Trainable ISTA for Sparse Signal Recovery

Daisuke Ito, Satoshi Takabe, *Member, IEEE*, and Tadashi Wadayama, *Member, IEEE*

Abstract—**In this paper, we propose a novel sparse signal recovery algorithm called the trainable iterative soft thresholding algorithm (TISTA). The proposed algorithm consists of two estimation units: a linear estimation unit and a minimum mean squared error (MMSE) estimator based shrinkage unit. The error variance required in the MMSE shrinkage unit is precisely estimated from a tentative estimate of the original signal. The remarkable feature of the proposed scheme is that TISTA includes adjustable variables that control step size and the error variance for the MMSE shrinkage function. The variables are adjusted by standard deep learning techniques. The number of trainable variables of TISTA is nearly equal to the number of iteration rounds and is much smaller than that of known learnable sparse signal recovery algorithms. This feature leads to highly stable and fast training processes of TISTA. Computer experiments show that TISTA is applicable to various classes of sensing matrices, such as Gaussian matrices, binary matrices, and matrices with large condition numbers. Numerical results also demonstrate that, in many cases, TISTA provides significantly faster convergence than approximate message passing (AMP) and the learned iterative shrinkage thresholding algorithm and also outperforms orthogonal AMP in the NMSE performance.**

Index Terms—**Compressed sensing, machine learning, supervised learning.**

I. Introduction

THE basic problem setup for *compressed sensing* [1], [2] is as follows. A real vector $\boldsymbol{x} \in \mathbb{R}^N$ represents a sparse source signal. It is assumed that we cannot directly observe $\boldsymbol{x}$, but we observe $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$, where $\boldsymbol{A} \in \mathbb{R}^{M \times N} (N > M)$ is a sensing matrix and $\boldsymbol{w} \in \mathbb{R}^M$ is a Gaussian noise vector. The goal is to estimate $\boldsymbol{x}$ from $\boldsymbol{y}$ as correctly as possible.

For a number of sparse reconstruction algorithms [3], the Lasso formulation [4] is fairly common for solving sparse signal recovery problems. In the Lasso formulation, the original problem is recast as a convex optimization problem for minimizing $\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1$. The regularization term $\lambda\|\boldsymbol{x}\|_1$ promotes the sparseness of a reconstruction vector, where λ is the regularization constant. A number of algorithms have been developed in order to solve Lasso problems efficiently [5]–[7]. The Iterative Shrinkage Thresholding Algorithm (ISTA) [8], [9] is one of the best-known algorithms for solving the Lasso problem. ISTA is an iterative algorithm comprising two processes: a linear estimation process and a shrinkage process based on a soft thresholding function. ISTA can be seen as a proximal gradient descent algorithm [10] and can be directly derived from the Lasso formulation.

Approximate Message Passing (AMP) [11], [12], which is a variant of approximate belief propagation, generally exhibits much faster convergence than the ISTA. The remarkable feature of AMP is that its asymptotic behavior is completely described by the state evolution equations [13], [14]. AMP is derived based on the assumption that the sensing matrices consist of i.i.d. Gaussian distributed components. Recently, Ma and Ping proposed Orthogonal AMP (OAMP) [15], which can handle various classes of sensing matrices, including unitary invariant matrices. Rangan *et al.* proposed Vector AMP [16] for right-rotationally invariant matrices and provided a theoretical justification for its state evolution. Independently, Takeuchi [17] also gave a rigorous analysis for a sparse recovery algorithm for unitary invariant measurements based on the expectation propagation framework.

The recent advent of powerful neural networks (NNs) triggered the remarkable spread of research activities and applications on deep neural networks (DNNs) [18], [19]. DNNs have found a number of practical applications such as image recognition [20], [21], speech recognition [22], [23], and robotics because of their outstanding performance compared with traditional methods. The advancement of DNNs has also had an impact on the design of algorithms for communications and signal processing [24]–[26]. By unfolding an iterative process of a sparse signal recovery algorithm, we can obtain a *signal-flow graph*. The signal-flow graph includes trainable variables that can be tuned with a supervised learning method, i.e., standard deep learning techniques such as stochastic gradient descent algorithms [27] based on back propagation [28] and mini-batches can be used to adjust the trainable variables. Gregor and LeCun presented the Learned ISTA (LISTA) [29], which uses learnable threshold variables for a shrinkage function. LISTA provides a recovery performance that is superior to that of the original ISTA. Borgerding *et al.* also presented variants of AMP and VAMP with learnable capability [30], [31].

The goal of the present study is to propose a simple sparse recovery algorithm based on deep learning techniques. The proposed algorithm, called the *Trainable ISTA (TISTA)*, borrows the basic structure of ISTA, and adopts the estimator of the squared error between true signals and tentative estimations, i.e., the *error variance estimator*, from OAMP [15]. Thus, TISTA

Manuscript received April 5, 2018; revised November 15, 2018 and January 22, 2019; accepted April 5, 2019. Date of publication April 23, 2019; date of current version May 6, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gwo Giun Lee. This work was supported by the Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Scientific Research (B) under Grant 16H02878, Grant 19H02138 (TW), and Grant-in-Aid for Young Scientists (start-up) under Grant 17H06758 (ST). This paper was presented in part at the IEEE International Conference on Communications Workshop, Sheraton Kansas City Hotel, Kansas City, MO, USA, May 2018. *(Corresponding author: Satoshi Takabe.)*

D. Ito and T. Wadayama are with the Nagoya Institute of Technology, Nagoya 466-8555, Japan (e-mail: d.ito.480@stn.nitech.co.jp; wadayama@nitech.ac.jp).

S. Takabe is with the Nagoya Institute of Technology, Nagoya 466-8555, Japan, and also with the RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan (e-mail: s_takabe@nitech.ac.jp).

Digital Object Identifier 10.1109/TSP.2019.2912879

consists of the three parts: a linear estimator, a minimum mean squared error (MMSE) estimator-based shrinkage function, and the above-mentioned error variance estimator. The linear estimator of TISTA includes trainable variables that can be adjusted via deep learning techniques. Zhang and Ghanem [32] proposed ISTA-Net, which is also an ISTA-based algorithm with learnable capability. The notable difference between ISTA-Net and TISTA is that TISTA uses an error variance estimator, which significantly improves the speed of convergence.

II. Brief Review of Known Recovery Algorithms

As preparation for describing the details of the proposed algorithm, several known sparse recovery algorithms are briefly reviewed in this section. In the following, the observation vector is assumed to be $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$, where $\boldsymbol{A} \in \mathbb{R}^{M\times N}(N > M)$ and $\boldsymbol{x} \in \mathbb{R}^N$. Each entry of the additive noise vector $\boldsymbol{w} \in \mathbb{R}^M$ follows a zero-mean Gaussian distribution with variance σ^2.

A. ISTA

The ISTA is a well-known sparse recovery algorithm [8] defined by the following simple recursion:

$$\boldsymbol{r}_t = \boldsymbol{s}_t + \beta \boldsymbol{A}^T(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}_t) \quad (1)$$

$$\boldsymbol{s}_{t+1} = \eta(\boldsymbol{r}_t; \tau), \quad (2)$$

where $\beta \in \mathbb{R}$ represents the step size, and $\eta(\cdot;\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ is the soft thresholding function defined by

$$\eta(\boldsymbol{r}; \tau) = (\tilde{\eta}(r_1; \tau), \ldots, \tilde{\eta}(r_n; \tau)),$$

where $\tilde{\eta}(\cdot;\cdot) : \mathbb{R} \to \mathbb{R}$ is given by

$$\tilde{\eta}(r; \tau) = \mathsf{sign}(r) \max\{|r| - \tau, 0\}. \quad (3)$$

The parameter $\tau \in \mathbb{R}(\tau > 0)$ indicates the threshold value. After T-iterations, the estimate $\hat{\boldsymbol{x}} = \boldsymbol{s}_T$ of the original sparse signal $\boldsymbol{x}$ is obtained. The initial value is assumed to be $\boldsymbol{s}_0 = \boldsymbol{0}$. In order to have convergence, the step size β should be carefully determined [8]. Several accelerated methods for ISTA using a momentum term, such as the Fast ISTA (FISTA), have been proposed [33], [34]. Since the proximal operator of the ℓ_1-regularization term $\|\boldsymbol{x}\|_1$ is the soft thresholding function, the ISTA can be seen as a proximal gradient descent algorithm [3].

B. AMP

AMP [12] is defined by the following recursion:

$$\boldsymbol{r}_t = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}_t + b_t \boldsymbol{r}_{t-1}, \quad (4)$$

$$\boldsymbol{s}_{t+1} = \eta(\boldsymbol{s}_t + \boldsymbol{A}^T \boldsymbol{r}_t; \tau_t), \quad (5)$$

$$b_t = \frac{1}{M}\|\boldsymbol{s}_t\|_0, \quad \tau_t = \frac{\theta}{\sqrt{M}}\|\boldsymbol{r}_t\|_2 \quad (6)$$

and provides the final estimate $\hat{\boldsymbol{x}} = \boldsymbol{s}_T$. Each entry of the sensing matrix $\boldsymbol{A}$ is assumed to be generated according to the Gaussian distribution $\mathcal{N}(0, 1/M)$, i.e., a Gaussian distribution with mean zero and variance $1/M$. At a glance, the recursive formula of AMP appears similar to that of ISTA, but there are several critical differences. Due to the *Onsager correction term* $b_t \boldsymbol{r}_{t-1}$ in (4), the output of the linear estimator becomes statistically decoupled, and an error between each output signal from the linear estimator and the true signal behaves as a white Gaussian random variable in the large system limit. This enables us to use a scalar recursion called the *state evolution* to track the evolution of the error variances.

Another difference between ISTA and AMP is the estimator of τ_t in (6), which is used as the threshold value for the shrinkage function (5). In [12], it was reported that AMP exhibits much faster convergence than ISTA if the sensing matrix satisfies the above condition. On the other hand, AMP cannot provide excellent recovery performance for sensing matrices violating the above condition such as non-Gaussian sensing matrices, Gaussian matrices with large variance, Gaussian matrices with nonzero means, and matrices with large condition numbers [35].

C. OAMP

OAMP [15] is defined by the following recursive formula:

$$\boldsymbol{r}_t = \boldsymbol{s}_t + \boldsymbol{W}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}_t), \quad (7)$$

$$\boldsymbol{s}_{t+1} = \eta_{\rm df}(\boldsymbol{r}_t; \tau_t), \quad (8)$$

$$v_t^2 = \max\left\{\frac{\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}_t\|_2^2 - M\sigma^2}{\mathsf{trace}(\boldsymbol{A}^T\boldsymbol{A})}, \epsilon\right\}, \quad (9)$$

$$\tau_t^2 = \frac{1}{N}\mathsf{trace}(\boldsymbol{B}\boldsymbol{B}^T)v_t^2 + \frac{1}{N}\mathsf{trace}(\boldsymbol{W}\boldsymbol{W}^T)\sigma^2, \quad (10)$$

for $t = 0, 1, 2, \ldots, T-1$. The matrix $\boldsymbol{B}$ is given by $\boldsymbol{B} = \boldsymbol{I} - \boldsymbol{W}\boldsymbol{A}$. To be precise, the estimator equations on v_t^2 (9) and τ_t^2 (10) (also presented in [36]) are not part of OAMP (for example, we can use the state evolution to provide v_t^2 and τ_t^2), but these estimators are used for numerical evaluation in [15]. The matrix $\boldsymbol{W}$ in linear estimator (7) is given by $\boldsymbol{W} = N\hat{\boldsymbol{W}}/\mathsf{trace}(\hat{\boldsymbol{W}}\boldsymbol{A})$ where $\hat{\boldsymbol{W}}$ can be chosen from the transpose of $\boldsymbol{A}$, the pseudo inverse of $\boldsymbol{A}$, and the LMMSE matrix. The nonlinear estimation unit (8) consists of a *divergence-free function*$\eta_{\rm df}$ that replaces the Onsager correction term. It is proved in [15] that the estimation errors of linear estimator (7) and non-linear estimator (8) are statistically orthogonal if a sensing matrix is i.i.d. Gaussian or unitary invariant. This provides a justification for the state evolution of OAMP.

III. Details of TISTA

This section describes the details of TISTA and its training process.

A. MMSE Estimator for an Additive Gaussian Noise Channel

Let X be a real-valued random variable with probability density function (PDF) $P_X(\cdot)$. We assume an additive Gaussian noise channel defined by $Y = X + N$, where Y represents a real-valued random variable as well. The random variable N is a Gaussian random variable with mean 0 and variance σ^2. Consider the situation in which a receiver can observe Y and we wish to estimate the value of X.

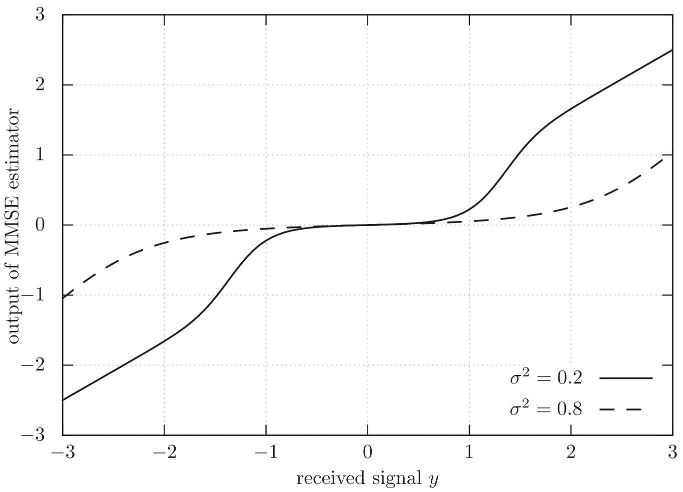

Fig. 1. Plots of η_{MMSE} as a function of a received signal y ($\alpha^2 = 1$, $\sigma^2 = 0.2, 0.8$, $p = 0.1$).

The MMSE estimator $\eta_{MMSE}(y)$ is defined by

$$\eta_{MMSE}(y) = \mathbb{E}[X|y], \tag{11}$$

where $\mathbb{E}[X|y]$ is the conditional expectation given by

$$\mathbb{E}[X|y] = \int_{-\infty}^{\infty} xP(x|y)dx. \tag{12}$$

The posterior PDF $P(x|y)$ is given by Bayes' Theorem:

$$P_{X|Y}(x|y) = \frac{P_X(x)P_{Y|X}(y|x)}{P_Y(y)}, \tag{13}$$

where the conditional PDF is Gaussian:

$$P_{Y|X}(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y-x)^2}{2\sigma^2}\right). \tag{14}$$

In the case of the Bernoulli-Gaussian prior, $P_X(x)$ is given by

$$P_X(x) = (1-p)\delta(x) + \frac{p}{\sqrt{2\pi\alpha^2}} \exp\left(-\frac{x^2}{2\alpha^2}\right), \tag{15}$$

where p represents the probability such that a nonzero element occurs. The function $\delta(\cdot)$ is Dirac's delta function. In this case, a nonzero element follows the Gaussian PDF with mean 0 and variance α^2. The MMSE estimator for the Bernoulli-Gaussian prior can be easily derived [37]–[39] using Stein's formula:

$$\eta_{MMSE}(y;\sigma^2) = y + \sigma^2 \frac{d}{dy} \ln P_Y(y) \tag{16}$$

and we have

$$\eta_{MMSE}(y;\sigma^2) = \left(\frac{y\alpha^2}{\xi}\right) \frac{pF(y;\xi)}{(1-p)F(y;\sigma^2) + pF(y;\xi)}, \tag{17}$$

where $\xi = \alpha^2 + \sigma^2$ and

$$F(z;v) = \frac{1}{\sqrt{2\pi v}} \exp\left(\frac{-z^2}{2v}\right). \tag{18}$$

For example, Fig. 1 shows the shapes of $\eta_{MMSE}(y;\sigma^2)$ as a function of a received signal y for $\sigma^2 = 0.2, 0.8$. The shapes can be observed to resemble those of the soft thresholding function but the function is differentiable everywhere with respect to y.

Let us consider another setting. If each sparse component takes a value in a finite discrete set $S = \{s_1, \ldots, s_M\} (s_i \in \mathbb{R})$ uniformly at random, then the corresponding prior becomes

$$P_X(x) = (1-p)\delta(x) + p\sum_{s\in S} \frac{1}{M}\delta(x-s), \tag{19}$$

and we have the MMSE estimator

$$\eta_{MMSE}(y;\sigma^2) = \frac{p\sum_s sF(s;\sigma^2)}{(1-p)MF(0;\sigma^2) + p\sum_s F(s;\sigma^2)}. \tag{20}$$

These MMSE estimators are going to be used as a building block of the TISTA to be presented in the next subsection.

B. Recursive Formula for TISTA

We assume that the sensing matrix $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ is a full-rank matrix. The recursive formula of TISTA is summarized as follows:

$$\boldsymbol{r}_t = \boldsymbol{s}_t + \gamma_t \boldsymbol{W}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}_t), \tag{21}$$

$$\boldsymbol{s}_{t+1} = \eta_{MMSE}(\boldsymbol{r}_t;\tau_t^2), \tag{22}$$

$$v_t^2 = \max\left\{\frac{\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}_t\|_2^2 - M\sigma^2}{\mathsf{trace}(\boldsymbol{A}^T\boldsymbol{A})}, \epsilon\right\}, \tag{23}$$

$$\tau_t^2 = \frac{v_t^2}{N}(N + (\gamma_t^2 - 2\gamma_t)M) + \frac{\gamma_t^2\sigma^2}{N}\mathsf{trace}(\boldsymbol{W}\boldsymbol{W}^T), \tag{24}$$

where the matrix $\boldsymbol{W} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}$ is the pseudo inverse matrix of the sensing matrix $\boldsymbol{A}$. The initial condition is $\boldsymbol{s}_0 = 0$, and the final estimate is given by $\hat{\boldsymbol{x}} = \boldsymbol{s}_T$. The scalar variables $\gamma_t \in \mathbb{R}$ $(t = 0, 1, \ldots, T-1)$ are learnable variables that are tuned in a training process. The number of learnable variables is thus T, which is much smaller than those of LISTA [29] and LAMP [30]. In addition to the step size parameters $\{\gamma_t\}_{t=0}^{T-1}$, one can also optimize parameters p and α^2 in the MMSE estimator (17) especially for nonsynthetic signals or real data. We assume that they are constant among iterations in TISTA for simplicity. The number of the trainable parameters in this case is thus $T+2$.

An appropriate MMSE shrinkage (22) is chosen according to the prior distribution of the original signal $\boldsymbol{x}$. Note that the MMSE shrinkage is also used in [30]. The real constant ϵ is a sufficiently small value, e.g., $\epsilon = 10^{-9}$. The max operator in (23) is used to prevent the estimate of the variance from being nonpositive. The learnable variables γ_t in (21) provide appropriate step sizes and control for the variance of the MMSE shrinkage.

The true error variances $\bar{\tau}_t^2$ and $\bar{v}_t^2$ are defined by

$$\bar{\tau}_t^2 = \frac{\mathbb{E}[\|\boldsymbol{r}_t - \boldsymbol{x}\|_2^2]}{N}, \quad \bar{v}_t^2 = \frac{\mathbb{E}[\|\boldsymbol{s}_t - \boldsymbol{x}\|_2^2]}{N}. \tag{25}$$

These error variances should be estimated as correctly as possible in a sparse recovery process because the MMSE shrinkage unit (22) requires knowing $\bar{\tau}_t^2$. As in the case of OAMP [15], we make the following assumptions on the residual errors in order to derive an error variance estimator.

The first assumption is that $\boldsymbol{r}_t - \boldsymbol{x}$ consists of i.i.d. zero-mean Gaussian entries. Based on this assumption, each entry of the output from the linear estimator (21) can be seen as an observation obtained from a virtual additive Gaussian noise channel with the noise variance $\bar{\tau}^2$. This justifies the use of the shrinkage function based on the MMSE estimator (22) with $\bar{\tau}^2$. Another assumption is that $\boldsymbol{s}_t - \boldsymbol{x}$ consists of zero-mean i.i.d. entries and satisfies $\mathbb{E}[(\boldsymbol{s}_t - \boldsymbol{x})^T \boldsymbol{A}^T \boldsymbol{w}] = \mathbb{E}[(\boldsymbol{s}_t - \boldsymbol{x})^T \boldsymbol{W}\boldsymbol{w}] = 0$ for any t.

The error variance estimator for $\bar{v}_t^2$ (23) is the same as that of OAMP [15], and its justification comes from the following proposition.

Proposition 1: If each entry of $\boldsymbol{s}_t - \boldsymbol{x}$ is i.i.d. with mean zero and $\mathbb{E}[(\boldsymbol{s}_t - \boldsymbol{x})^T \boldsymbol{A}^T \boldsymbol{w}] = 0$ is satisfied, then

$$\bar{v}_t^2 = \frac{\mathbb{E}[\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}_t\|_2^2] - M\sigma^2}{\mathsf{trace}(\boldsymbol{A}^T\boldsymbol{A})} \tag{26}$$

holds.

(Proof) From the right-hand side of (26), we have

$$\begin{aligned}
&\frac{\mathbb{E}[\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}_t\|_2^2] - M\sigma^2}{\mathsf{trace}(\boldsymbol{A}^T\boldsymbol{A})}\\
&= \frac{\mathbb{E}[\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{w} - \boldsymbol{A}\boldsymbol{s}_t\|_2^2] - M\sigma^2}{\mathsf{trace}(\boldsymbol{A}^T\boldsymbol{A})}\\
&= \frac{\mathbb{E}[\|\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{s}_t) + \boldsymbol{w}\|_2^2] - \mathbb{E}[\boldsymbol{w}^T\boldsymbol{w}]}{\mathsf{trace}(\boldsymbol{A}^T\boldsymbol{A})}\\
&= \frac{\mathbb{E}[(\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{s}_t))^T\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{s}_t) + (\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{s}_t))^T\boldsymbol{w}]}{\mathsf{trace}(\boldsymbol{A}^T\boldsymbol{A})}\\
&= \frac{\mathbb{E}[(\boldsymbol{x} - \boldsymbol{s}_t)^T\boldsymbol{A}^T\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{s}_t)]}{\mathsf{trace}(\boldsymbol{A}^T\boldsymbol{A})}\\
&= \frac{1}{N}\mathsf{trace}(\boldsymbol{A}^T\boldsymbol{A})\mathbb{E}[\|\boldsymbol{s}_t - \boldsymbol{x}\|_2^2]\frac{1}{\mathsf{trace}(\boldsymbol{A}^T\boldsymbol{A})}\\
&= \frac{1}{N}\mathbb{E}[\|\boldsymbol{s}_t - \boldsymbol{x}\|_2^2] = v_t^2.
\end{aligned}$$

■

The justification of the error variance estimator (24) for $\bar{\tau}_t^2$ is also provided by the following proposition.

Proposition 2: If each entry of $\boldsymbol{s}_t - \boldsymbol{x}$ is i.i.d. with mean zero and $\mathbb{E}[(\boldsymbol{s}_t - \boldsymbol{x})^T \boldsymbol{W}\boldsymbol{w}] = 0$ is satisfied, then

$$\begin{aligned}
\bar{\tau}_t^2 = {} & \frac{\bar{v}_t^2}{N}(N - 2\gamma_t\mathsf{trace}(\boldsymbol{Z}) + \gamma_t^2\mathsf{trace}(\boldsymbol{Z}\boldsymbol{Z}^T))\\
& + \frac{\gamma_t^2\sigma^2}{N}\mathsf{trace}(\boldsymbol{W}\boldsymbol{W}^T)
\end{aligned} \tag{27}$$

holds, where $\boldsymbol{Z} = \boldsymbol{W}\boldsymbol{A}$.

(Proof) The residual error $\boldsymbol{r}_t - \boldsymbol{x}$ can be rewritten as

$$\begin{aligned}
\boldsymbol{r}_t - \boldsymbol{x} &= \boldsymbol{s}_t + \gamma_t\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}_t) - \boldsymbol{x}\\
&= \boldsymbol{s}_t + \gamma_t\boldsymbol{W}(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}) - \gamma_t\boldsymbol{W}\boldsymbol{A}\boldsymbol{s}_t - \boldsymbol{x}\\
&= (\boldsymbol{I} - \gamma_t\boldsymbol{Z})(\boldsymbol{s}_t - \boldsymbol{x}) + \gamma_t\boldsymbol{W}\boldsymbol{w}.
\end{aligned}$$

From the definition $\bar{\tau}_t^2$, we have

$$\begin{aligned}
\bar{\tau}_t^2 &= \frac{1}{N}\mathbb{E}[\|(\boldsymbol{I} - \gamma_t\boldsymbol{Z})(\boldsymbol{s}_t - \boldsymbol{x}) + \gamma_t\boldsymbol{W}\boldsymbol{w}\|_2^2]\\
&= \frac{1}{N}\mathbb{E}[(\boldsymbol{s}_t - \boldsymbol{x})^T(\boldsymbol{I} - \gamma_t\boldsymbol{Z})(\boldsymbol{I} - \gamma_t\boldsymbol{Z})^T(\boldsymbol{s}_t - \boldsymbol{x})]\\
&\quad + \frac{\gamma_t^2}{N}\mathbb{E}[\boldsymbol{w}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{w}]\\
&\quad + \frac{2\gamma_t}{N}\mathbb{E}[(\boldsymbol{s}_t - \boldsymbol{x})^T(I - \gamma_t\boldsymbol{Z})^T\boldsymbol{W}\boldsymbol{w}]\\
&= \frac{1}{N}\mathsf{trace}((\boldsymbol{I} - \gamma_t\boldsymbol{Z})(\boldsymbol{I} - \gamma_t\boldsymbol{Z})^T)\bar{v}_t^2\\
&\quad + \frac{\gamma_t^2}{N}\mathsf{trace}(\boldsymbol{W}\boldsymbol{W}^T)\sigma^2 + \frac{2(\gamma_t - \gamma_t^2)}{N}\mathbb{E}[(\boldsymbol{s}_t - \boldsymbol{x})^T\boldsymbol{W}\boldsymbol{w}].
\end{aligned}$$

The last term vanishes due to the assumption $\mathbb{E}[(\boldsymbol{s}_t - \boldsymbol{x})^T \boldsymbol{W}\boldsymbol{w}] = 0$, and the first term can be rewritten as

$$\begin{aligned}
&\mathsf{trace}((\boldsymbol{I} - \gamma_t\boldsymbol{Z})(\boldsymbol{I} - \gamma_t\boldsymbol{Z})^T)\\
&\quad = \sum_{i,j:i\neq j}(\gamma_t\boldsymbol{Z}_{i,j})^2 + \sum_i(1 - \gamma_t\boldsymbol{Z}_{i,i})^2\\
&\quad = \gamma_t^2\sum_{i,j:i\neq j}\boldsymbol{Z}_{i,j}^2 + \sum_i(1 - 2\gamma_t\boldsymbol{Z}_{i,i} + \gamma_t^2\boldsymbol{Z}_{i,i}^2)\\
&\quad = N - 2\gamma_t\mathsf{trace}(\boldsymbol{Z}) + \gamma_t^2\mathsf{trace}(\boldsymbol{Z}\boldsymbol{Z}^T).
\end{aligned} \tag{28}$$

The proposition is thus proved. ■

The identity $\mathsf{trace}(\boldsymbol{Z}) = \mathsf{trace}(\boldsymbol{Z}\boldsymbol{Z}^T) = M$ holds because $\boldsymbol{A}$ and $\boldsymbol{Z}$ have full rank. Combining this identity, we have the estimation formula (24) for τ_t^2.

These error variance estimators (23) and (24) play a crucial role in providing appropriate variance estimates required for the MMSE shrinkage. Since the validity of these assumptions on the residual errors cannot be proved, it will be experimentally confirmed in the next section. Moreover, note that the TISTA recursive formula does not include either an Onsager correction term or a divergence-free function. Thus, we cannot expect stochastic orthogonality guaranteed in OAMP in a process of TISTA. This means that the state evolution cannot be used to analyze the asymptotic performance of TISTA.

C. Time Complexity and Number of Trainable Variables

For treating a large-scale problem, a sparse recovery algorithm should require low computational complexity for each iteration. The time complexity required for evaluating the recursive formula of TISTA per iteration is $O(N^2)$, which is the same time complexity as those of ISTA and AMP. This fact means that the TISTA has sufficient scalability for large problems. The evaluation of the matrix-vector products $\boldsymbol{A}\boldsymbol{s}_t$ and $\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}_t)$ requires $O(N^2)$ time, which is dominant in an iteration. The evaluation of the scalar constants $\mathsf{trace}(\boldsymbol{A}^T\boldsymbol{A})$ and $\mathsf{trace}(\boldsymbol{W}\boldsymbol{W}^T)$ requires $O(N^2)$ time. Although computation of the pseudo inverse of $\boldsymbol{A}$ requires $O(N^3)$ time, it can be pre-computed only once in advance.

TABLE I
NUMBERS OF TRAINABLE VARIABLES IN THE T-ROUND PROCESS

	TISTA	LISTA	LAMP
# of params	$T+2$	$T(N^2+MN+1)$	$T(NM+2)$

Since the t'-th round of TISTA contains only trainable variables $\{\gamma_t\}_{t=0}^{t'-1}$ (or $\{\gamma_t\}_{t=0}^{t'-1}$, α^2 and p), the total number of trainable variables is T (or $T+2$) for TISTA with T iteration rounds. On the other hand, LISTA and LAMP require N^2+MN+1 and $NM+2$ trainable variables for each round, respectively. Table I summarizes the required numbers of trainable variables in T rounds. TISTA requires the least trainable variables among them, and the number of trainable variables of TISTA is independent of the system size, i.e., N and M. This is an advantageous feature for large-scale problems. The number of trainable variables also affects the stability and speed of convergence in training processes.

D. Incremental Training for TISTA

In order to achieve reasonable recovery performance, the trainable variables $\{\gamma_t\}_{t=0}^{T-1}$ (and possibly α^2 and p) should be appropriately adjusted. By unfolding the recursive formula of TISTA, we immediately have a signal-flow graph which is similar to a multi-layer feedforward neural network. Fig. 2 depicts a unit of the signal-flow graph corresponding to the t-th iteration of TISTA, and we can stack the units to compose a whole signal-flow graph. Here, we follow a standard recipe of deep learning techniques; namely, we apply mini-batch training with a stochastic gradient descent algorithm to the signal-flow graph of TISTA. Based on several experiments, we found that the following *incremental training* is considerably effective for learning appropriate values that provide superior performance. This is because the *vanishing gradient problem* makes one-shot training for the whole network difficult. The incremental training discussed below can reduce the effect of the vanishing gradient.

The training data consists of a number of randomly generated pairs $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$. The sample $\boldsymbol{x}$ follows the prior distribution $P_X(\boldsymbol{x})$ and the observation noise $\boldsymbol{w}$ is an i.i.d. Gaussian random vector. The entire set of training data is divided into mini-batches to be used in a stochastic gradient descent algorithm such as SGD, RMSprop, or Adam.

In the t-th round of the incremental training (referred to as a *generation*), an optimizer attempts to minimize $\mathbb{E}[\|\boldsymbol{s}_t - \boldsymbol{x}\|_2^2]$ by tuning $\{\gamma_{t'}\}_{t'=0}^{t-1}$ (and possibly α^2 and p). The number of mini-batches used in the t-th generation is denoted by D. After processing D mini-batches, the objective function of the optimizer is changed to $\mathbb{E}[\|\boldsymbol{s}_{t+1} - \boldsymbol{x}\|_2^2]$. Namely, after training the first to t-th layers, the new $(t+1)$-th layer is appended to the network, and the entire network is trained again for D mini-batches. Although the objective function is changed, the values of the variables $\gamma_0, \ldots, \gamma_{t-1}$ (and possibly α^2 and p) of the previous generation are taken as the initial values in the optimization process for the new generation. In summary, the incremental training updates the variables $\{\gamma_t\}$ in a sequential manner from the first layer to the last layer.

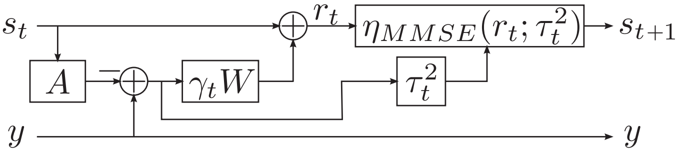

Fig. 2. Schematic diagram of the t-th iteration of TISTA with learnable variable γ_t.

IV. PERFORMANCE EVALUATION

In this section, the sparse recovery performance of TISTA is evaluated by computer experiments.

A. Details of Experiments

The basic conditions for the computer experiments shown in this section are summarized as follows. Each component of the sparse signal $\boldsymbol{x}$ is assumed to be a realization of an i.i.d. random variable following the Bernoulli-Gaussian PDF (15) with $p = 0.1, \alpha^2 = 1$. The Bernoulli-Gaussian PDF is often assumed as a benchmark setting in related researches [30], [31]. We thus use the MMSE estimator (22) for the Bernoulli-Gaussian prior. Each component of the noise vector $\boldsymbol{w}$ follows the zero-mean Gaussian PDF with variance σ^2. The signal-to-noise ratio (SNR) of the system is defined as

$$SNR = \frac{\mathbb{E}[\|\boldsymbol{A}\boldsymbol{x}\|_2^2]}{\mathbb{E}[\|\boldsymbol{w}\|_2^2]}. \quad (29)$$

The size of the mini-batch is set to 1000, and $D = 200$ mini-batches are allocated for each generation. We used the Adam optimizer [40]. The learning rate of the optimizer is set to 4.0×10^{-2} in the first 10 generations and 8.0×10^{-4} in the remaining generations. The experimental system was implemented in TensorFlow [41] and PyTorch [42]. For comparison purposes, we will include the NMSE performances of AMP and other algorithms in the following subsections. The hyperparameter θ used in AMP is set to $\theta = 1.14$. We used an implementation of LISTA [43] by the authors of [30].

B. IID Gaussian Matrix With Small Variance

Here, we consider the conventional setting for compressed sensing in which AMP successfully indicates convergence. The trainable parameters of TISTA in this subsection are $\{\gamma_t\}_{t=0}^{T-1}$, α^2, and p.

1) Comparison With AMP and Other Algorithms: This subsection describes the case in which $\boldsymbol{A}_{i,j} \sim \mathcal{N}(0, 1/M)$, i.e., each component of the sensing matrix $\boldsymbol{A}$ obeys a zero-mean Gaussian distribution with variance $1/M$. Note that AMP is designed for this matrix ensemble. The dimensions of the sensing matrices are set to be $N = 500, M = 250$.

Figure 3 shows the estimate τ^2 by (24) and the empirically estimated values of the true error variance $\bar{\tau}^2$. The estimator τ^2 provides accurate estimations, which justifies the use of (23) and (24), and our assumptions on the residual errors. We find that the error variance does not monotonically decrease. Because the residual error depends on the trainable parameters $\{\gamma_t\}_{t=0}^{T-1}$, the

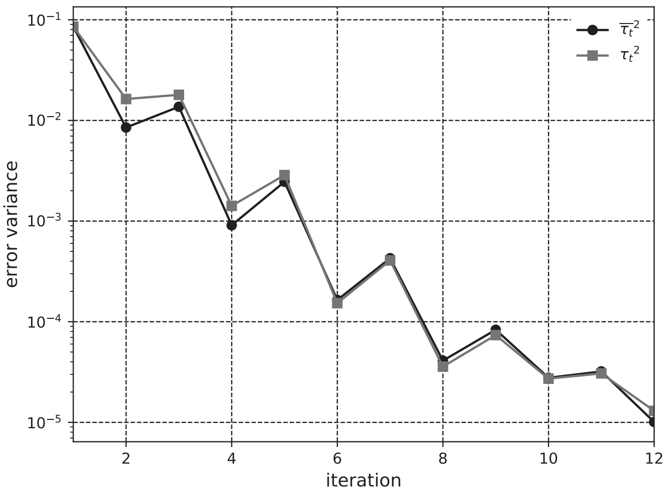

Fig. 3. Estimate $\bar{\tau}^2$ and the true error variance τ^2; $\boldsymbol{A}_{i,j} \sim \mathcal{N}(0, 1/M), N = 500, M = 250$, SNR = 40 dB.

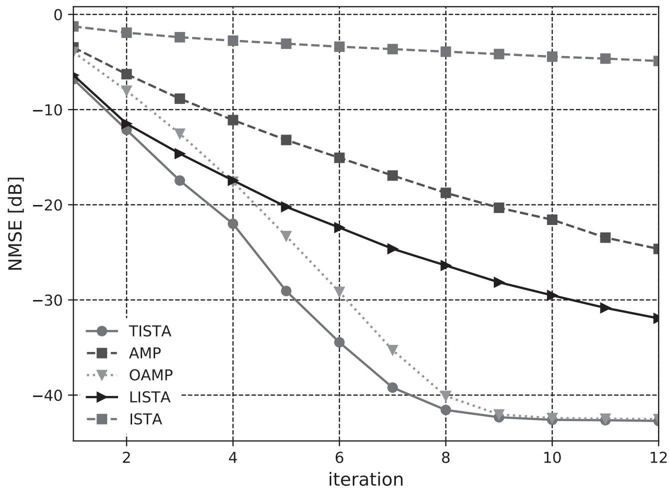

Fig. 4. NMSE of TISTA and other algorithms; $\boldsymbol{A}_{i,j} \sim \mathcal{N}(0, 1/M), N = 500, M = 250$, SNR = 40 dB. Condition $\boldsymbol{A}_{i,j} \sim \mathcal{N}(0, 1/M)$ is required for AMP to converge.

zigzag shape of γ_t's (see Fig. 11) may affect the shapes of τ^2 and $\bar{\tau}^2$. In spite of this nontrivial tendency, the residual error decreases rapidly indicating a successful signal recovery.

Figure 4 presents the average normalized MSE (NMSE) of TISTA, ISTA, LISTA, AMP, and OAMP as functions of iteration when SNR = 40 dB. The NMSE is defined by

$$NMSE = 10 \log_{10} \mathbb{E}\left[\frac{\|\boldsymbol{s}_{t+1} - \boldsymbol{x}\|_2^2}{\|\boldsymbol{x}\|_2^2}\right]. \tag{30}$$

In the experiment, the pseudo inverse matrix is chosen as the matrix $\hat{\boldsymbol{W}}$ in OAMP to make the time complexity $O(N^2)$ in each iteration. The divergence-free function of OAMP in (8) is based on the MMSE estimator (17).

From Fig. 4, we can observe that TISTA provides the steepest NMSE curve among those algorithms in the first 12 rounds. For example, OAMP and LISTA require 6 and 10 rounds, respectively, in order to achieve NMSE = −30 dB, whereas TISTA requires only 5 rounds. The NMSE curve of TISTA saturates at around −42 dB, at which TISTA and OAMP converge. This means that TISTA shows significantly faster convergence than AMP and LISTA in this setting. TISTA also overwhelms OAMP in the NMSE performance. TISTA has about 5.8 dB and 4.0 dB gains at $T = 5$ and 7 compared with OAMP, respectively.

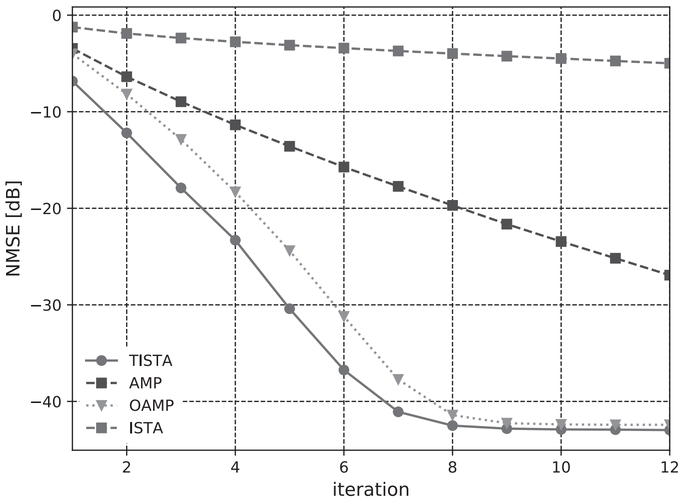

Fig. 5. NMSE of TISTA and and other algorithms; $N = 5000, M = 2500, p = 0.1, \boldsymbol{A}_{i,j} \sim \mathcal{N}(0, 1/M)$, SNR = 40 dB.

2) Large-Scale Problem: As discussed in the previous section, the number of trainable variables of TISTA is considerably small. This feature enables us to handle large-scale problems. Fig. 5 shows the NMSEs for the cases of $(N, M) = (5000, 2500)$. LISTA is omitted from the comparison because it is computationally intractable to execute in our environment. We find that the NMSE performance of each algorithm are slightly better than that in the small system ($N = 500$). The gain of TISTA, however, is still large in this case. In addition, TISTA saturates about −43 dB, which is 0.6 dB lower than OAMP. From these observations, we find that TISTA exhibits a good NMSE performance even in a large system.

3) Running Time: In order to demonstrate the scalability of TISTA explicitly, we show the CPU time required for training processes in Fig. 6. The CPU time is measured by a PC with Intel Xeon(R) CPU (3.6 GHz, 6 cores) and no GPUs. It consists of the whole incremental training process up to T layers and execution process of TISTA implemented by PyTorch 0.4.1. In the experiment, we fix the rate M/N to 0.5 and SNR to 40 dB as the same setting with the previous experiments. The results show that, in the case of $N = 500$, TISTA is about 37 times faster than LISTA in addition to better NMSE performance as shown in Fig. 4. We also find that TISTA has a notable scalability. The CPU time of TISTA ($T = 7$) for $N = 10^4$ signals is nearly equal to that of LISTA ($T = 7$) for $N = 500$. Simple linear regressions estimate that the CPU time roughly depends on $N^{1.2}$ and $T^{2.0}$. These facts suggest that the small number of trainable parameters in TISTA enables its fast learning process for large problems.

C. Gaussian Sensing Matrices With Large Variance

In the next experiment, we changed the variance of the sensing matrices to a larger value, i.e., each element in $\boldsymbol{A}$ follows

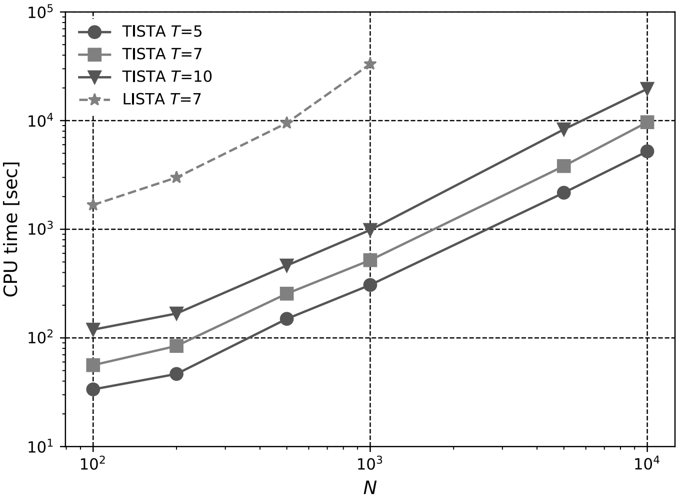

Fig. 6. CPU time for learning and executing TISTA (solid lines) and LISTA (dashed line) as a function of N with various T; $\boldsymbol{A}_{i,j} \sim \mathcal{N}(0, 1/M)$, $M/N = 0.5$, SNR= 40 dB.

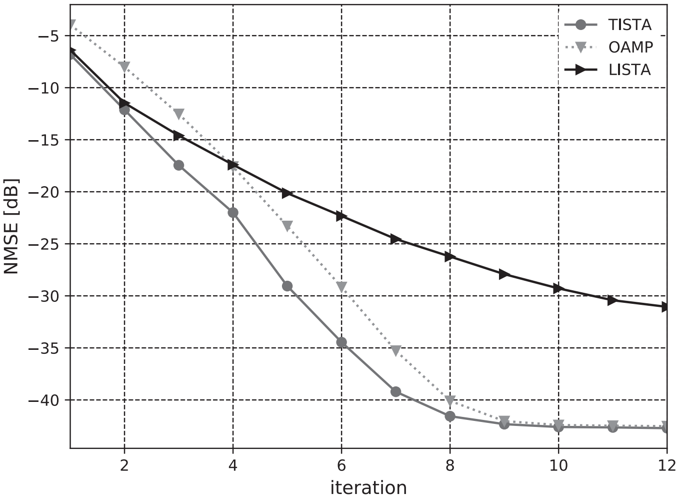

Fig. 7. NMSE of TISTA, OAMP, and LISTA; $\boldsymbol{A}_{i,j} \sim \mathcal{N}(0, 1)$, $N = 500, M = 250$, SNR = 40 dB. In this case, AMP cannot converge because the variance of the matrix components is too large.

$\mathcal{N}(0, 1)$ instead of $\mathcal{N}(0, 1/M)$. The trainable parameters of TISTA are $\{\gamma_t\}_{t=0}^{T-1}$, α^2, and p. Fig. 7 shows the NMSE curves of TISTA, OAMP, and LISTA. Note that, under this condition, AMP does not perform well, i.e., AMP actually cannot converge at all, because the setting does not fit the required condition ($\boldsymbol{A}_{i,j} \sim \mathcal{N}(0, 1/M)$) for achieving its guaranteed performance. As shown in Fig. 7, TISTA behaves soundly and shows faster convergence than that of OAMP and LISTA. This result suggests that TISTA is appreciably robust against the change of the variance.

D. Binary Matrix

In this subsection, we will discuss the case in which the sensing matrices are binary, i.e., $\boldsymbol{A} \in \{\pm 1\}^{M \times N}$. Each entry of $\boldsymbol{A}$ is selected uniformly at random on $\{\pm 1\}$. This situation is closely related to multiuser detection in Coded Division Multiple Access (CDMA) [11]. Fig. 8 shows the NMSE curves of TISTA, OAMP, and LISTA as a function of iteration. As the previous subsections, TISTA trains $\{\gamma_t\}_{t=0}^{T-1}$, α^2, and p. The NMSE curves of TISTA approximately coincide with those of the Gaussian sensing matrices. This result can be regarded as an evidence for the robustness of TISTA for non-Gaussian sensing matrices.

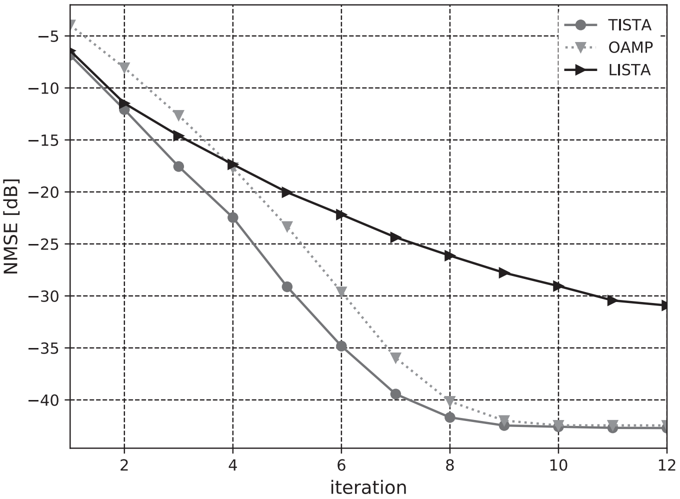

Fig. 8. NMSE of TISTA, OAMP, and LISTA; $\boldsymbol{A}_{i,j}$ takes a value in $\{\pm 1\}$ uniformly at random. $N = 500$, $M = 250$, SNR = 40 dB. AMP is not applicable in this case.

E. Sensing Matrices With a Large Condition Number

Regression problems regarding a matrix with a large condition number are difficult to solve in an accurate manner. The condition number κ of a matrix is defined as the ratio of the largest and smallest singular values, i.e., $\kappa = s_1/s_M$, where $s_1 \geq s_2 \geq \cdots \geq s_M$ are the singular values of the matrix. In this subsection, we assess the performance of TISTA for sensing matrices with a large condition number. In this subsection, the trainable parameters of TISTA are only $\{\gamma_t\}_{t=0}^{T-1}$ because it shows enough performance improvement.

The setting for the experiments is as follows. For a given condition number κ, we assume that the ratio s_i/s_{i-1} is constant for each i in order to fulfill $s_1/s_M = \kappa$ and $\mathsf{trace}(\boldsymbol{A}\boldsymbol{A}^T) = N$. We first sample a matrix $\boldsymbol{G} \in \mathbb{R}^{M \times N}$, where each entry of $\boldsymbol{G}$ follows an i.i.d. zero-mean Gaussian distribution with variance 1. The matrix $\boldsymbol{G}$ is then decomposed by singular value decomposition and we obtain $\boldsymbol{G} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, where $\boldsymbol{U} \in \mathbb{R}^{M \times M}$, $\boldsymbol{V} \in \mathbb{R}^{N \times N}$, and $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times N}$. From the set of singular values $s_1, \ldots, s_M$ satisfying the above conditions, $\boldsymbol{\Sigma}^*$ is defined by $\boldsymbol{\Sigma}^* = (\boldsymbol{\Delta}\ \boldsymbol{O})$, where the matrix $\boldsymbol{\Delta} = \mathrm{diag}(s_1, \ldots, s_M)$, and $\boldsymbol{O}$ is the zero matrix. A sensing matrix $\boldsymbol{A}$ with the condition number κ is obtained by calculating $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}^*\boldsymbol{V}^T$.

Figure 9 shows the NMSE of TISTA and AMP without observation noise, i.e., $\sigma^2 = 0$. As shown in Fig. 9, there is almost no performance degradation in the NMSE even for a large condition number such as $\kappa = 5000$. On the other hand, AMP converges

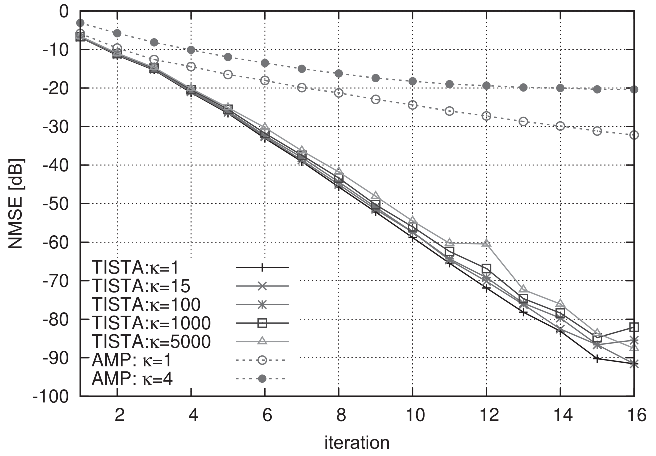

Fig. 9. NMSE of TISTA and AMP; κ represents the condition number. No observation noise ($\sigma^2 = 0$).

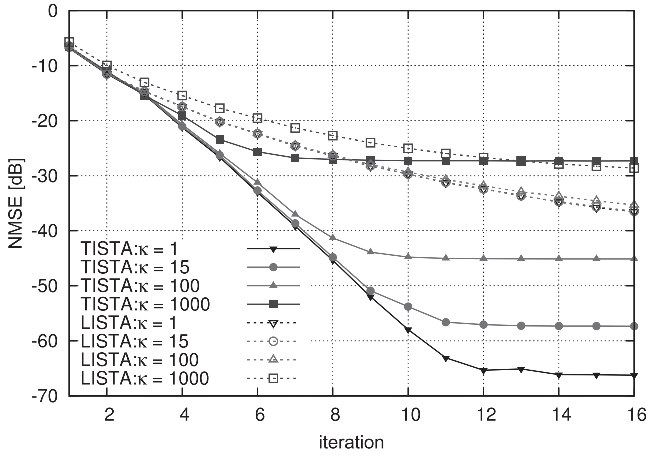

Fig. 10. NMSE of TISTA and LISTA; κ represents the condition number. SNR $= 60$ dB.

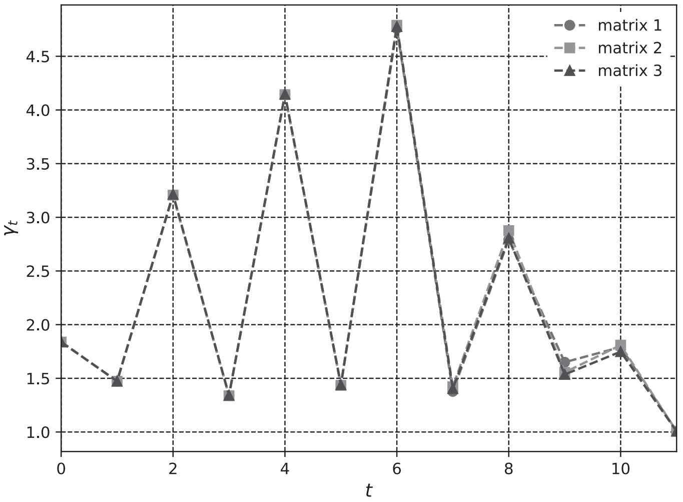

Fig. 11. Three sequences of learned variables γ_t; $\boldsymbol{A}_{i,j} \sim \mathcal{N}(0, 1/M), N = 500, M = 250, p = 0.1$, SNR $= 40$ dB.

up to $\kappa = 4$, but the output diverges when $\kappa \geq 5$. These results indicate the robustness of TISTA with respect to sensing matrices with a large condition number in the noiseless case.

Figure 10 shows the NMSE of TISTA and LISTA when there are observation noises (SNR $= 60$ dB). Compared with the NMSE curve of LISTA, TISTA provides a much smaller NMSE in the cases of $\kappa = 1, 15, 100$. However, in contrast to the noiseless case (Fig. 9), the NMSE performance of TISTA severely degrades as κ increases. This phenomenon can be considered as a consequence of the use of the pseudo inverse linear estimator $\boldsymbol{W}$, which tends to cause noise enhancement if the condition number is large.

F. Trained Parameters

In order to study the behavior of the learned trainable variables $\{\gamma_t\}$, we conducted the following experiments. For a fixed sensing matrix ($(N, M) = (500, 250)$, $\boldsymbol{A}_{i,j} \sim \mathcal{N}(0, 1/M)$), we trained TISTA ($T = 12$) three times with distinct random number seeds. The learned variables $\{\gamma_t\}$ (denoted by matrix 1–3) are shown in Fig. 11. The three sequences of learned parameters approximately coincide with each other. Furthermore, the sequences have a zigzag shape, and the values of γ_t lie in the range from 1 to 10. As for other trainable parameters, α^2 is tuned to 3.68–3.71 and p is tuned to 0.08–0.09. Interestingly, the trained α^2 becomes larger than the true value 1.0 though p does not change largely from the true value 0.1. Note that training these values improves the NMSE performance of TISTA, which suggests that the true values of parameters in the MMSE estimator are not always best for TISTA.

To explain the zig-zag shape of learned parameters $\{\gamma_t\}_{t=0}^{T-1}$, we show a toy example where the shape of trainable parameters accelerates the convergence speed of an iterative algorithm. Let us consider a gradient descent (GD) method which minimizes a quadratic function $f(x_1, x_2) = x_1^2 + 10x_2^2$. The function is simple but the condition number regarding the problem is relatively large. This means that a naive GD method is not suitable for attaining fast convergence to the minimum point. The main step of the GD method is the update of the search point as

$$\boldsymbol{s}_{t+1} = \boldsymbol{s}_t - \gamma \nabla f(\boldsymbol{s}_t) \tag{31}$$

for $t = 0, 1, \ldots, T-1$. The parameter γ is the step size parameter that significantly affects the behavior of the search process. In this section, we assume that each element of the initial point $\boldsymbol{s}_1 = (s_{1,1}, s_{1,2})$ is chosen in the closed domain $[-10, 10]^2$ uniformly at random.

Figure 12 (center, bottom) shows typical minimization processes of the GD method. A small step size (center) leads to considerably slow convergence but a large step size (bottom) induces oscillation behaviors that also slow down the convergence or lead to divergence.

According to the idea of TISTA, i.e., embedding of trainable parameters, we can embed trainable parameters in the GD step as

$$\boldsymbol{s}_{t+1} = \boldsymbol{s}_t - \gamma_t \nabla f(\boldsymbol{s}_t), \tag{32}$$

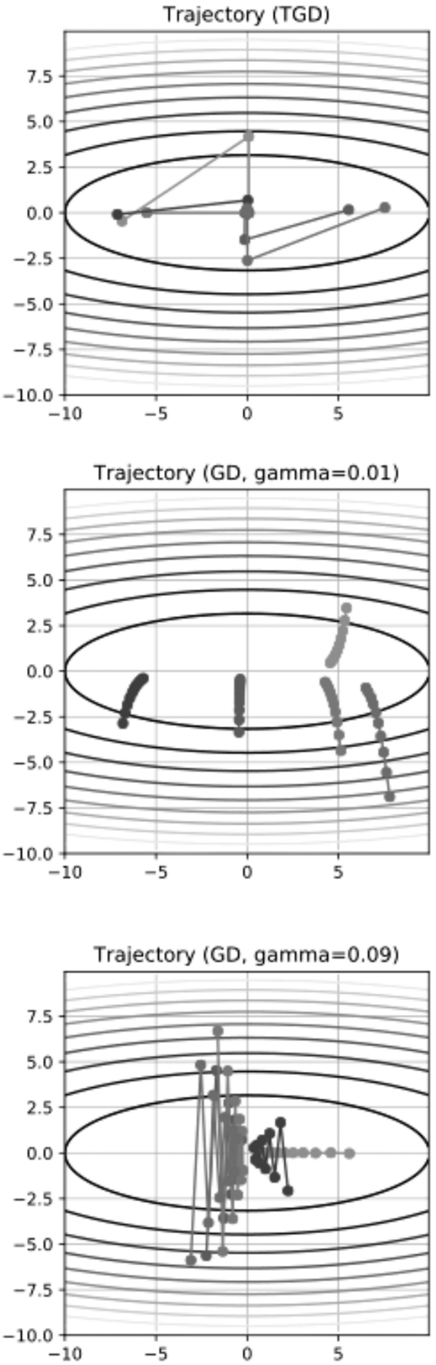

Fig. 12. Trajectories of search points (5 trials) in GD processes for $f(x_1, x_2) = x_1^2 + 10x_2^2$: TGD (top), GD with $\gamma = 0.01$ (center), GD with $\gamma = 0.09$ (bottom). The optimal point is $(0, 0)$. The ovals are contour of the objective function.

where $\{\gamma_t\}_{t=0}^{T-1}$ is a set of trainable parameters. The incremental training can be applied to train these parameters in order to accelerate the convergence. We call this method the trainable GD (TGD) hereafter.

Figure 13 shows the averaged error of TGD and GD as a function of the number of iterations. TGD significantly outperforms GD methods and provides much faster convergence. From the training process, TGD learns an appropriate strategy to yield

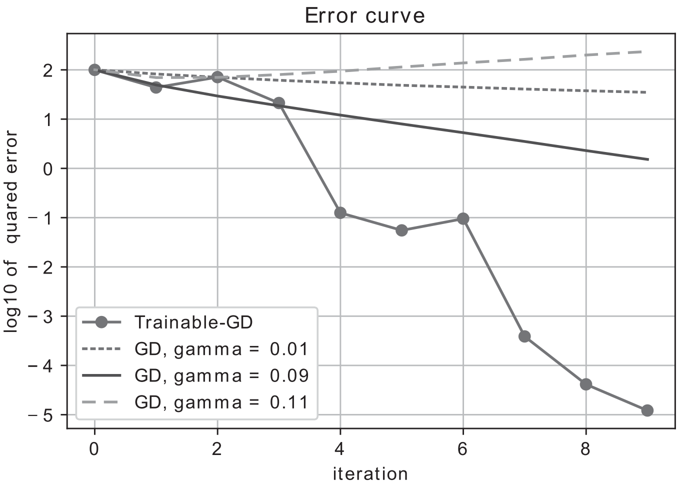

Fig. 13. Averaged error curves of TGD and GD: The horizontal axis represents the number of iterations and the vertical axis represents the averaged error $\log_{10} \|\boldsymbol{s}_{t+1} - \boldsymbol{s}^*\|_2^2$ where $\boldsymbol{s}_{t+1}$ is the search point after t iterations, and $\boldsymbol{s}^*$ is the optimal solution. In the evaluation process, the outcomes of 10000 minimization trials with random starting points are averaged.

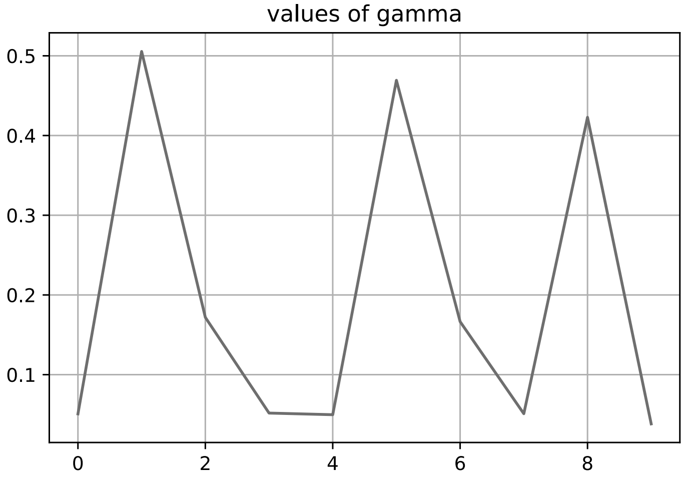

Fig. 14. Trained values of γ_i: the details of the training is as follows. The incremental training with the mini-batch size 50 is used. In a generation, 500 mini-batches are processed. The optimizer is Adam with learning rate 0.001.

fast convergence. The trained values of $\{\gamma_t\}_{t=0}^{T-1}$ are plotted in Fig. 14. We can observe a zigzag shape that represents the learned acceleration strategy for this problem. It is interesting to see that the behavior of the search point shown in Fig. 12 (top) is not similar to those of $\gamma = 0.01$ (center) nor $\gamma = 0.09$ (bottom).

Our hypothesis of the zigzag shapes is that a similar situation happens in signal recovery processes of TISTA as well. The linear estimation step (21) of TISTA is closely related to the gradient descent step for the quadratic problem to minimize $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$, i.e., we have the exact gradient descent step by replacing $\boldsymbol{W}$ with $\boldsymbol{A}^T$. If the quadratic problem is ill-conditioned or nearly ill-conditioned, the preferable strategy would be the *zigzag strategy* observed in Fig. 14 as well. We still lack enough evidences to confirm the validity of the hypothesis and it should be confirmed in a future work.

V. Sparse Signal Recovery for MNIST Images

In Sec. IV, we have seen results of the numerical experiments based on artificial sparse signals generated according to the i.i.d. Bernoulli-Gaussian prior model. The feasibility of TISTA for sparse signals in the real world has not yet been clear because a real sparse signal may not follow the i.i.d. assumption. In order to evaluate the performance of TISTA for non-i.i.d. signals, we made experiments of sparse signal recovery based on the MNIST dataset. The MNIST dataset is a dataset including monochrome images of hand-written numerals and the corresponding labels. Since most of pixels of an MNIST image is zero, the MNIST dataset can be regarded as a dataset of sparse signals. The goal of this section is to discuss the sparse signal recovery performance of TISTA for the MNIST dataset.

The details of the experiment is as follows. An MNIST image consists $28 \times 28 = 784$ pixels where a pixel takes an integer value from 0 to 255. We first normalize the pixel values to $[0, 1]$ and then rasterize the pixels as 784-dimensional vectors. In the following, we let $N = 784$ and $M = 392$. As a sensing matrix, we prepare a random matrix $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ where each element in $\boldsymbol{A}$ follows Gaussian distribution with zero mean and variance $1/M$. We assume a noisy observation by the matrix $\boldsymbol{A}$ with the additive Gaussian noise $\boldsymbol{w}$ with zero mean and variance 4×10^{-4}, i.e., the received signal $\boldsymbol{y}$ is generated by $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$. As a sparse signal recovery algorithms, we compare TISTA with OAMP. We choose the MMSE estimator (17) for the Bernoulli-Gaussian prior as their MMSE functions because we assume that we have no knowledge on the prior PDF of the images. We set the parameters of the prior to $\alpha^2 = 1.0$, $p = 0.5$ for OAMP while these parameters are trained from the dataset in TISTA.

The detail of the training processes is as follows. In the training process of TISTA, as well as $\{\gamma_t\}_{t=0}^{T-1}$, the parameters α^2 and p are treated as trainable parameters. The size of mini-batch is set to 200. For a generation of incremental training, we used all the images in the MNIST training set (60000 images). Adam optimizer with learning rate 0.005 was used for training.

Figure 15 shows the recovered images by TISTA (left column) and OAMP (right column) with $t = 1, 4, 8$ iterations. These images are recovered from the same noisy observation of the original image displayed on the left bottom. It can be observed that TISTA with $t = 8$ provides a reconstructed image considerably close to the original ($\mathrm{MSE} = 0.0091$). The number "0" is not perfectly recovered because the original image is not so sparse. The quality of the reconstructed images of TISTA evidently outperforms that of OAMP. For example, even with $t = 100$, the image reconstruction by OAMP ($\mathrm{MSE} = 0.0148$) is worse than that by TISTA in terms of MSE. In fact, we find that the reconstructed "2" by OAMP is not so crisp and clear compared with those of TISTA (right bottom of Fig. 15). It implies that the training parameters α^2 (trained value 1.59) and p (trained value 0.4) positively affects the image reconstruction quality.

Moreover, comparing the images of $t = 1, 4, 8$, it can be confirmed that TISTA shows much faster convergence than OAMP. This tendency exactly coincides with the results reported in Section IV.

The result of this section strongly suggests that TISTA can be applied to sparse signal recovery problems based on the real data with non-i.i.d. sparse signals if we have enough data to train the trainable parameters.

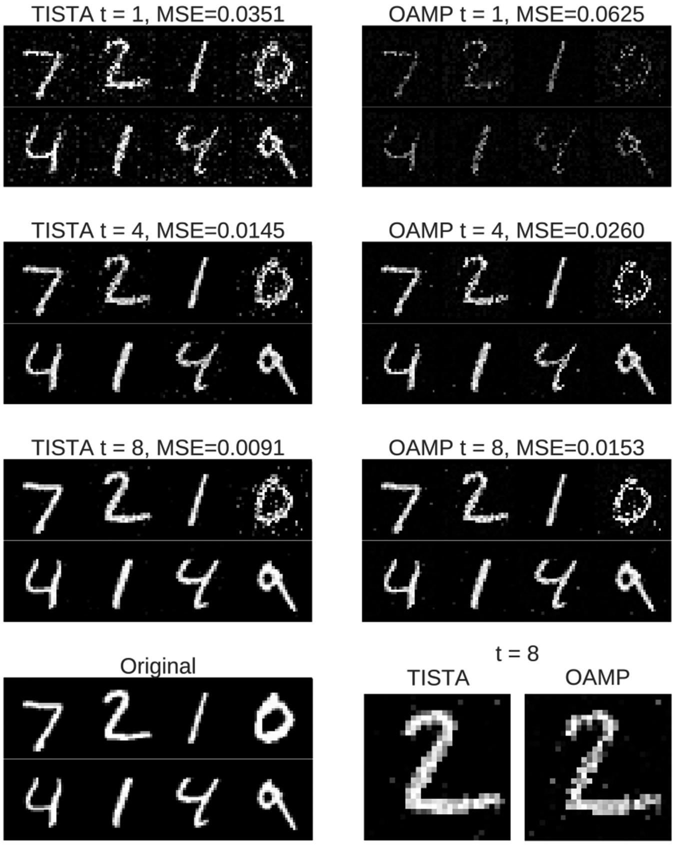

Fig. 15. Reconstructed images by TISTA (left column) and OAMP (right column). Parameters: $N = 784, M = 392$, $\boldsymbol{A}_{i,j} \sim \mathcal{N}(0, 1/M)$, noise variance 4×10^{-4}. The "2" images reconstructed by TISTA and OAMP with $t = 8$ are shown in the right bottom for comparison.

VI. Extensions

In this section, we propose a few extensions of TISTA to treat a sensing matrix with nonzero-mean components or with a large condition number. The numerical results show that the proposed extensions outperform the original TISTA in each situation without additional computational costs in the learning process. In this section, the trainable parameters of TISTA are only $\{\gamma_t\}_{t=0}^{T-1}$.

A. Sensing Matrices With Nonzero-Mean Components

In this subsection, we propose an extension of TISTA for a sensing matrix with nonzero-mean components. It is known that, e.g., generalized AMP [44] (GAMP), which is constructed for zero-mean Gaussian random matrices, fails to converge to a fixed point when a sensing matrix consists of nonzero-mean components [35]. To overcome this difficulty, Vila *et al.* proposed a variant of GAMP with damping of messages and mean removal from a sensing matrix and signals [45]. Following these advances in AMP, we apply a mean removal technique to TISTA to improve its performance for large nonzero-mean sensing matrices.

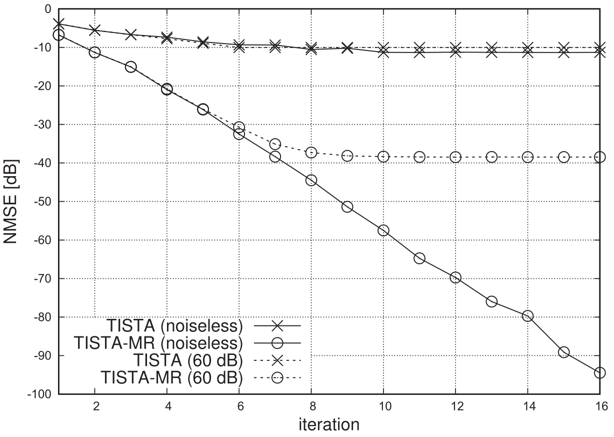

Fig. 16. NMSE of the original TISTA (cross marks) and TISTA-MR (circles) with mean removal; $A_{i,j} \sim \mathcal{N}(1, 1/M)$, $N = 500$, $M = 250$. No observation noise ($\sigma^2 = 0$) and SNR = 60 dB cases.

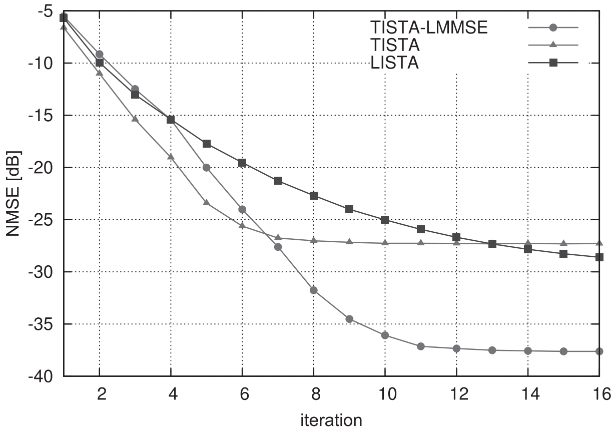

Fig. 17. NMSE of LISTA, the original TISTA, and TISTA-LMMSE with (39) ($\beta = 5.0 \times 10^{-4}$); condition number $\kappa = 1000$, SNR = 60 dB.

Let us consider *TISTA-MR*, TISTA with the mean removal technique. We assume that the sensing matrix $\boldsymbol{A}$ is generated according to the Gaussian distribution $\mathcal{N}(\mu_{\boldsymbol{A}}, \sigma^2)$ with a nonzero mean μ_A. In fact, without any modifications, TISTA shows poor performance as μ_A increases. The simplest extension involves the use of a modified sensing matrix $\boldsymbol{A}' = (\boldsymbol{A}'_{i,j})$, where $\boldsymbol{A}'_{i,j} = \boldsymbol{A}_{i,j} - \mu_A$ instead of an original sensing matrix $\boldsymbol{A} = (\boldsymbol{A}_{i,j})$. The modified recursion formula of TISTA is then written as follows:

$$\boldsymbol{u}_t = \boldsymbol{y} - \boldsymbol{A}'\boldsymbol{s}_t, \tag{33}$$

$$\boldsymbol{r}_t = \boldsymbol{s}_t + \gamma_t \boldsymbol{W}' \left(\boldsymbol{u}_t - \frac{1}{M}\boldsymbol{1}_M^T \boldsymbol{u}_t \boldsymbol{1}_M \right) \tag{34}$$

$$\boldsymbol{s}_{t+1} = \eta_{MMSE}(\boldsymbol{r}_t; \tau_t^2) \tag{35}$$

$$v_t^2 = \max\left\{ \frac{\|\boldsymbol{u}_t - \frac{1}{M}\boldsymbol{1}_M^T \boldsymbol{u}_t \boldsymbol{1}_M\|_2^2 - M\sigma^2}{\mathsf{trace}(\boldsymbol{A}'^T\boldsymbol{A}')}, \epsilon \right\} \tag{36}$$

$$\tau_t^2 = \frac{v_t^2}{N}(N + (\gamma_t^2 - 2\gamma_t^2)M) + \frac{\gamma_t^2\sigma^2}{N}\mathsf{trace}(\boldsymbol{W}'\boldsymbol{W}'^T), \tag{37}$$

where $\boldsymbol{1}_M = (1, 1, \ldots, 1)^T$ is an M-dimensional vector, the elements of which are 1s, and matrix $\boldsymbol{W}'$ is the pseudo inverse matrix of $\boldsymbol{A}'$. In the formula, $\boldsymbol{r}_t$ is calculated via $\boldsymbol{u}_t - M^{-1}\boldsymbol{1}_M^T \boldsymbol{u}_t \boldsymbol{1}_M$ to remove the mean of $\boldsymbol{u}_t$. These modifications enable the performance of TISTA-MR to be improved because it attempts to recover a sparse signal with a modified sensing matrix, the components of which have sufficiently small means. Note that further performance improvement may be achieved when we use a modified sensing matrix for which the means of rows and columns are expected to be zero, as in [45].

Figure 16 shows the NMSE of the original TISTA and TISTA-MR for noiseless case in the case of noiseless observations and noisy observations with SNR = 60 dB. Each element of a sensing matrix $\boldsymbol{A}$ is generated from $\mathcal{N}(1, 1/M)$, where the original AMP has difficulty in convergence. TISTA-MR outperforms the original TISTA for which the NMSE saturates around -10 dB in both cases. In the case of SNR = 60 dB, TISTA-MR scores -38 dB in the NMSE with about 28 dB gain against TISTA when $T = 10$. These numerical results indicate that TISTA-MR based on mean removal gives drastically improved signal recovery performance without increasing the time complexity.

B. Sensing Matrices With a Large Condition Number

As discussed in the previous section, TISTA exhibits a non-negligible performance degradation (except for the noiseless case) when the condition number of the sensing matrix is large. In this subsection, we present a method for improving the sparse recovery performance of TISTA in such a case by using an LMMSE-like matrix as a linear estimator. A naive approach to suppress the noise enhancement in linear estimation is to use the LMMSE-like matrix

$$\boldsymbol{W}_t = v_t^2 \boldsymbol{A}^T (v_t^2 \boldsymbol{A}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I})^{-1} \tag{38}$$

as a linear estimator in TISTA recursions. Note that the error variance v_t^2 is calculated in a recursive calculation process of TISTA. Ma and Ping [15] took this approach in their OAMP experiments. A drawback of this approach is that it is necessary to calculate an $M \times M$ matrix inversion in (38) for each iteration, which requires $O(M^3)$ time for an iteration. In order to avoid the matrix inversion for each iteration, we use a simple ad-hoc solution, and define the matrix $\boldsymbol{W}$ as

$$\boldsymbol{W} = \boldsymbol{A}^T (\boldsymbol{A}\boldsymbol{A}^T + \beta \boldsymbol{I})^{-1}, \tag{39}$$

where β is a real constant. We call TISTA with (39) *TISTA-LMMSE*. This is the only difference from the original TISTA using the pseudo inverse matrix of $\boldsymbol{A}$ as $\boldsymbol{W}$. The term $\beta\boldsymbol{I}$ can decrease the condition number of $\boldsymbol{W}$ and prevents noise enhancement. Matrix inversion is necessary only once at the beginning of a recovery process. Thus, the required time complexity of TISTA-LMMSE is the same as that of the original TISTA. The parameter β is determined to minimize the value of the NMSE after training.

Figure 17 shows the NMSE curves for the case of $\kappa = 1000$, which includes the NMSE curve of TISTA-LMMSE with (39).

In TISTA-LMMSE, we used the parameter $\beta = 5.0 \times 10^{-4}$. From Fig. 17, we can confirm that TISTA-LMMSE exhibits much better NMSE performance as compared with the original TISTA using the pseudo inverse matrix in the linear estimator. This example shows that this simple ad-hoc approach is fairly effective without additional costs.

VII. Conclusion

The crucial feature of TISTA is that it includes adjustable variables which can be tuned by standard deep learning techniques. The number of trainable variables of TISTA is equal to the number of iterative rounds and is much smaller than those of the known learnable sparse signal recovery algorithms [29]–[31]. This feature leads to the highly stable and fast training processes of TISTA. Computer experiments indicate that TISTA is applicable to various classes of sensing matrices such as Gaussian matrices, binary matrices, and matrices with large condition numbers. Furthermore, numerical results demonstrate that TISTA shows significantly faster convergence than AMP or LISTA in many cases and remarkably large gains compared to OAMP. The experimental results on the MNIST image set imply that TISTA is also applicable for non-i.i.d. sparse signals in the real world. In summary, TISTA achieves remarkable performance improvement for artificial data and promising flexibility to real data with fast learning process, high stability, and high scalability using a quite simple architecture.

For a future plan, by replacing the MMSE shrinkage function, we can expect that TISTA is also applicable to non-sparse signal recovery problems such as detection of BPSK signals in overloaded MIMO systems [46]. Another possibility is to replace the MMSE shrinkage function with a small neural network that can learn an appropriate shrinkage function matched to the prior of the sparse signals. This change could significantly broaden the target of TISTA.

Acknowledgment

The authors would like to thank the anonymous reviewers of the IEEE International Conference on Communications 2018 and IEEE Transactions on Signal Processing for their constructive comments. T. Wadayama would like to thank Dr. K. Takeuchi for the inspiring seminar at the Nagoya Institute of Technology.

References

[1] D. L. Donoho, “Compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[2] E. J. Candes and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?” *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[3] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, “A survey of sparse representation: Algorithms and applications,” *IEEE Access*, vol. 3, pp. 490–530, 2015.

[4] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. Roy. Statist. Soc. Ser. B*, vol. 58, pp. 267–288, 1996.

[5] G. Davis, S. Mallat, and M. Avellaneda, “Adaptive greedy approximations,” *Constructive Approx.*, vol. 13, no. 1, pp. 57–98, Mar. 1997.

[6] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, Apr. 2004.

[7] T. T. Wu and K. Lange, “Coordinate descent algorithms for lasso penalized regression,” *Ann. Appl. Statist.*, vol. 2, no. 1, pp. 224–244, 2008.

[8] A. Chambolle, R. A. DeVore, N. Lee, and B. J. Lucier, “Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage,” *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 319–335, Mar. 1998.

[9] I. Daubechies, M. Defrise, and C. De Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Aug. 2004.

[10] N. Parikh and S. Boyd, “Proximal algorithms,” *Found. Trends Optim.*, vol. 1, no. 3, pp. 123–231, 2014.

[11] Y. Kabashima, “A CDMA multiuser detection algorithm on the basis of belief propagation,” *J. Phys. A: Math. General*, vol. 36 pp. 11111–11121, Oct. 2003.

[12] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 45, pp. 18914–18919, Nov. 2009.

[13] D. L. Donoho, A. Maleki, and A. Montanari, “Message passing algorithms for compressed sensing: I. Motivation and construction,” in *Proc. IEEE Inf. Theory Workshop*, Jan. 2010, pp. 1–5.

[14] M. Bayati and A. Montanari, “The dynamics of message passing on dense graphs, with applications to compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 764–785, Jan. 2011.

[15] J. Ma and L. Ping, “Orthogonal AMP,” *IEEE Access*, vol. 5, pp. 2020–2033, 2017.

[16] S. Rangan, P. Schniter, and A. K. Fletcher, “Vector approximate message passing,” in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2017, pp. 1588–1592.

[17] K. Takeuchi, “Rigorous dynamics of expectation-propagation-based signal recovery from unitarily invariant measurements,” in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2017, pp. 501–505.

[18] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980.

[19] M. Riesenhuber and T. Poggio, “Hierarchical models of object recognition in cortex,” *Nature Neurosci.*, vol. 2, no. 11, pp. 1019–1025, Nov. 1999.

[20] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, Jun. 2006.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2012, pp. 1097–1105.

[22] G. Hinton *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[23] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[24] B. Aazhang, B. P. Paris, and G. C. Orsak, “Neural networks for multiuser detection in code-division multiple-access communications,” *IEEE Trans. Commun.*, vol. 40, no. 7, pp. 1212–1222, Jul. 1992.

[25] E. Nachmani, Y. Beéry, and D. Burshtein, “Learning to decode linear codes using deep learning,” in *Proc. 54th Annu. Allerton Conf. Commun., Control, Comput.*, 2016, pp. 341–346.

[26] T. O’Shea and J. Hoydis, “An introduction to deep learning for the physical layer,” *Proc. IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[27] Y. A. LeCun, L. Bottou, G. B. Orr, and K. R. Müller, “Efficient backprop,” in *Neural Networks: Tricks of the Trade*, G. B. Orr and K. R. Müller, Eds. London, U.K.: Springer-Verlag, 1998, pp. 9–50.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[29] K. Gregor and Y. LeCun, “Learning fast approximations of sparse coding,” in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

[30] M. Borgerding and P. Schniter, “Onsager-corrected deep learning for sparse linear inverse problems,” in *Proc. IEEE Global Conf. Signal Inf. Process.*, Washington, DC, USA, Dec. 2016, pp. 227–231.

[31] M. Borgerding, P. Schniter, and S. Rangan, “AMP-inspired deep networks for sparse linear inverse problems,” *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4293–4308, Aug. 2017.

[32] J. Zhang and B. Ghanem, “ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing,” *IEEE/CVF Conf. Comput. Vis. Pattern Recogn.*, Salt Lake City, UT, pp. 1828–1837, 2018.

[33] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[34] J. M. Bioucas-Dias and M. A. T. Figueiredo, “A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration,” *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2992–3004, Dec. 2007.

[35] F. Caltagirone, L. Zdeborova, and F. Krzakala, “On convergence of approximate message passing,” in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2014, pp. 1812–1816.

[36] J. Vila and P. Schniter, “Expectation-maximization Gaussian-mixture approximate message passing,” *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4658–4672, Oct. 2013.

[37] P. Schniter, L. C. Potter, and J. Ziniel, “Fast Bayesian matching pursuit,” in *Proc. Inf. Theory Appl. Workshop*, Jan. 2008, pp. 326–333.

[38] R. Gribonval, “Should penalized least squares regression be interpreted as maximum a posteriori estimation?” *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2405–2410, May. 2011.

[39] A. Kazerouni, U. S. Kamilov, E. Bostan, and M. Unser, “Bayesian denoising: From MAP to MMSE using consistent cycle spinning,” *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 249–252, Mar. 2013.

[40] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *Proc. 3rd Int. Conf. Learning Represent.* (ICLR 2015), San Diego, CA, USA, May 2015.

[41] “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. [Online]. Available: http://tensorflow.org/

[42] A. Paszke *et al.*, “Automatic differentiation in PyTorch,” in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017, pp. 1–4. [Online]. Available: pytorch.org

[43] 2017. [Online]. Available: https://github.com/mborgerding/onsager_deep_learning/blob/master/README.md

[44] S. Rangan, “Generalized approximate message passing for estimation with random linear mixing,” in *Proc. IEEE Int. Symp. Inf. Theory*, Aug. 2011, pp. 2168–2172.

[45] J. Vila, P. Schniter, S. Rangan, F. Krzakala, and L. Zdeborova, “Adaptive damping and mean removal for the generalized approximate message passing algorithm,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2015, pp. 2021–2025.

[46] R. Hayakawa and K. Hayashi, “Convex optimization-based signal detection for massive overloaded MIMO systems,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7080–7091, Nov. 2017.

Daisuke Ito received the B.E. and M.E. degrees from the Nagoya Institute of Technology, Nagoya, Japan, in 2016 and 2018, respectively. His research interests include signal processing and machine learning.

Satoshi Takabe (M’17) received the B.Sc., M.Sc., and Ph.D. degrees in multidisciplinary sciences from The University of Tokyo, Tokyo, Japan, in 2012, 2014, and 2017, respectively. He is currently an Assistant Professor with the Department of Computer Science, Nagoya Institute of Technology, Nagoya, Japan. His research interests include signal processing, information theory, and machine learning. He is a member of the Institute of Electronics, Information and Communication Engineers, Operations Research Society of Japan, and JPS. He was the recipient of the IEEE Information Theory Society Japan Chapter Young Researcher Best Paper Award in 2018.

Tadashi Wadayama (M’96) was born in Kyoto, Japan, on May 9, 1968. He received the B.E., M.E., and D.E. degrees from the Kyoto Institute of Technology, Kyoto, Japan, in 1991, 1993, and 1997, respectively. In 1995, he was with the Faculty of Computer Science and System Engineering, Okayama Prefectural University, as a Research Associate. From April 1999 to March 2000, he was with the Institute of Experimental Mathematics, Essen University (Germany), as a Visiting Researcher. In 2004, he moved to the Nagoya Institute of Technology, Nagoya, Japan, as an Associate Professor. Since 2010, he has been a Full Professor with the Nagoya Institute of Technology. His research interests are in coding theory, information theory, and coding and signal processing for digital communication/storage systems. He is a member of the Institute of Electronics, Information and Communication Engineers.