{m/5, m/10\}$
- Tune b and η on 100 observations.
- $\boldsymbol{A}$ used here is Gaussian matrix

Two experimental conclusions:

- SGD-based methods are more stable
- Capture the graph-structured sparsity



Breast Cancer Dataset:

- 295 samples with 78 positives (metastatic) and 217 negatives (non-metastatic) provided in Van De Vijver et al. (2002).
- PPI network with 637 pathways is provided in Jacob et al. (2009).

Four ℓ^1/ℓ^2 mixed norm-based algorithms:

- ℓ^1-Pathway uses pathways as groups
- ℓ^1/ℓ^2-Pathway uses pathways as groups
- ℓ^1-Edge uses use edges as groups
- ℓ^1/ℓ^2-Edge uses use edges as groups

Algorithm	Cancer related genes	$\|\boldsymbol{w}^t\|_0$	AUC
GraphStoIHT	BRCA2, CCND2, CDKN1A, ATM, AR, TOP2A	051.7	**0.715**
GraphIHT	ATM, CDKN1A, BRCA2, AR, TOP2A	055.2	0.714
ℓ^1-Path	BRCA1, CDKN1A, ATM, DSC2	061.2	0.675
StoIHT	MKI67, NAT1, AR, TOP2A	059.6	0.708
ℓ^1/ℓ^2-Edge	CCND3, ATM, CDH3	051.4	0.705
ℓ^1-Edge	CCND3, AR, CDH3	039.9	0.698
ℓ^1/ℓ^2-Path	BRCA1, CDKN1A	147.6	0.705
IHT	NAT1, TOP2A	067.9	0.707

Conclusion and Future Work

- We proposed GraphStoIHT.
- It enjoys a linear convergence property.
- Two real-world applications.

In future, it would be interesting to see if one can apply the variance reduction techniques such as SAGA (Defazio et al., 2014) and SVRG (Johnson and Zhang, 2013) to GraphStoIHT.

Code & Datasets

- Code & Datasets can be found at GitHub: https://github.com/baojianzhou/graph-sto-iht
- Email: bzhou6@albany.edu
- **Baojian Zhou is open to postdoc positions.**