Stochastic Natural Thresholding Algorithms

Rachel Grotheer*, Shuang Li†, Anna Ma‡, Deanna Needel†, and Jing Qin§
* Dept. of Mathematics, Wofford College, Spartanburg, USA
† Dept. of Mathematics, University of California, Los Angeles, USA
‡ Dept. of Mathematics, University of California, Irvine, USA
§ Dept. of Mathematics, University of Kentucky, Lexington, USA

2023 57th Asilomar Conference on Signals, Systems, and Computers | 979-8-3503-2574-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/IEEECONF59524.2023.10477079

***Abstract*—Sparse signal recovery is one of the most fundamental problems in various applications, including medical imaging and remote sensing. Many greedy algorithms based on the family of hard thresholding operators have been developed to solve the sparse signal recovery problem. More recently, Natural Thresholding (NT) has been proposed with improved computational efficiency. This paper proposes and discusses convergence guarantees for stochastic natural thresholding algorithms by extending the NT from the deterministic version with linear measurements to the stochastic version with a general objective function. We also conduct various numerical experiments on linear and nonlinear measurements to demonstrate the performance of StoNT.**

***Index Terms*—natural thresholding, stochastic, gradient matching pursuit, sparse signal recovery**

I. Introduction

In various fields, such as machine learning, computer vision, and signal processing, there is a widespread need to make inferences about data with a high number of dimensions, even when only limited measurements are available. Effective algorithms for data inference from a limited number of measurements often rely on the observation that even though most real-world data exists in high-dimensional spaces, they often possess a low-dimensional complexity, such as sparsity. Many signal recovery algorithms have been developed to exploit sparsity with promising effectiveness and efficiency for data inference and recovery.

In the sparse signal recovery, the underlying data $\mathbf{x} \in \mathbb{R}^n$ is typically recovered by solving an optimization problem of the form

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad ||\mathbf{x}||_0 \leq k, \tag{1}$$

where the objective function, $f(\mathbf{x})$, measures the model discrepancy and k is a preassigned sparsity level of $\mathbf{x}$. For example, compressed sensing assumes the measurements are linearly related to the underlying sparse signal up to noise, which has the objective function

$$f(\mathbf{x}) = ||\mathbf{A}\mathbf{x} - \mathbf{y}||_2^2 \tag{2}$$

DN was partially supported by NSF DMS 2011140 and NSF DMS 2108479. JQ was supported by NSF DMS 1941197.

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the sensing matrix, and $\mathbf{y} := \mathbf{A}\mathbf{x} + \nu \in \mathbb{R}^m$ is the vector of measurements, with Gaussian noise ν.

The optimization problem in (1) can be solved using greedy iterative methods that employ thresholding operators. Thresholding algorithms are particularly effective at solving these optimization problems due to their low computational complexity. To enforce the sparsity, a thresholding operator is usually involved to either restrict the support of the estimated solution at each iteration with a fixed cardinality or approximate the support of the actual solution through iterations. For example, Iterative Hard Thresholding (IHT) [2] and its variants [1], [5], [6], and Gradient Matching Pursuit (GradMP) [8] which are based on the hard thesholding operator have shown the promising performance in many applications. Several other types of thresholding operators exist, such as soft thresholding [3], [4] and optimal k-thresholding (OT) [9]. More recently, natural thresholding [10] has been proposed to significantly reduce the computational cost of OT.

Specifically, to solve (1), application of gradient descent and thresholding operator yields the IHT with the following iterative algorithm

$$\mathbf{x}^{(i+1)} = \mathcal{H}_k(\mathbf{x}^{(i)} - \lambda \nabla f(\mathbf{x}^{(i)}))$$

where $\mathcal{H}_k$ is a hard thresholding that sets all but the largest k components of a vector to zero, $\nabla f(\mathbf{x}^{(i)})$ is the gradient of f at $\mathbf{x}^{(i)}$, and $\lambda > 0$ is the step size. In the linear case (2), $\nabla f(\mathbf{x}) = \mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{y})$. However, the IHT type of algorithms easily cause numerical instability when the hard thresholding is independent of the objective function, especially in the linear case [9]. To address this issue, OT selects the k components of a vector that achieves the least residual among all possible k-sparse selections.

To further enhance the performance of OT, the Natural Thresholding algorithm (NT) restricts the gradient of the regularized objective function of the OT given in [9] to its k-smallest elements. The regularized objective function is given by

$$g_\alpha(\mathbf{w}) = ||\mathbf{y} - \mathbf{A}(\mathbf{u} \otimes \mathbf{w})||_2^2 + \alpha\phi(\mathbf{w}), \tag{3}$$

where $\mathbf{u} \in \mathbb{R}^n$ is a given vector, $\otimes$ is the Hadamard multiplication, $\mathbf{w}$ is a binary vector, α is the regularization parameter and $\phi(\mathbf{w})$ is the regularization function that enforces the binary condition on $\mathbf{w}$. The Natural Thresholding Pursuit algorithm (NTP) is an extension of the NT algorithm that includes an orthogonal projection in the last step. Both algorithms are given in Algorithm 1, where a general objective function is used while only a linear objective function was presented in [10].

Algorithm 1 Natural Thresholding (NT) and Natural Thresholding Pursuit (NTP)

Inputs: $\mathbf{x}^{(0)}$, sparsity level k, stepsize λ, tolerance ε, regularization parameter $\alpha > 0$, maximum number of iterations T

for $i = 1, 2, \ldots, T$ **do**

$$\mathbf{u}^{(i)} = \mathbf{x}^{(i)} - \lambda \nabla f(\mathbf{x}^{(i)})$$

$$\mathbf{w}^- = \underset{\mathbf{w}\in\{0,1\}^n}{\operatorname{argmin}} \|\mathbf{w} - \mathbf{u}^{(i)}\|_2$$

$$\nabla g_\alpha(\mathbf{w}^-) = \nabla f(\mathbf{w}^- \otimes \mathbf{u}^{(i)}) + \alpha \nabla \phi(\mathbf{w}^-)$$

$$\mathbf{w}^+ = \underset{\mathbf{w}\in\{0,1\}^n,\, \mathbf{e}^T\mathbf{w}=k}{\operatorname{argmin}} \nabla g_\alpha(\mathbf{w}^-)^T \mathbf{w}$$

$$S^{(i)} = \operatorname{supp}(\mathbf{w}^+ \otimes \mathbf{u}^{(i)})$$

$$\mathbf{x}^{i+1} = \begin{cases} \mathbf{w}^+ \otimes \mathbf{u}^{(i)} & (NT) \\ \underset{\operatorname{supp}(\mathbf{z})\subseteq S^{(i)}}{\operatorname{argmin}} f(\mathbf{z}) & (NTP) \end{cases}$$

end for

When the data size is growing, stochastic versions of these thresholding algorithms, such as stochastic GradMP (StoGradMP) and stochastic IHT (StoIHT) [7], have the benefit of reduced computational complexity and running time. Here the objective function $f(\mathbf{x})$ is assumed to be separable, that is, $f(\mathbf{x}) = \sum_i f_i(\mathbf{x})$. At each iteration, a small subset of indices n_i are randomly chosen, and the gradient is computed only for the f_i where $i \in n_i$.

In this paper, we propose two new algorithms– stochastic Natural Thresholding (StoNT) and Stochastic Natural Thresholding Pursuit (StoNTP). These algorithms are the respective stochastic version of NT and NTP proposed by [10]. The convergence of our algorithm is discussed when solving (1) with the objective function given by (2). A variety of numerical simulations have shown that StoNTP converges faster than the NTP algorithm with proper parameters.

II. Stochastic Iterative Natural Thresholding

Before introducing our algorithms, we provide the necessary assumptions for the objective function. First, we require that f satisfy the restricted strong convexity (RSC) condition and that each of the f_i satisfy the restricted strongly smooth condition.

Definition 1 (RSS). *A function $f : \mathbb{R}^n \to \mathbb{R}$ is called restricted strongly smooth (RSS) with a constant $\rho_k^+ > 0$ if the following condition is satisfied*

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}')\|_2 \le \rho_k^+ \|\mathbf{x} - \mathbf{x}'\|_2$$

for any $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$ with $|\operatorname{supp}(\mathbf{x}') \cup \operatorname{supp}(\mathbf{x})| \le k$.

Definition 2 (RSC). *A function $f : \mathbb{R}^n \to \mathbb{R}$ is called restricted strongly convexity (RSC) with a constant $\rho_k^- > 0$ if the following condition is satisfied:*

$$f(\mathbf{x}') - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{x}' - \mathbf{x} \rangle \ge \frac{\rho_k^-}{2} \|\mathbf{x}' - \mathbf{x}\|_2^2$$

for any $\mathbf{x}', \mathbf{x} \in \mathbb{R}^n$ with $|\operatorname{supp}(\mathbf{x}') \cup \operatorname{supp}(\mathbf{x})| \le k$.

A. Proposed Algorithms

Given a function $f : \mathbb{R}^n \to \mathbb{R}$ which is differentiable and separable, i.e.,

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}), \quad n \in \mathbb{N},$$

we consider the sparsity-constrained minimization problem

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad \|\mathbf{x}\|_0 \le k \tag{4}$$

where $k \in \{1, 2, \ldots, n\}$. By letting $\mathbf{x} = \mathbf{u} \otimes \mathbf{w}$, the sparsity constraint can be recast as

$$\mathbf{e}^T \mathbf{w} = k, \quad \mathbf{w} \in \{0, 1\}^n,$$

where $\mathbf{e} = [1, 1, \ldots, 1]^T \in \mathbb{R}^n$.

We propose two new algorithms, the Stochastic Natural Thresholding (StoNT) algorithm, and the Stochastic Natural Thresholding Pursuit (StoNTP) algorithm, described in Algorithm 2. For generality, we select an index or batch of indices n_i with probability $p(n_i)$ instead of prescribing a specific probability distribution for index/batch choice. In applications where no prior information is known, this distribution is typically taken to be the uniform distribution.

III. Theoretical Guarantees

In this section, we will focus on the linear measurement case for convergence analysis, which can be further extended to the nonlinear case. Consider $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2$ where $\mathbf{y} = \mathbf{A}\mathbf{x}^* + \nu$ and $\|\mathbf{x}^*\|_0 \le k$ where $\mathbf{A} \in \mathbb{R}^{m\times n}$ ($m \ll n$) satisfies the RIP Condition for k-sparse vectors with RIP constant δ_k.

Theorem 1. *(Linear Convergence of StoIHT [7, Theorem 1]) Let $\mathbf{x}_s$ be a feasible solution of*

$$\min_{\mathbf{x}} \frac{1}{m} \sum_{i=1}^{m} f_i(\mathbf{x}) \quad \textit{s.t.} \quad \|\mathbf{x}\|_0 \le k.$$

Algorithm 2 Stochastic Natural Thresholding (StoNT) and Stochastic Natural Thresholding Pursuit (StoNTP)

Inputs: $\mathbf{x}^{(0)}$, sparsity level k, stepsize λ, probability $p(n_i)$, tolerance ε, regularization parameter $\alpha > 0$, maximum number of iterations T

for $i = 1, 2, \ldots, T$ **do**

Randomly select an index or a batch of indices n_i with a probability $p(n_i)$

$$\mathbf{u}^{(i)} = \mathbf{x}^{(i)} - \frac{\lambda}{np(n_i)} \nabla f_{n_i}(\mathbf{x}^{(i)})$$

$$\mathbf{w}^- = \underset{\mathbf{w}\in\{0,1\}^n}{\operatorname{argmin}} \|\mathbf{w} - \mathbf{u}^{(i)}\|_2$$

$$\nabla g_\alpha(\mathbf{w}^-) = \nabla f(\mathbf{w}^- \otimes \mathbf{u}^{(i)}) + \alpha \nabla \phi(\mathbf{w}^-)$$

$$\mathbf{w}^+ = \underset{\mathbf{w}\in\{0,1\}^n,\, \mathbf{e}^T\mathbf{w}=k}{\operatorname{argmin}} \nabla g_\alpha(\mathbf{w}^-)^T \mathbf{w}$$

$$S^{(i)} = \operatorname{supp}(\mathbf{w}^+ \otimes \mathbf{u}^{(i)})$$

$$\mathbf{x}^{i+1} = \begin{cases} \mathbf{w}^+ \otimes \mathbf{u}^{(i)} & (StoNT) \\ \underset{\operatorname{supp}(\mathbf{z})\subseteq S^{(i)}}{\operatorname{argmin}} f(\mathbf{z}) & (StoNTP) \end{cases}$$

end for

Suppose that $i \sim [m]$ with probability $p(i)$ and let

$$\mathbf{x}^{t+1} = \mathcal{H}_k\left(\mathbf{x}^t - \frac{\lambda}{mp(i)} \nabla f_i(\mathbf{x}_t)\right).$$

If $\lambda < 2/\alpha_{3k}$ then:

$$\mathbb{E}\|\mathbf{x}^{t+1} - \mathbf{x}_S\|_2 \le \kappa \|\mathbf{x}^t - \mathbf{x}_S\|_2 + \sigma_{\mathbf{x}_S}, \qquad (5)$$

where κ and $\sigma_{\mathbf{x}_S}$ are constants that depend on the RSS and RSC constant and $\alpha_k = \max_i \frac{\rho_k^+(i)}{mp(i)}$.

Theorem 2. *Assume the rows of $\mathbf{A}$ have unit norms. Consider Algorithm 2 with batch size bs=1, and choose $\lambda < 2/\alpha_{3k}$ where $\alpha_{3k} = \max_i \frac{\rho_{3k}^+(i)}{mp(i)}$. Then*

$$\mathbb{E}\|\mathbf{x}_S - \mathbf{x}^{(p+1)}\|_2 \le \kappa_{new} \|\mathbf{x}_S - \mathbf{x}^{(p)}\|^2 + \sigma_{new},$$

where $\kappa_{new} = \sqrt{\frac{1+\delta_{2k}}{1-\delta_{2k}}}\kappa$ and $\sigma = \frac{\sqrt{1+\delta_{2k}}\sigma_{\mathbf{x}_S} + 2\|\nu'\|_2}{\sqrt{1-\delta_{2k}}}$.

Proof. Starting with Eq. (34) in [10], we have:

$$\begin{aligned} &\mathbb{E}\|\mathbf{x}_S - \mathbf{x}^{(p+1)}\|_2 \\ &\le \sqrt{\frac{1+\delta_{2k}}{1-\delta_{2k}}} \mathbb{E}\|\mathbf{x}_S - \mathcal{H}_k(u^{(p)})\|_2 + \frac{2\|\nu'\|_2}{\sqrt{1-\delta_{2k}}} \\ &\le \sqrt{\frac{1+\delta_{2k}}{1-\delta_{2k}}} \kappa \|\mathbf{x}_S - \mathbf{x}^{(p)}\|_2 + \frac{\sqrt{1+\delta_{2k}}\sigma_{\mathbf{x}_S} + 2\|\nu'\|_2}{\sqrt{1-\delta_{2k}}}. \end{aligned}$$

where in the first inequality, we are taking an expectation conditional on the first p iterations of Algorithm 2, and in the second inequality, we use Theorem 1. Iterating the expectation obtains the desired result. □

IV. Numerical Experiments

Various experiments on linear and nonlinear measurements are conducted to evaluate the proposed performance between NTP and StoNTP. We refer the reader to [10] for further comparisons of NTP with alternative approaches such as CoSAMP, HTP, and OMP . We adopt the following two comparison metrics: (1) relative error $\|\mathbf{x} - \mathbf{x}^*\|_2/\|\mathbf{x}^*\|_2$ where $\mathbf{x}$ is an approximation of the ground truth vector $\mathbf{x}^*$; (2) success rate which is a percentage of successful cases with correctly identified support out of the total trials. Numerical experiments were run on a 2015 Macbook Pro in MATLAB R2017b with 8 GB RAM and a 2.7 GHz Dual-Core Intel Core i5.

A. Linear Measurements

First, we illustrate the performance of StoNTP on the least squares problem, where the objective function is given as in (2). We generate $\mathbf{x}^\star \in \mathbb{R}^{800}$ as a normalized sparse Gaussian random vector with 10 uniformly distributed nonzero entries. The sensing matrix $\mathbf{A} \in \mathbb{R}^{100\times 800}$ is generated as a Gaussian random matrix with normalized columns. We then get the random measurements as $\mathbf{y} = \mathbf{A}\mathbf{x}^\star$. We set the maximal number of iterations as 150 and the batch size as 10. The algorithm stops either when it achieves the maximal number of iterations or the loss function $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \le 10^{-3}$. To see the best choice of the regularization parameter α, we first fix the step size $\lambda = 2$. The value of the loss function and distance between the estimated $\mathbf{x}$ and $\mathbf{x}^\star$ evaluated at each iteration and versus the running time are illustrated in Fig. 1. It can be seen that the best choice is $\alpha = 1$. Next, we repeat the experiment by fixing $\alpha = 1$ and test on a variety of λ values. As is shown in Fig. 2, the best step size is $\lambda = 2$. We also compare our StoNTP algorithm with the NTP algorithm. It can be seen from Fig. 3 that the StoNTP algorithm significantly outperforms the NTP algorithm. In addition, we test the success rates for NTP and StoNTP for various parameters in Fig. 4.

B. Nonlinear Measurements

We extend the measurements from the linear case to the nonlinear case, and consider the L_2-regularized logistic regression model and support vector machine (SVM). In what follows, we set $a = 5$, $\epsilon = 10^{-3}$, and batch size to be 20. We also select each component function uniformly.

First, we consider the logistic regression model with the following objective function $f(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^m \log(1 + \exp(-2y_i(\mathbf{a}_i\mathbf{x})))$, where $\mathbf{a}_i$ represents the i-th row from the measurement matrix $A \in \mathbb{R}^{100\times 800}$ and classifiers $y_i \in \{-1, 1\}$ such that $y_i = 1$

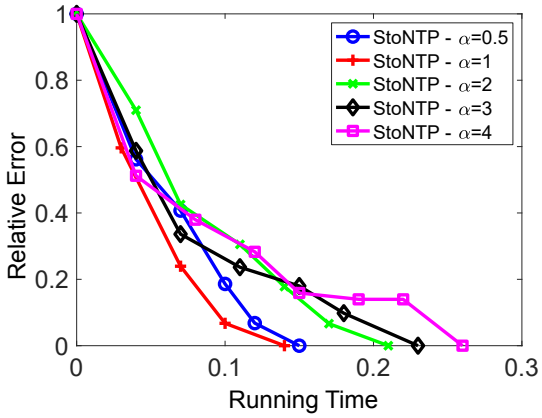


Fig. 1. Test StoNTP with various α's: $m = 100$, $n = 800$, $k = 10$, $\lambda = 2$. Batch size for StoNTP is 10. The best choice is $\alpha = 1$.

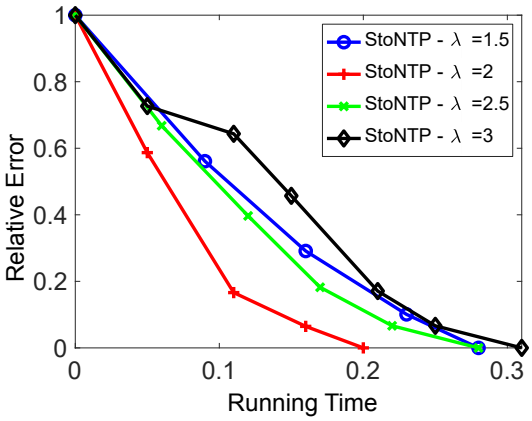


Fig. 2. Test StoNTP with different λ's: $m = 100$, $n = 800$, $k = 10$, $\alpha = 1$. Batch size for StoNTP is 10. The best step size is $\lambda = 2$.

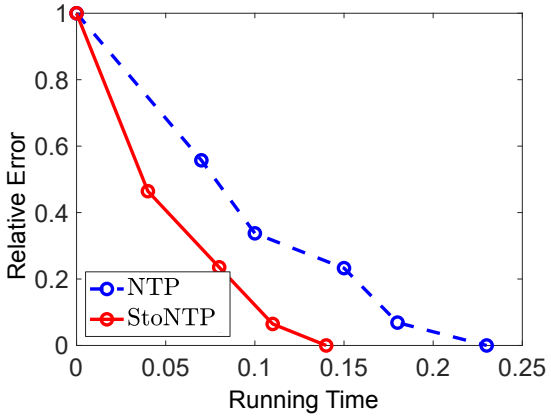
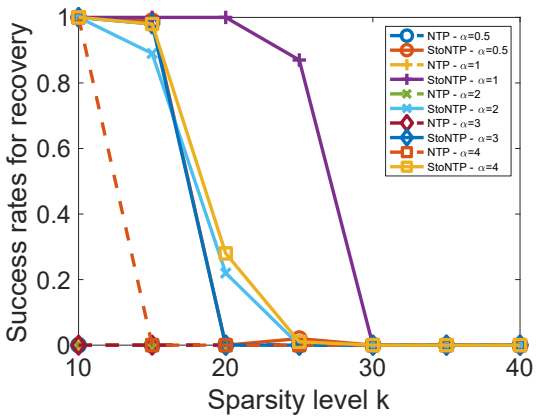
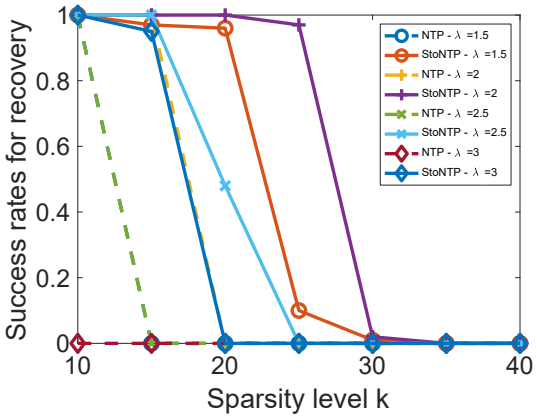


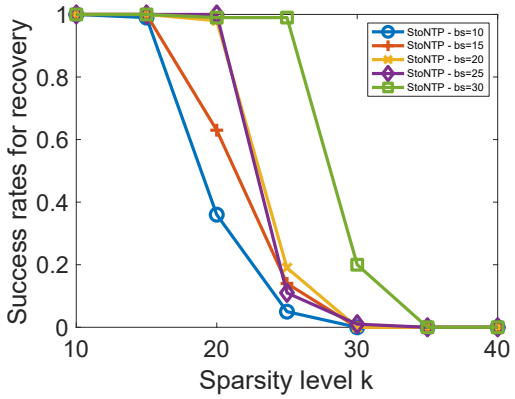
Fig. 3. NTP vs StoNTP: $m = 100$, $n = 800$, $k = 10$. For NTP, we choose $\lambda = 2$, $\alpha = 5$. For StoNTP, we choose $\lambda = 2$, $\alpha = 1$. The batch size for StoNTP is 20.



StoNTP: $bs = 30$, $\lambda = 2$



NTP: $bs = 30$, $\lambda = 2$, StoNTP $\alpha = 1$



StoNTP: $\alpha = 1$, $\lambda = 2$.

Fig. 4. Success rates for NTP and StoNTP with different parameters: $m = 100$, $n = 800$, $\alpha \in \{0.5, 1, 2, 3, 4\}$, $\lambda \in \{1.5, 2, 2.5, 3\}$, batch size $bs \in \{10, 15, 20, 25, 30\}$.

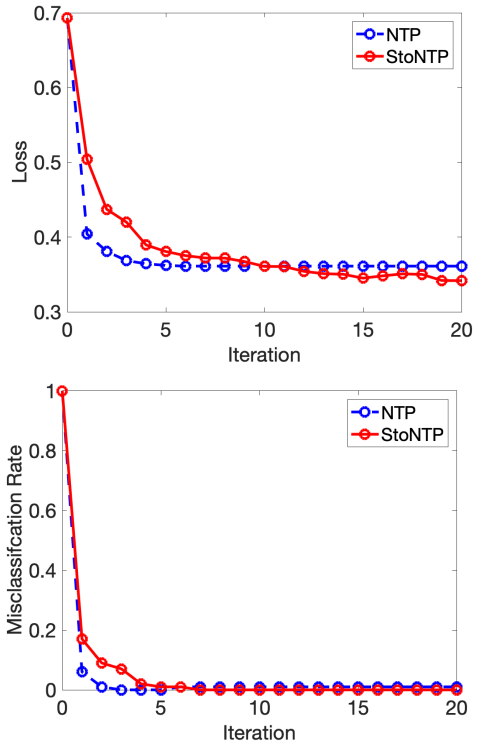


Fig. 5. Comparison of NTP and StoNTP for logistic regression: convergence of loss (top) and misclassification rate (bottom).

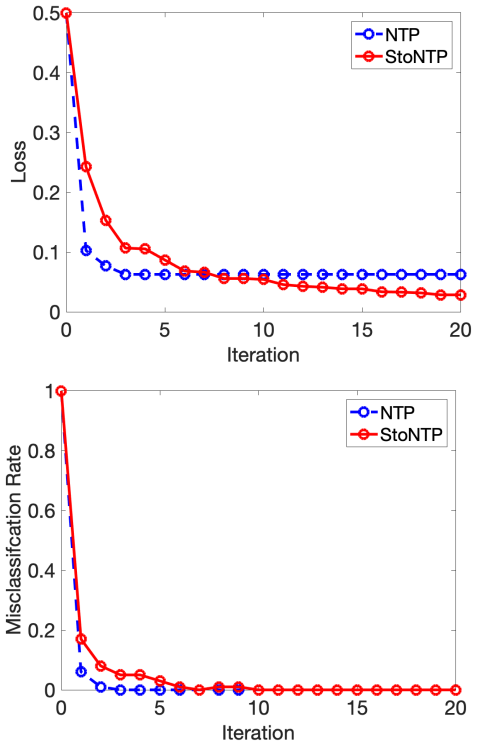


Fig. 6. Comparison of NTP and StoNTP for SVM: convergence of loss (top) and misclassification rate (bottom).

with probability $p = \exp(\mathbf{a}_i\mathbf{x}^*)/(1+\exp(\mathbf{a}_i\mathbf{x}^*))$ for a fixed $\mathbf{x}^*$ (the solution). The performance of our algorithms is shown in Fig. 5. In this experiment, the vectors a_i are drawn i.i.d. from a Gaussian distribution and normalized to have unit norm. We set $m = 100$, $n = 800$, and $k = 40$. For NTP the step size is $\lambda = 10$ and the step size for StoNTP is $\lambda = 30$. As shown in Fig. 6, both StoNTP and NTP can attain a zero misclassification error. Notably, StoNTP can obtain a smaller loss than NTP.

Next, we consider the SVM problem with $f(\mathbf{x}) = \frac{1}{2m}\sum_{i=1}^{m}(\max\{0, 1 - y_i\mathbf{a}_i\mathbf{x}\})^2$ where $\mathbf{a}_i$'s, y_i's are defined as before. We set $m = 100$, $n = 800$, and $k = 40$, and obtained the results in Fig. 6. For NTP the step size is $\lambda = 10$, and the step size for StoNTP is $\lambda = 20$. As shown in Fig. 6, both StoNTP and NTP can attain a zero misclassification error. Notably, StoNTP can obtain a smaller loss than NTP. The vectors a_i are drawn i.i.d. from a Gaussian distribution and normalized to have a unit norm.

V. Conclusion

In this paper, we propose two stochastic natural thresholding algorithms, i.e., StoNT and StoNTP, by extending the natural thresholding from the linear case to a general one and from the deterministic version to the stochastic one. Numerical simulations on linear and nonlinear measurements have shown the great potential of our algorithms in improving the recovery accuracy and computational efficiency.

References

[1] T Blumensath and M Davies. Iterative hard thresholding for sparse approximations: The journal of fourier analysis and applications, 14, no. 5-6, 629–654, 2008.

[2] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[3] Kristian Bredies and Dirk A Lorenz. Linear convergence of iterative soft-thresholding. *Journal of Fourier Analysis and Applications*, 14(5):813–837, 2008.

[4] David L Donoho. De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627, 1995.

[5] Simon Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM Journal on numerical analysis*, 49(6):2543–2563, 2011.

[6] Kyle K Herrity, Anna C Gilbert, and Joel A Tropp. Sparse approximation via iterative thresholding. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 3, pages III–III. IEEE, 2006.

[7] Nam Nguyen, Deanna Needell, and Tina Woolf. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *IEEE Transactions on Information Theory*, 63(11):6869–6895, 2017.

[8] NH Nguyen, S Chin, and TD Tran. A unified iterative greedy algorithm for sparsityconstrainted optimization. 2013. Available: https://sites.google.com/site/namnguyenjhu/gradMP.pdf.

[9] Yun-Bin Zhao. Optimal k-thresholding algorithms for sparse optimization problems. *SIAM Journal on Optimization*, 30(1):31–55, 2020.

[10] Yun-Bin Zhao and Zhi-Quan Luo. Natural thresholding algorithms for signal recovery with sparsity. *IEEE Open Journal of Signal Processing*, 3:417–431, 2022.