Georgia Tech Mathematics
College of Sciences

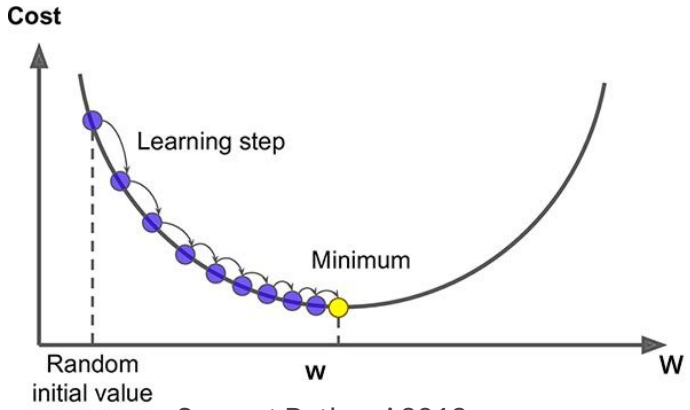
Dynamical Characterization & Analysis of the Optimization Algorithms: Linearized Bregman and Iterative Shrinkage Thresholding Algorithm

Yihua Xu and Ariba Khan
Mentor: Dr. Rachel Kuske

Gradient Descent (GD)

What is GD? One of the most common optimization algorithm used to calculate the minimum value of a differentiable function.

$$x_{k+1} = x_k - t_k \, A^T \, (Ax_k - b)$$



Sauggat Batharai 2018

Basic Algorithm Process:

1. x is a minimizer of ||Ax-b||
2. Set random initial condition for x
3. Iterate over the time step
4. Find the local minimum

2 Ways of Measuring Algorithm Behavior:

Normalized Residual $\mathcal{R}(x_k) = \frac{\|Ax_k - b\|_2}{\|b\|_2}.$

Model Error $M(x_k) = \frac{\|x_{True} - x_k\|_2}{\|x_{True}\|_2}$

**Other optimization algorithms have been developed based on GD that deal with different data types

Methods Introduction

LB & ISTA are both based on Gradient Descent but deal with sparse solutions

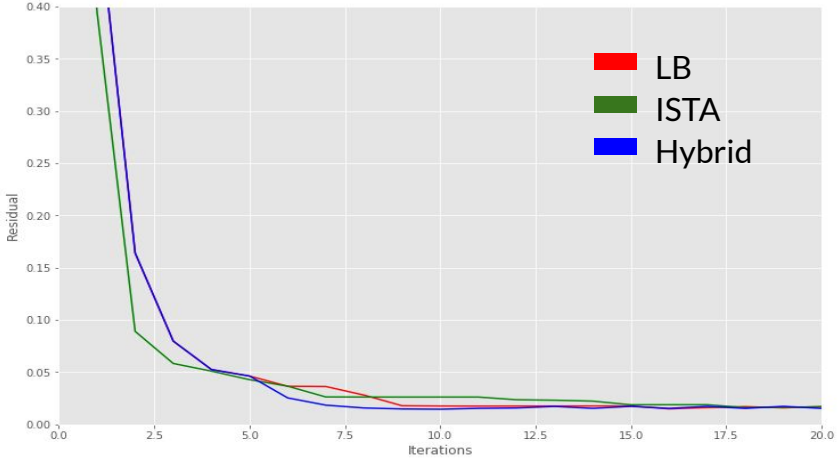
	Linearized Bregman (LB)	Iterative Shrinkage Thresholding Algorithm (ISTA)
Iterative Formula	$z_{k+1} = z_k - t_k\ A^T\ (Ax_k - b)$ $x_{k+1} = S_\lambda\ (z_{k+1})$	$z_{k+1} = x_k - t_k\ A^T\ (Ax_k - b)$ $x_{k+1} = S_\lambda\ (z_{k+1})$
Time Step	$t_k = \frac{\|Ax_k - b\|_2^2}{\|A^\top(Ax_k - b)\|_2^2}.$	$t_k = \frac{1}{\|A\|_2^2}$
Shrinkage Process	$[S_\lambda(z_{k+1})]_i = \begin{cases} z_i - \lambda & \text{if } z_i > \lambda \\ 0 & \text{if } -\lambda \le z_i \le \lambda \\ z_i + \lambda & \text{if } z_i < -\lambda \end{cases}$	

Variable Definitions & Data Setup

Variable	Definition	Data Setup
k	Number of iterations	Chose varying k, we will specify in the following slides
A	A matrix; normally a tall matrix because in the real world there are often many variables to consider	800*70 matrix of normally distributed random variables
b	Created by the multiplication of A and xTrue, with added noise	A * xTrue + (0.1 * noise)
xTrue	The correct value, the algorithm's final output should be around this vector value	A size 70 vector created with a random exponential decay, including large, small, and zero entries; this is done to mimic sparse solutions in the real world
Noise	Source of chatter	random normal distribution with mean=0 and variance=1
t_k	Time step	Dynamic time step for LB and constant time step for ISTA
x_0	Initial guess for x	Vector of zeros

Preview of Our Findings & Applications

- Linearized Bregman performs best with large entries and a large λ
- ISTA performs best with small entries and small λ
- Linearized Bregman outperforms ISTA when subsampling
- Hybrid Method is a combination of both methods and outperforms both LB and ISTA when dealing with a mix of entry sizes



Real World Applications

Machine Learning

Imaging

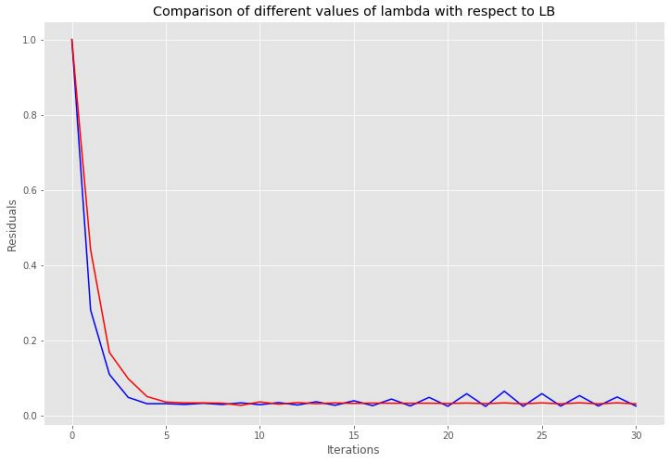
Subsampling makes problem solvable

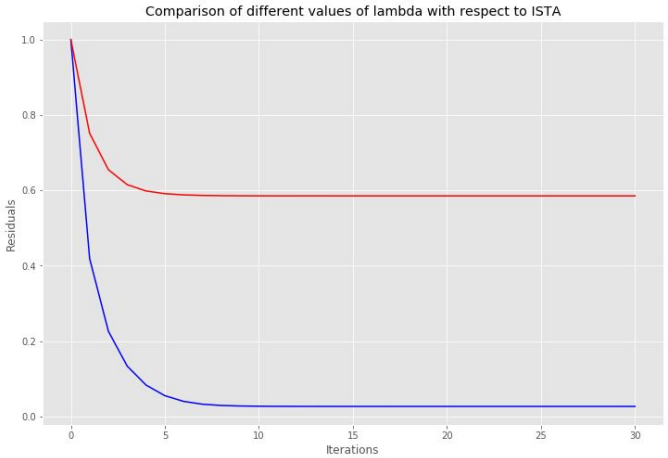
Inverse Parameter Fitting

Choices concerning hyperparameter λ

0.01
1

LB





ISTA

Though converging slightly quicker early on when λ is small, it would lead to overfitting/ fluctuation later on. Overall, a moderate or comparatively large λ is most suitable for LB, a reasonable range could be from 0.5 - 1.

$$z_{k+1} = z_k - t_k \, A^T \, (Ax_k - b)$$
$$x_{k+1} = S_\lambda \, (z_{k+1})$$

Obviously, if we put a large λ in ISTA, the residual would decrease to a very limited extent. (Basically predicting most entries as zero). ISTA needs a very small λ to reach a low residual. A good value for λ_{ISTA} is around 0.01.

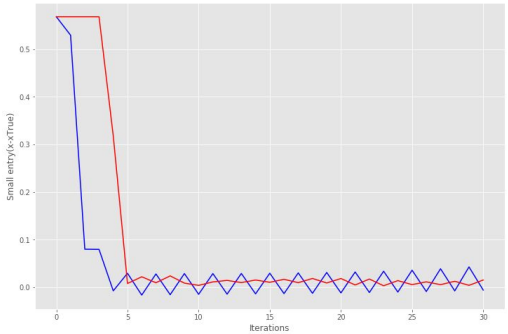
$$z_{k+1} = x_k - t_k \, A^T \, (Ax_k - b)$$
$$x_{k+1} = S_\lambda \, (z_{k+1})$$

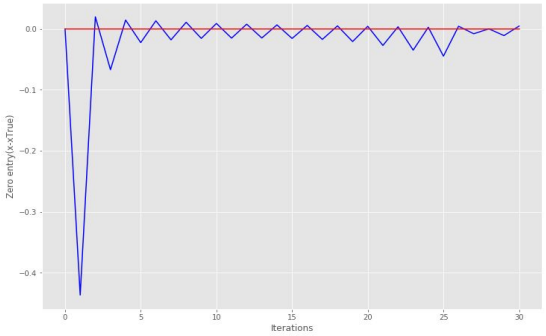
Choices concerning hyperparameter λ

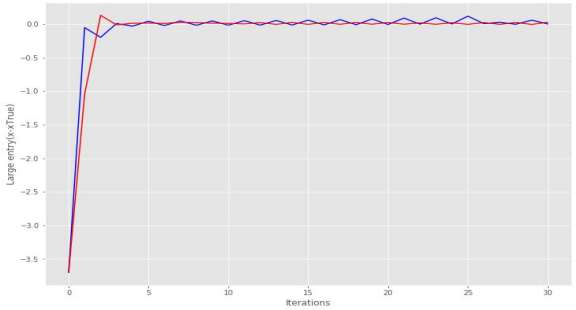
0.01
1

Small Entries | Zero Entries | Large Entries

LB

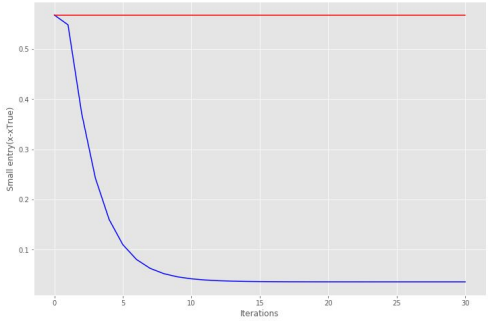


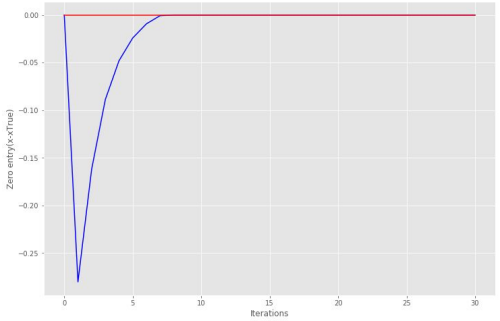


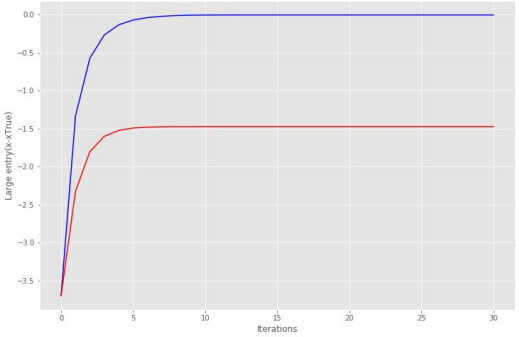


For large entries, The difference between different λ is very small early on in the iterations and overall convergence rate is quick, while comparatively larger λ could result in smaller chatter later on.

ISTA







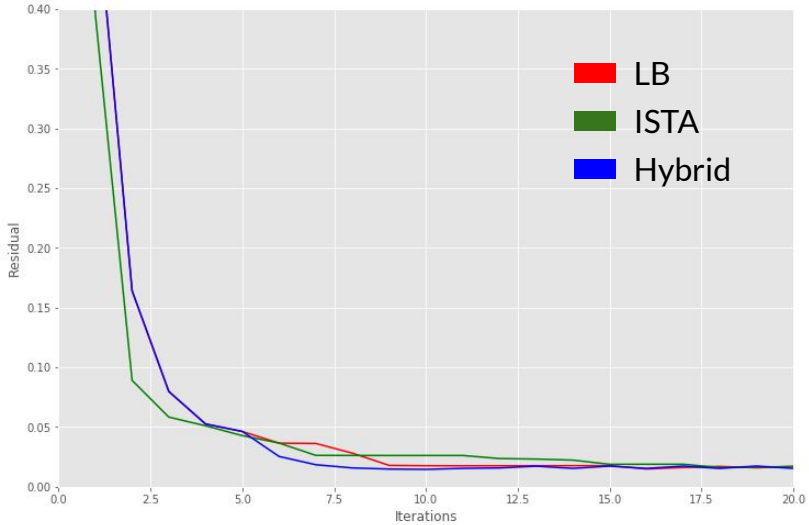
For small entries and small λ, ISTA tends to predict well and converge quickly without fluctuation in later iterations.

The Hybrid Method

General Idea: x_{hybrid} follows x_{lb} for the first K iterations, then smaller entries follow ISTA and larger entries follow LB

Base Case Scenario:

$\lambda_{LB} = 1$
$\lambda_{ISTA} = 0.01$
$K = 5$
$K_{max} = 20$
$X_{mid} = 1.0$

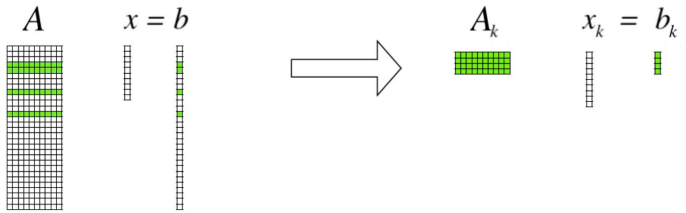


Observations:

- As the algorithm states, for first five iterations, the Hybrid method is exactly same as LB, thus following the same pattern.
- After five iterations, the residual for Hybrid method starts to be lower than other two methods.
- As a result, the Hybrid method makes sense and we will include the Hybrid method in following analysis.

Subsampling .vs. No Subsampling

What is subsampling? Taking a different random subset of the data by choosing k rows out of entire set for each iteration.



Why subsampling? In real world setting, iterating over the entire dataset usually mean a very time-consuming process. By subsampling, we strive to solve problems in shorter periods of time with only few full datapasses.

Why is LB more suited for subsampling?
ISTA leaves out certain information, because it uses x_k, which undergoes a shrinkage process every iteration, also since A_k is mostly different from iteration to iteration when subsampling, when throwing out information, the impact would be more severe than using the complete dataset.

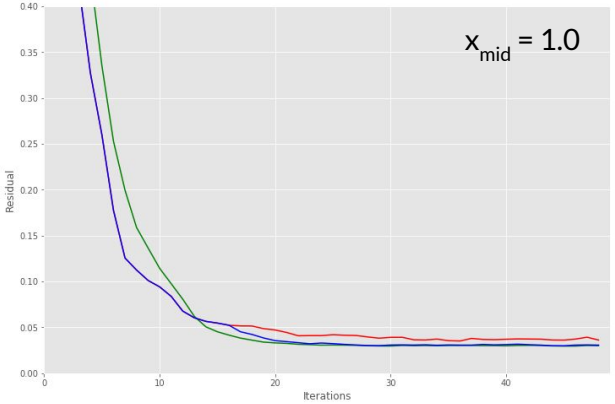
$$z_{k+1} = x_k - t_k \, A_k^T \, (A_k x_k - b_k)$$
$$x_{k+1} = S_\lambda \, (z_{k+1})$$

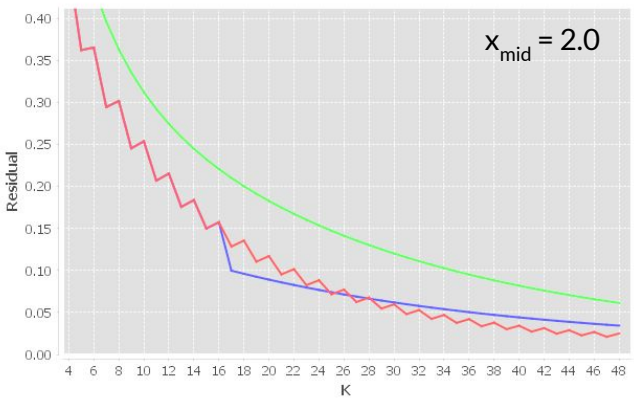
while LB retains all information which naturally makes LB more fitted for subsampling

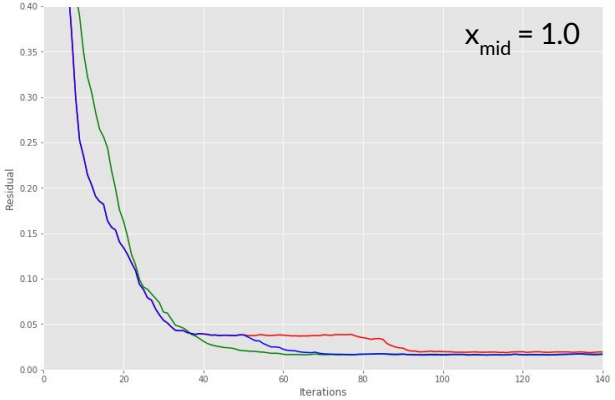
$$z_{k+1} = z_k - t_k \, A_k^T \, (A_k x_k - b_k)$$
$$x_{k+1} = S_\lambda \, (z_{k+1})$$

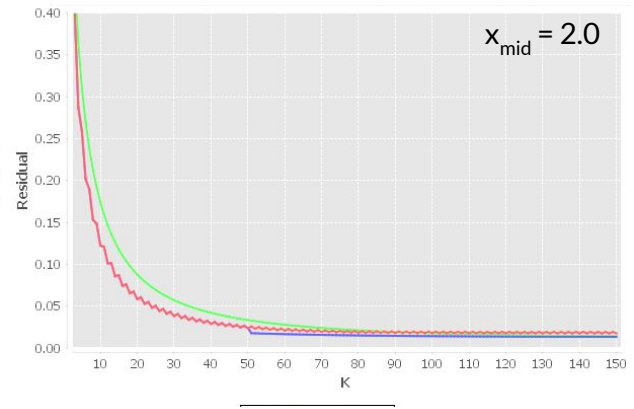
After Subsampling

LB
ISTA
Hybrid









Parameters :

- K= 16(top) / 50 (bottom)
- K_{max} = 48(top) / 150 (bottom)
- Rows per iteration = 50 (top)/ 16 (bottom)
- Noise = 0.1 * random normal variable

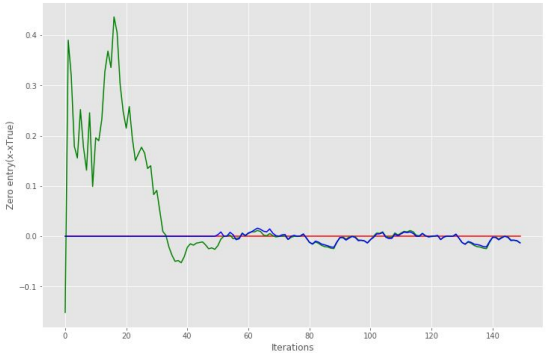
Observations from both setups:

- Consistent with the analytical result, Linearized Bregman converges much quicker than ISTA when subsampling
- Hybrid method has a lower residual than ISTA and LB since it is able to perform well with both large and small entries

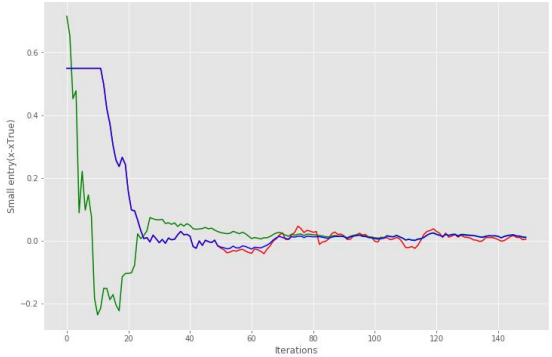
Comparing Entries

x-xTrue

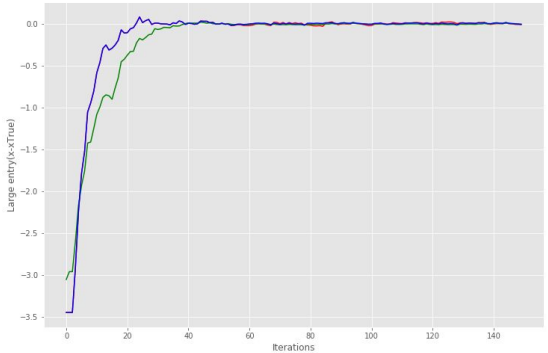
Zero Entries



Small Entries

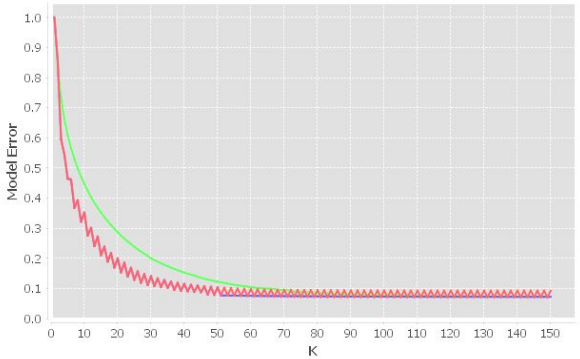


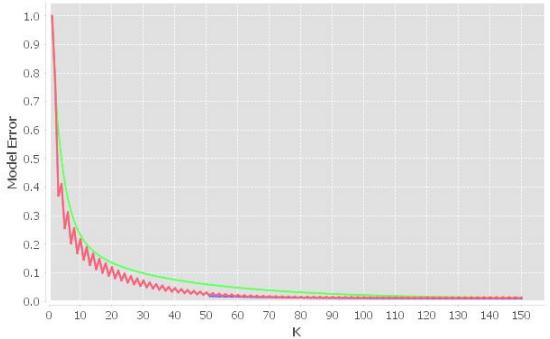
Large Entries



undefined

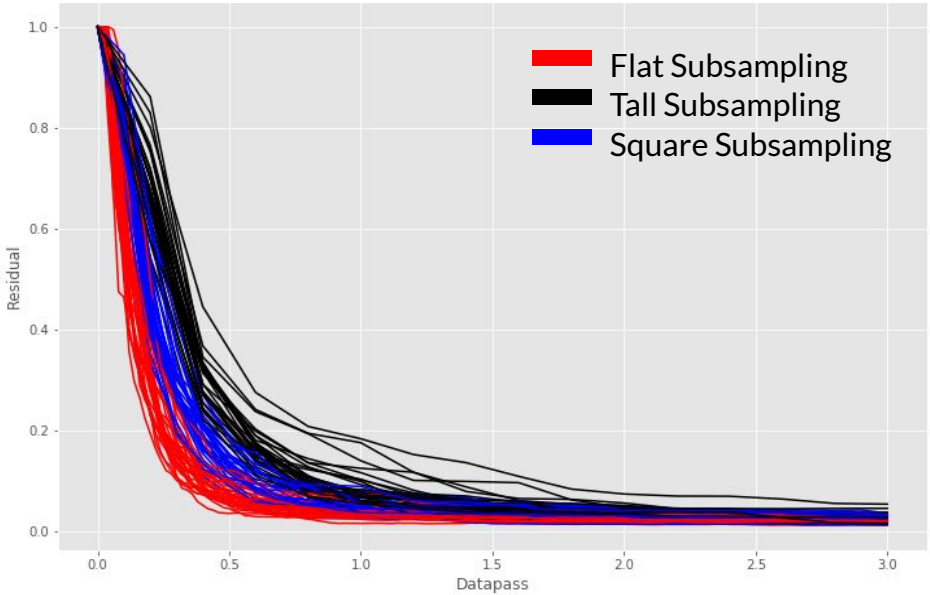
- LB
- ISTA
- Hybrid





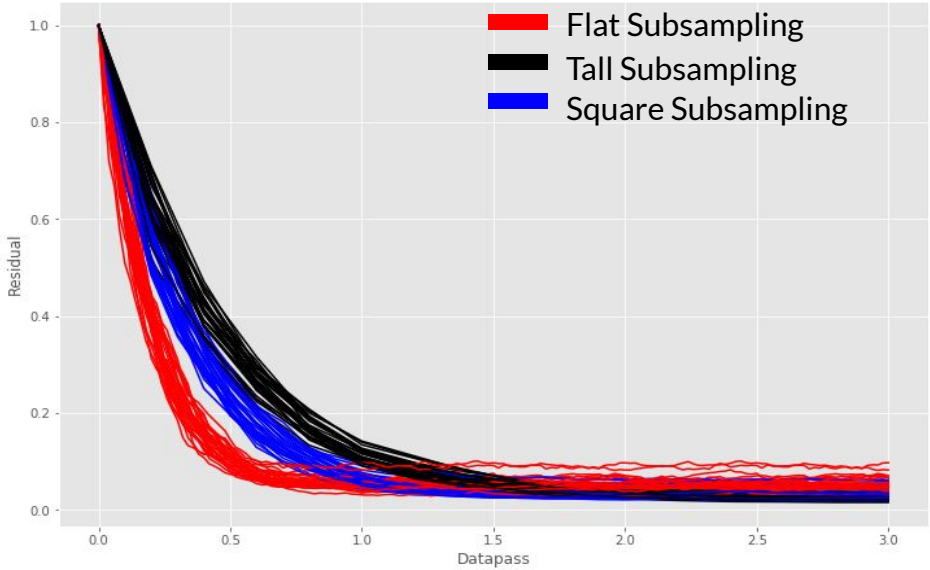
Submatrix size for subsampling

Linearized Bregman



Flat subsampling mean residual: 0.0194
Tall subsampling mean residual: 0.0255
Square subsampling mean residual: 0.0218

ISTA



Flat subsampling mean residual: 0.05381
Tall subsampling mean residual: 0.02483
Square subsampling mean residual: 0.0311

Conclusion & References

Through this dynamical study, we observed that Linearized Bregman functions best with large entries and large λ while ISTA operates best with small entries and small λ. This prompted us to create the Hybrid algorithm which benefits from both quick convergence as well as less chatter. Furthermore, we discovered that LB outperforms ISTA when subsampled. To extend on this research, more studies regarding the parameter choices (eg. noise, initial guess x_0, data setup, specific rows when subsampling) could be conducted .

Emmanoil Daskalakis, Felix J. Herrman, and Rachel Kuske, *Accelerating Sparse Recovery by Reducing Chatter* (2020)

W. Yin, S. Osher, D. Goldfarb, and J. Darbon, Bregman iterative algorithms for l1- minimization with applications to compressed sensing, SIAM Journal on Imaging Sciences, 1 (2008), pp. 143–168, https://doi.org/10.1137/070703983, http://dx.doi.org/10. 1137/070703983, https://arxiv.org/abs/http://dx.doi.org/10.1137/070703983.

D. A. Lorenz, F. Schöpfer, and S. Wenger, The linearized Bregman method via split feasibility problems: Analysis and generalizations, SIAM Journal on Imaging Sciences, 7 (2014), pp. 1237–1262, https://doi.org/10.1137/130936269, http://dx.doi.org/10.1137/ 130936269, https://arxiv.org/abs/http://dx.doi.org/10.1137/130936269.

D. A. Lorenz, S. Wenger, F. Schöpfer, and M. Magnor, A sparse Kaczmarz solver and a linearized Bregman method for online compressed sensing, in 2014 IEEE International Conference on Image Processing (ICIP), Oct 2014, pp. 1347–1351, https://doi.org/10.1109/ ICIP.2014.7025269.