Methods and Applications of Analysis.
Vol. 26, No. 3, pp. 235–248, September 2019

WEIGHTED NONLOCAL TOTAL VARIATION IN IMAGE PROCESSING*

HAOHAN LI[†], ZUOQIANG SHI[‡], AND XIAO-PING WANG[§]

Abstract. In this paper, a novel weighted nonlocal total variation (WNTV) method is proposed. Compared to the classical nonlocal total variation methods, our method modifies the energy functional to introduce a weight to keep balance between the labeled sets and unlabeled sets. With extensive numerical examples in semi-supervised clustering, image inpaiting and image colorization, we demonstrate that WNTV provides an effective and efficient method in many image processing and machine learning problems.

Key words. Total variation, nonlocal method, point cloud, nonlocal Laplacian.

Mathematics Subject Classification. 65D05, 65D25, 41A05.

1. Introduction.

Interpolation on point cloud in high dimensional space is a fundamental problem in many machine learning and image processing applications. It can be formulated as follows. Let $V = \{\mathbf{p}_1, \cdots, \mathbf{p}_n\}$ be a set of points in $\mathbb{R}^d$ and $S = \{\mathbf{s}_1, \cdots, \mathbf{s}_m\}$ be a subset of V. Let u be a function on the point set V and the value of u on $S \subset V$ is given as a function g over S. The goal of the interpolation is to find the function u on V with the given values on S.

Since the point set V is unstructured in high dimensional space, traditional interpolation methods do not apply. In recent years, manifold learning has been demonstrated to be effective and attract more and more attentions. One basic assumption in manifold learning is that the point cloud V samples a low dimensional smooth manifold, $\mathcal{M}$, embedded in $\mathbb{R}^d$. Another assumption is that the interpolation function u is a smooth function in $\mathcal{M}$. Based on these two assumptions, one popular approach is to solve u by minimizing the L_2 norm of its gradient in $\mathcal{M}$. This gives us an optimization problem to solve:

$$\min_u \|\nabla_{\mathcal{M}} u\|_2, \quad \text{subject to:} \quad u(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in S, \tag{1}$$

with

$$\|\nabla_{\mathcal{M}} u\|_2 = \left(\int_{\mathcal{M}} |\nabla_{\mathcal{M}} u(\mathbf{x})|^2 d\mathbf{x} \right)^{1/2}.$$

This model is closely related to nonlocal methods in image processing [1, 2, 4, 5]. Usually, $\nabla_{\mathcal{M}} u$ is approximated by nonlocal gradient

$$D_{\mathbf{y}} u(\mathbf{x}) = \sqrt{w(\mathbf{x}, \mathbf{y})}(u(y) - u(\mathbf{x})). \tag{2}$$

Then, the discrete version of (1) is

$$\min \sum_{\mathbf{x},\mathbf{y} \in V} w(\mathbf{x}, \mathbf{y})(u(\mathbf{x}) - u(\mathbf{y}))^2, \quad \text{subject to:} \quad u(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in S, \tag{3}$$

*Received January 30, 2018; accepted for publication April 12, 2019. Research supported by NSFC Grant 11671005.

[†]Department of Mathematics, The Hong Kong University of Science and Technology, Hong Kong (hlibb@connect.ust.hk).

[‡]Corresponding author. Department of Mathematical Sciences & Yau Mathematical Sciences Center, Tsinghua University, Beijing, China (zqshi@tsinghua.edu.cn).

[§]Department of Mathematics, The Hong Kong University of Science and Technology, Hong Kong (mawang@ust.hk).

from which, we can derive a linear system to solve u on point cloud P, which is given as follows,

$$\begin{cases} \sum_{\mathbf{y}\in V}(w(\mathbf{x},\mathbf{y})+w(\mathbf{y},\mathbf{x}))(u(\mathbf{x})-u(\mathbf{y}))=0, & \mathbf{x}\in V\backslash S, \\ u(\mathbf{x})=g(\mathbf{x}), & \mathbf{x}\in S. \end{cases}$$

This is just the well known nonlocal Laplacian which is widely used in nonlocal methods of image processing [1, 2, 4, 5]. It is also called graph Laplacian in graph and machine learning literature [3, 13]. Recently, it was found that, when the sample rate is low, i.e. $|S|/|P| \ll 1$, graph Laplacian method fails to give a smooth interpolation [12, 11]. Continuous interpolation can be obtained by using point integral method [12] or weighted nonlocal Laplacian [11].

In many problems, such as data classification or image segmentation, minimizing the total variation seems to be a better way to compute the interpolation function, since it prefers piecewise constant solutions. This observation motives another optimization problem:

$$\min \|u\|_{TV_{\mathcal{M}}}, \quad \text{subject to:} \quad u(\mathbf{x})=g(\mathbf{x}), \quad \mathbf{x}\in S, \tag{4}$$

with

$$\|u\|_{TV_{\mathcal{M}}} = \int_{\mathcal{M}} |\nabla_{\mathcal{M}} u(\mathbf{x})| d\mathbf{x}.$$

Total variation model has been studied extensively in image processing since it was first proposed by Rudin, Osher and Fatemi(ROF) in [10]. It is well known that total variation has the advantage of preserving edges, which is always preferable because edges are significant features in the image, and usually indicate boudaries of objects. Despite its good performance of restoring ”cartoon” part of the image, TV based methods fail to achieve satisfactory results when texture, or repetative structures, are present in the image. To address this problem, Buades et al proposed a nonlocal means method based on patch distances for image denoising [1]. Later, Gilboa and Osher [4, 5] formalized a systematic framework, include nonlocal total variation model, for nonlocal image processing.

Using nonlocal gradient to approximate the total variation, we can write down the discrete version of (4),

$$\min \sum_{\mathbf{x}\in V} \left(\sum_{\mathbf{y}\in V} w(\mathbf{x},\mathbf{y})(u(\mathbf{x})-u(\mathbf{y}))^2 \right)^{1/2}, \quad \text{subject to:} \quad u(\mathbf{x})=g(\mathbf{x}), \quad \mathbf{x}\in S. \tag{5}$$

This problem can be solved efficiently by split Bregman iteration [8, 6]. However, it was reported in [9], when the sample rate is low, above nonlocal TV model has the same defect as that in graph Laplacian approach (3). The interpolation obtained by solving above optimization problem is not continuous on the sample points.

In this paper, inspired by weighted nonlocal Laplacian method proposed in [11], we propose a weighted nonlocal TV method (WNTV) to fix this discontinuous issue. The idea is to introduce a weight related to the sample rate to balance the labeled terms and unlabeled terms. More specifically, we modify model (5) a little bit by introducing a weight,

$$\min_u \sum_{\mathbf{x}\in V\setminus S}\left(\sum_{\mathbf{y}\in V}\omega(\mathbf{x},\mathbf{y})(u(\mathbf{x})-u(\mathbf{y}))^2\right)^{1/2}+\frac{|V|}{|S|}\sum_{\mathbf{x}\in S}\left(\sum_{\mathbf{y}\in V}\omega(\mathbf{x},\mathbf{y})(u(\mathbf{x})-u(\mathbf{y}))^2\right)^{1/2},$$
$$\text{subject to:}\quad u(\mathbf{x})=g(\mathbf{x}),\quad \mathbf{x}\in S.$$

$|V|,|S|$ are the number of points in sets V and S. This optimization problem also can be solved by split Bregman iteration. Based on our experience, the convergence is even faster than the split Bregman iteration in the original nonlocal total variation model (5). Extensive examples in image inpaiting, semi-supervised learning, image colorization demonstrate that the weighted nonlocal total variation model has very good performance. It provides an effective and efficient method for many image processing and machine learning problem.

The rest of the paper is organized as follows. In section 1, we review the interpolation problem on point cloud, which is typically hard to solve by traditional interpolation method. Then the weighted nonlocal TV method (WNTV) is introduced in section 2. We apply the split Bregman iteration algorithm to our method, which is a well-known algorithm to solve a very broad class of L1-regularization problems. Numerical experiments including semi-supervised clustering, image inpainting and image colorization are shown in section 3, 4 and 5 respectively. Here we compared our results to those obtained using graph Laplacian, nonlocal TV and weighted nonlocal Laplacian. Conclusions are made in the section 6.

2. Weighted Nonlocal TV.

As introduced at the beginning of the introduction, we consider an interpolation problem in a high dimentional point cloud. Let $V=\{\mathbf{p}_1,\cdots,\mathbf{p}_n\}$ be a set of points in $\mathbb{R}^d$ and $S=\{\mathbf{s}_1,\cdots,\mathbf{s}_m\}$ be a subset of V. u is a function on V and $u(\mathbf{s})=g(\mathbf{s})$, $\forall\mathbf{s}\in S$ with given g. We assume that V samples a smooth manifold $\mathcal{M}$ embedded in $\mathbb{R}^d$ and we want to minimize the total variation of u on $\mathcal{M}$ to solve u on the whole point cloud V. Direct nonlocal gradient approximation gives nonlocal total variation model (5).

When the sample rate is low, the solution of nonlocal total variation model (5) is not continuous as reported in [9, 11]. As shown in Fig. 1, recovered images by model (5) are not continuous on sample pixels.

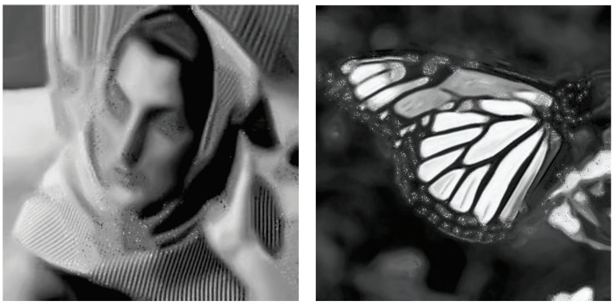

(a) (b)

Fig. 1: *Recovered image by Nonlocal TV model* (5) *from 10% random samples.*

To remove the discontinuity, inspired by the weighted nonlocal Laplacian method proposed by Shi et. al. in [11], we modify the functional in (5) by adding a weight to balance the energy between labeled and unlabeled sets:

$$\begin{aligned}&\min_u \sum_{\mathbf{x}\in V\backslash S}\left(\sum_{\mathbf{y}\in V}\omega(\mathbf{x},\mathbf{y})(u(\mathbf{x})-u(\mathbf{y}))^2\right)^{1/2}+\frac{|V|}{|S|}\sum_{\mathbf{x}\in S}\left(\sum_{\mathbf{y}\in V}\omega(\mathbf{x},\mathbf{y})(u(\mathbf{x})-u(\mathbf{y}))^2\right)^{1/2}\\ &\text{subject to:}\quad u(\mathbf{x})=g(\mathbf{x}),\quad \mathbf{x}\in S,\end{aligned}\tag{6}$$

where S is a subset of the vertices set V, and $|V|,|S|$ are the number of points in sets V and S, respectively. The idea is that when the sample rate is low, the summation over the unlabeled set overwhelms the summation over the labeled set such that the continuity on the labeled set is sacrificed. To maintain the continuity of the interpolation on the labeled points, we introduce a weight to balance the labeled term and the unlabeled term.

The weighted nonlocal total variation model (WNTV) (6) can be solved by split Bregman iteration [6]. To simplify the notation, we introduce an operator as follows,

$$D_{NG}u(\mathbf{x},\mathbf{y})=\begin{cases}\sqrt{\omega(\mathbf{x},\mathbf{y})}(u(\mathbf{x})-u(\mathbf{y})), & \textbf{if}\quad \mathbf{x}\in V\backslash S,\\ \frac{|V|}{|S|}\sqrt{\omega(\mathbf{x},\mathbf{y})}(u(\mathbf{x})-u(\mathbf{y})), & \textbf{if}\quad \mathbf{x}\in S.\end{cases}$$

With above operator, WNTV model (6) can be rewritten as

$$\begin{aligned}&\min_{u,D}\sum_{x\in V}\left(\sum_{y\in V}|D(\mathbf{x},\mathbf{y})|^2\right)^{1/2},\\ &\text{subject to:}\quad D(\mathbf{x},\mathbf{y})=D_{NG}u(\mathbf{x},\mathbf{y}),\quad u(\mathbf{x})=g(\mathbf{x}),x\in S.\end{aligned}\tag{7}$$

We then use Bregman iteration to enforce the constraint $D(\mathbf{x},\mathbf{y})=D_{NG}u(\mathbf{x},\mathbf{y})$ to get a two-step iteration,

$$\begin{aligned}(u^{k+1},D^{k+1})=&\arg\min_{u,D}\sum_{x\in V}\left(\sum_{y\in V}|D(\mathbf{x},\mathbf{y})|^2\right)^{1/2}\\ &+\frac{\lambda}{2}\sum_{x\in V}\sum_{y\in V}\left(D(\mathbf{x},\mathbf{y})-D_{NG}u(\mathbf{x},\mathbf{y})-Q^k(\mathbf{x},\mathbf{y})\right)^2,\\ &\text{subject to:}\quad u(\mathbf{x})=g(\mathbf{x}),\quad \mathbf{x}\in S.\end{aligned}\tag{8}$$

$$Q^{k+1}=Q^k+(D_{NG}u^{k+1}-D^{k+1}),\tag{9}$$

where λ is a positive parameter.

In above iteration, (9) is easy to solve. To solve the minimization problem (8), we use the idea in the split Bregman iteration to solve u and D alternatively.

$$u^{k+1}=\arg\min_u ||D^k-D_{NG}u-Q^k||_2^2,\quad \text{subject to:}\quad u(\mathbf{x})=g(\mathbf{x}),\quad \mathbf{x}\in S.\tag{10}$$

$$D^{k+1}=\arg\min_D \|D\|_1+\frac{\lambda}{2}||D-D_{NG}u^{k+1}-Q^k||_2^2.\tag{11}$$

$$Q^{k+1}=Q^k+(D_{NG}u^{k+1}-D^{k+1}).\tag{12}$$

where

$$\|D\|_1=\sum_{\mathbf{x}\in V}\left(\sum_{\mathbf{y}\in V}|D(\mathbf{x},\mathbf{y})|^2\right)^{1/2}.$$

The first step is a standard least-squares problem. It is staightforward to see that u^{k+1} satisfies a linear system,

$$\begin{aligned}
&\sum_{\mathbf{y}\in V\backslash S}(\omega(\mathbf{x},\mathbf{y})+\omega(\mathbf{y},\mathbf{x}))(u(\mathbf{x})-u(\mathbf{y}))+\sum_{\mathbf{y}\in S}\left(\omega(\mathbf{x},\mathbf{y})+\left(\frac{|V|}{|S|}\right)^2\omega(\mathbf{y},\mathbf{x})\right)(u(\mathbf{x})-u(\mathbf{y}))\\
&-\sum_{\mathbf{y}\in V\backslash S}\left((D^k(\mathbf{x},\mathbf{y})-Q^k(\mathbf{x},\mathbf{y}))\sqrt{\omega(\mathbf{x},\mathbf{y})}-(D^k(\mathbf{y},\mathbf{x})-Q^k(\mathbf{y},\mathbf{x}))\sqrt{\omega(\mathbf{y},\mathbf{x})}\right)\\
&-\sum_{\mathbf{y}\in S}\left((D^k(\mathbf{x},\mathbf{y})-Q^k(\mathbf{x},\mathbf{y}))\sqrt{\omega(\mathbf{x},\mathbf{y})}-\frac{|V|}{|S|}(D^k(\mathbf{y},\mathbf{x})-Q^k(\mathbf{y},\mathbf{x}))\sqrt{\omega(\mathbf{y},\mathbf{x})}\right)=0,\\
&\mathbf{x}\in V\backslash S, \qquad (13)\\
&u(\mathbf{x})=g(\mathbf{x}),\quad \mathbf{x}\in S. \qquad (14)
\end{aligned}$$

The linear system (13)-(14) looks like complicated. Its coefficient matrix is sparse, symmetric and postive definite which can be solved efficiently by conjugate gradient method.

The minimizer of the optimization problem (11) can be explicitly computed using shrinkage operators.

$$D_{\mathbf{x}}^{k+1}=\text{shrink}(D_{NG\mathbf{x}}u^{k+1}+Q_{\mathbf{x}}^k,1/\lambda)$$

where

$$\text{shrink}(\mathbf{z},\gamma)=\frac{\mathbf{z}}{||\mathbf{z}||_2}\max(||\mathbf{z}||_2-\gamma,0).$$

Summarizing above discussion, we get an iterative algorithm to solve weighted nonlocal total variation model,

1. Solve (13)-(14) to get u^{k+1}.
2. Compute D^{k+1} by

$$D^{k+1}(\mathbf{x},\mathbf{y})=\frac{\bar{D}(\mathbf{x},\mathbf{y})}{\left(\sum_{\mathbf{y}\in V}|\bar{D}(\mathbf{x},\mathbf{y})|^2\right)^{1/2}}\max\left(\sqrt{\sum_{\mathbf{y}\in V}|\bar{D}(\mathbf{x},\mathbf{y})|^2}-\frac{1}{\lambda},0\right)$$

with $\bar{D}(\mathbf{x},\mathbf{y})=\sqrt{\omega(\mathbf{x},\mathbf{y})}(u^{k+1}(\mathbf{x})-u^{k+1}(\mathbf{y}))+Q^k(\mathbf{x},\mathbf{y})$.

3. Update Q by

$$Q_{ij}^{k+1}=Q_{ij}^k+((D_{NG}u^{k+1})_{ij}-D_{ij}^{k+1}).$$

Algorithm 1: Algorithm for WNTV

3. Semi-supervised Clustering. In this section, we test WNTV in a semi-supervised clustering problem on the famous MNIST data set [7]. The MNIST database consists of 70,000 grayscale 28×28 pixel images of handwritten digits, see Fig. 2, which is divided into a training set of 60,000 examples, and a test set of 10,000

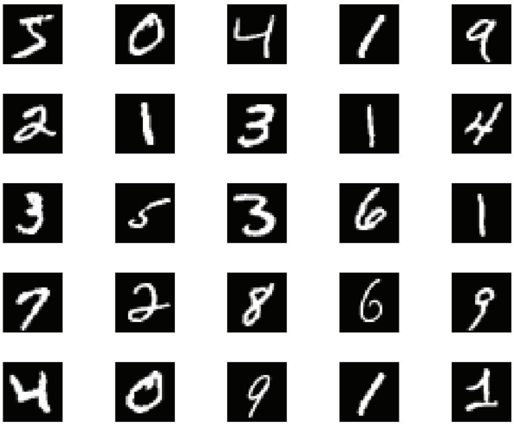


Fig. 2: *Some examples in the MNIST handwritten digits dataset*

examples. The images include digits from 0 to 9, which can be viewed as 10 classes segmentation.

From geometrical point of view, 70,000 28×28 images form a point cloud V in 784-dimension Euclidean space. In the tests, we randomly select a small subset $S \subset V$ to label,

$$S = \bigcup_{i}^{l} S_i,$$

where S_i is a subset of S with label i. Our task here is to label the rest of unlabeled images. The algorithm we used is summarized in Algorithm 2.

```
Data: A set of points $V$ with a small subset labeled $S = \bigcup_{i}^{l} S_i$
Result: labels of the whole points set $V$
1. Compute the corresponding weight function $\omega(\mathbf{x},\mathbf{y})$ for $\mathbf{x},\mathbf{y} \in V$;
for $i = 0:9$ do
    2. Compute $u_i$ by WNTV using Algorithm 1 with the constraint
        $u_i(\mathbf{x}) = 1,\ \mathbf{x} \in S_i, \quad u_i(\mathbf{x}) = 0,\ \mathbf{x} \in S \backslash S_i.$
end
3. Label $x \in V \setminus S$ as $k$ when $k = \arg\max_{1 \le i \le l} u_i(x)$
```

Algorithm 2: Semi-Supervised Learning

In our experiment of MNIST dataset, the weight function $\omega(\mathbf{x},\mathbf{y})$ is then constructed using the Gaussian,

$$\omega(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma(\mathbf{x})^2}\right),$$

Table 1: *Rate of correct classification in percentage for MNIST dataset*

Methods	700/70000	100/70000	50/70000
WNTV	94.08	89.86	78.35
Nonlcal TV	93.78	32.55	28.00
WNLL	93.25	87.84	73.60
GL	93.15	35.17	20.09

where $\| \cdot \|$ denotes the Euclidean distance, $\sigma(\mathbf{x})$ is the distance between $\mathbf{x}$ and its 10th nearest neighbor. The weight $\omega(\mathbf{x}, \mathbf{y})$ is made sparse by setting $\omega(\mathbf{x}, \mathbf{y})$ equal to zero if point $\mathbf{y}$ is not among the 20th closest points to point $\mathbf{x}$.

From the result of table (1), we can see that with high label rate (700/70000), all four methods give good classification. Nevertheless, as the label rate is reduced (100/70000, 50/70000), graph Laplacian and nonlocal TV both fail. The results given by WNTV and WNLL still have reasonable accuracy. WNTV is slightly better than WNLL in our tests.

4. Image Inpainting.

The problem of fitting the missing pixels of a corrupted image is always of interest in image processing. This problem can be formulated as an interpolation problem on point cloud by considering patches of the image. Consider a discrete image $f \in \mathbb{R}^{m \times n}$, around each pixel (i, j), we define a patch $p_{ij}(f)$ that is $s_1 \times s_2$ collection of pixels of image f. The collection of all patches is defined to be the patch set $\mathscr{P}(f)$ [9],

$$\mathscr{P}(f) = \{p_{ij}(f) : (i, j) \in \{1, 2, ..., m\} \times \{1, 2, ..., n\}\}.$$

Here $\mathscr{P}(f)$ forms a point set V.

Then the image can be viewed as a function u on the point cloud $\mathscr{P}(f)$. u is defined to be the intensity of the central pixel of the patch,

$$u(p_{ij}(f)) = f(i, j),$$

where $f(i, j)$ is the intensity of pixel (i, j).

Now, given subsample of the image, the problem of image inpainting is to fit the missing value of u on the patch set $\mathscr{P}(f)$. However, this problem is actually more difficult than the interpolation, since the patches is also unknown. In the image inpainting, we also need to recover the point cloud in addition to the interpolation function. We achieve this by a simple iterative scheme. First, we fill in the missing pixels by random number to get a complete image. For this complete (quality is bad) image, we construct point cloud by extracting patches. On this point cloud, we run WNTV to compute an interpolation function. From this interpolation function, we can construct an image. Then the patch set is updated from this new image. By repeating this process until convergence, we get the restoration of the image. We summarize this ideas in algorithm (3).

```
Data: A subsampled image
Result: Recovered image $u$
initialize $u^0$ such that $u^0_S = f_S$ and $D^0, Q^0 = 0$;
while not converge do
    1. Construct patch set $\mathscr{P}(u^n)$ from the current recovered image $u^n$ at step $n$;
    2. Compute the corresponding weight function $\omega^n(\mathbf{x},\mathbf{y})$ for $\mathbf{x},\mathbf{y} \in \mathscr{P}(u^n)$;
    3. Compute $u^{n+1}$ by solving system (1),then update image correspondingly;
    4. goto step 1;
end
```

Algorithm 3: Image Inpainting

4.1. Grayscale image inpainting. We first apply the algorithm to grayscale images. In this case, we also use Gaussian weight,

$$\omega(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma(\mathbf{x})^2}\right)$$

where $\|\mathbf{x}-\mathbf{y}\|^2$ is the Euclidean distance between patches $\mathbf{x}$ and $\mathbf{y}$. $\sigma(\mathbf{x})$ is the distance between $\mathbf{x}$ and its 20th nearest neighbor. The weight $\omega(\mathbf{x},\mathbf{y})$ is made sparse by setting $\omega(\mathbf{x},\mathbf{y})$ equal to zero if point $\mathbf{y}$ is not among the 50th closest points to point $\mathbf{x}$. For each pixel, we assign a 11×11 patch around it consisting of intensity values of pixels in the patch. In order to accelerate the speed of convergence, we use the semi-local patch by adding the local coordinate to the end of the patches,

$$p_{ij}(I) = [p_{ij}, \lambda_1 i, \lambda_2 j]$$

where

$$\lambda_1 = \frac{3||f_S||_\infty}{m}, \quad \lambda_2 = \frac{3||f_S||_\infty}{n}.$$

An approximate nearest neighbor algorithm (ANN) is used to obtain nearest neighbors. We use the Peak Signal-to-Noise Ratio (PSNR) to measure the quality of restored images,

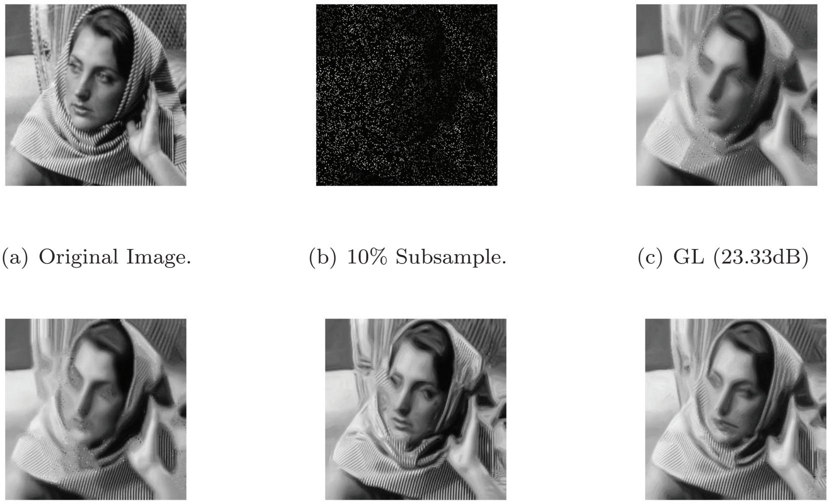
$$\mathrm{PSNR}(u, u_{gt}) = -20\log_{10}(\|u - u_{gt}\|/255)$$

where u and u_{gt} are the restored image and the original image respectively.

The results are displayed in Fig. 3, 4 and 5. For each image, we fix the number of iterations to be 10. As we can see, WNTV and WNLL performs much better than classical nonlocal TV method and graph Laplacian. The results of WNLL are comparable to proposed WNTV. As expected, WNTV works better for cartoon image as shown in Fig. 5. Evolutions of PSNR for Barbara image is shown in Fig. 6. As we can see, WNTV outperforms NTV in PSNR.

4.2. Color Image Inpainting. Now, we apply the algorithm to color images. The basic settings are similar to the grayscale image examples. In color image, patch becomes a 3D cube. The size we used is $11 \times 11 \times 3$. We also use Gaussian weight,

$$\omega(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma(\mathbf{x})^2}\right)$$



(a) Original Image.

(b) 10% Subsample.

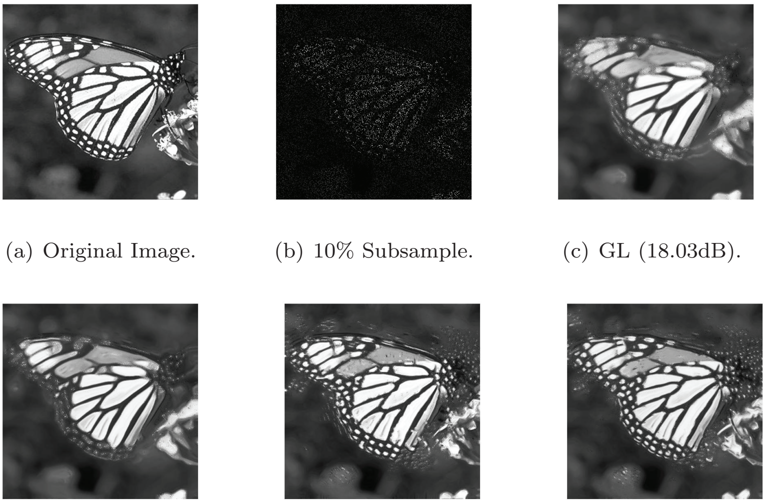
(c) GL (23.33dB)

(d) NTV (22.85dB).

(e) WNLL (25.35dB).

(f) WNTV (25.52dB).

Fig. 3: *Results of Graph Laplacian (GL), nonlocal TV (NTV), weighted nonlocal Laplacian (WNLL) and weighted nonlocal TV (WNTV) in image of Barbara.*



(a) Original Image.

(b) 10% Subsample.

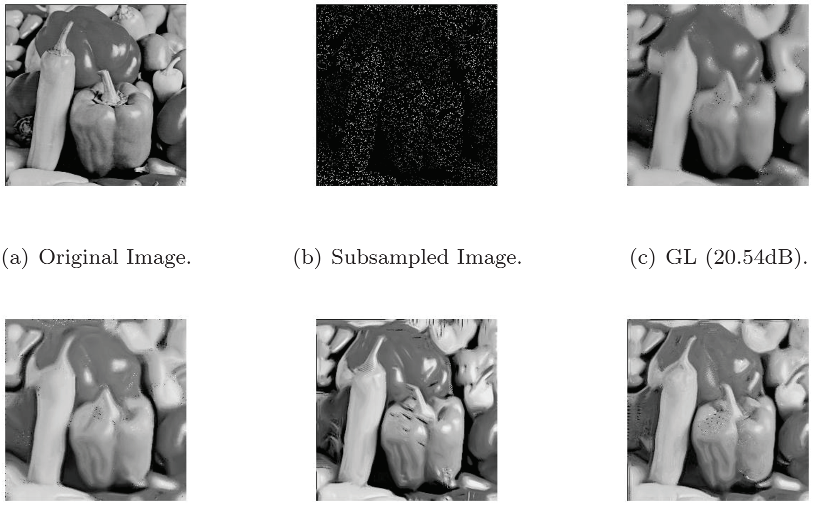
(c) GL (18.03dB).

(d) NTV (17.89dB).

(e) WNLL (20.28dB).

(f) WNTV (20.46dB).

Fig. 4: *Results of Graph Laplacian (GL), nonlocal TV (NTV), weighted nonlocal Laplacian (WNLL) and weighted nonlocal TV (WNTV) in the butterfly image.*



(a) Original Image.

(b) Subsampled Image.

(c) GL (20.54dB).

(d) NTV (20.93dB).

(e) WNLL (22.80dB).

(f) WNTV (23.03dB).

Fig. 5: *Results of Graph Laplacian (GL), nonlocal TV (NTV), weighted nonlocal Laplacian (WNLL) and weighted nonlocal TV (WNTV) in the pepper image.*

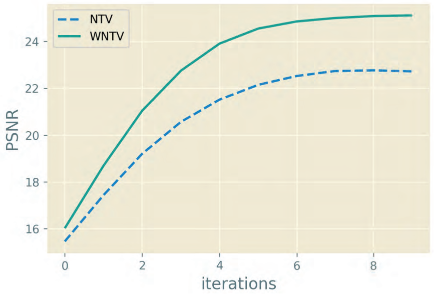
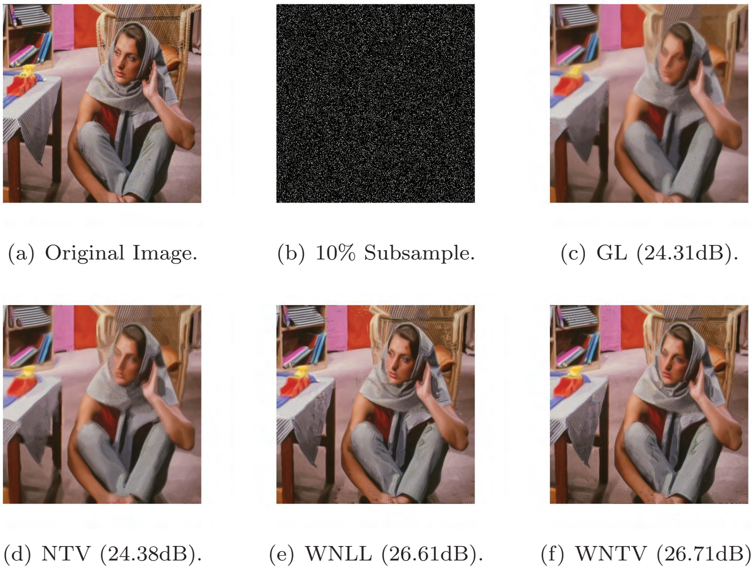


Fig. 6: *Evolutions of PSNR with number of iterations.*

where $\|\mathbf{x}-\mathbf{y}\|^2$ is the Euclidean distance between patches $\mathbf{x}$ and $\mathbf{y}$. $\sigma(\mathbf{x})$ is the distance between $\mathbf{x}$ and its 20th nearest neighbor. The weight $\omega(\mathbf{x},\mathbf{y})$ is made sparse by setting $\omega(\mathbf{x},\mathbf{y})$ equal to zero if point $\mathbf{y}$ is not among the 50th closest points to point $\mathbf{x}$. The color image is recovered in RGB channels separately.

We apply our algorithm to Fig. 7(a) and 8(a). Again, WNTV and WNLL outperform NTV and GL. In the image of house, in which cartoon dominates, the result of WNTV is better than WNLL. While in the image of Barbara, WNTV and WNLL are comparable since this image is rich in textures.

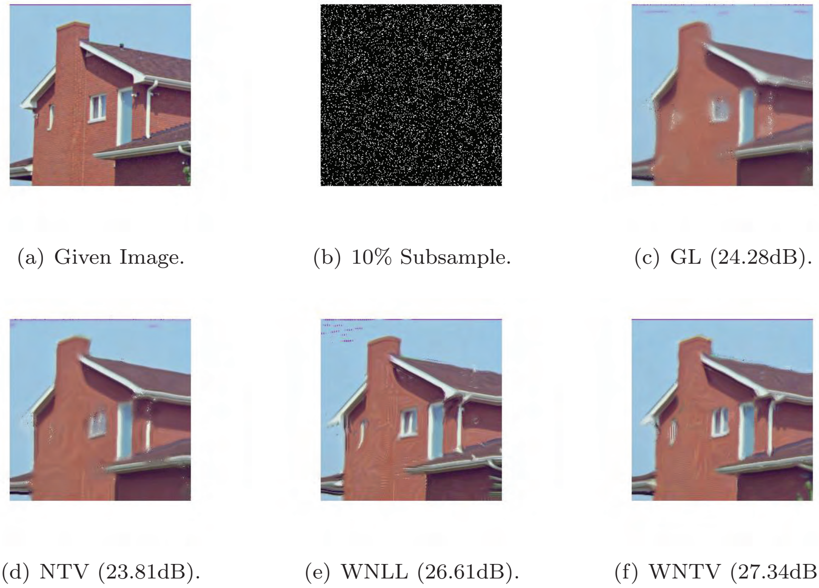
5. Image Colorization. Colorization is the process of adding color to monochrome images. It is usually done by person who is color expert but still this process is time consuming and sometimes could be boring. One way to reduce the



(a) Original Image. (b) 10% Subsample. (c) GL (24.31dB).

(d) NTV (24.38dB). (e) WNLL (26.61dB). (f) WNTV (26.71dB).

Fig. 7: *Results of Graph Laplacian (GL), nonlocal TV (NTV), weighted nonlocal Laplacian (WNLL) and weighted nonlocal TV (WNTV) in the color image of Barbara.*



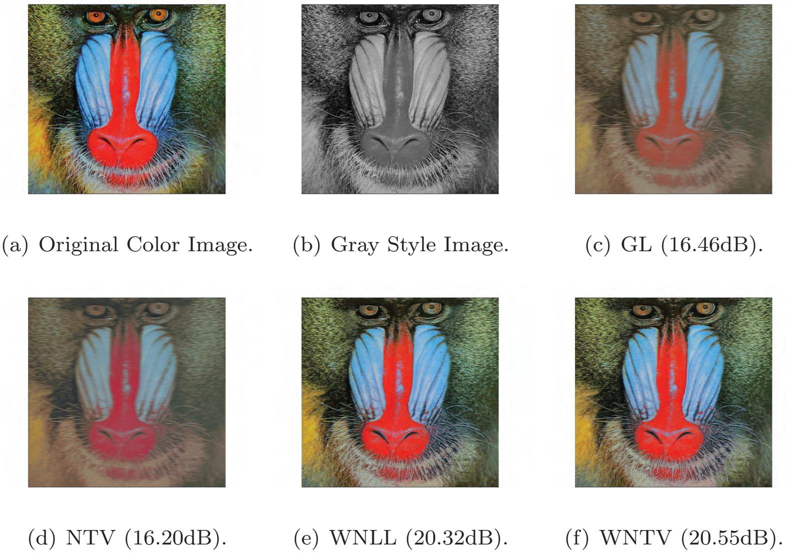
(a) Given Image. (b) 10% Subsample. (c) GL (24.28dB).

(d) NTV (23.81dB). (e) WNLL (26.61dB). (f) WNTV (27.34dB).

Fig. 8: *Results of Graph Laplacian, nonlocal TV, weighted Graph Laplacian and weighted nonlocal TV applied to color house image.*

working load is just adding color in part of the pixels artificially and applying some colorization method to extend the color to other pixels.

This problem can be natrually formulated as an interpolation on point cloud. The point cloud is constructed by taking patches from the gray image. On the patches, we have three functions, u_R, u_G and u_B corresponding to three channels of the color image. Then WNTV is used to interpolate u_R, u_G and u_B over the whole patch set. The weight is computed in the same way as that in image inpainting.



(a) Original Color Image. (b) Gray Style Image. (c) GL (16.46dB).

(d) NTV (16.20dB). (e) WNLL (20.32dB). (f) WNTV (20.55dB).

Fig. 9: *Results of Graph Laplacian (GL), nonlocal TV (NTV), weighted nonlocal Laplacian (WNLL) and weighted nonlocal TV (WNTV) in the baboon image colorization from 1% samples.*

The colorization results from 1% samples are demonstrated in Fig. 9, 10 and 11. Face of the baboon, snow mountains and wings of butterfly are not properly colored in GL and NTV. The face of baboon are blured. Part of the wings are colored in yellow by mistake. Snow mountain and text on F16 are also blured. In WNLL and WNTV, they are all properly colored. In addition, PSNR value also suggest that WNTV has the best performance.

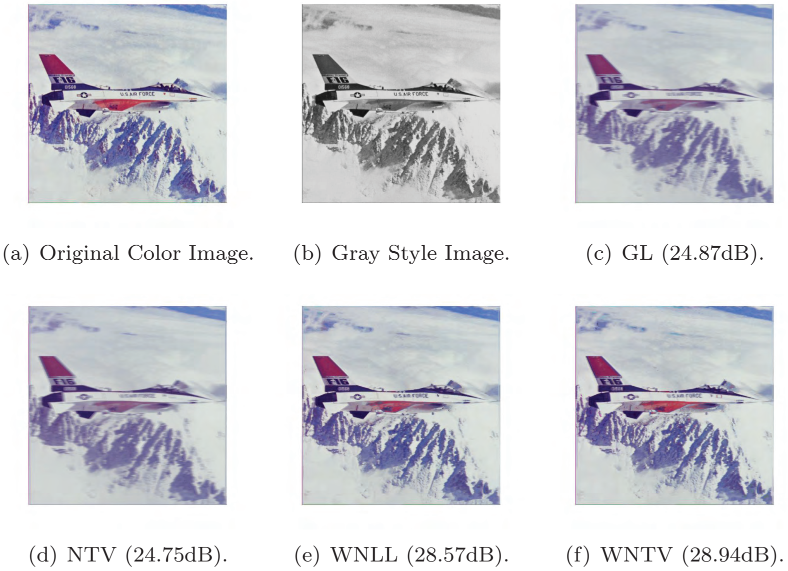
6. Conclusion. In this paper, we propose a weighted nonlocal total variation (WNTV) model for interpolations on high dimensional point cloud. This model can be solved efficiently by split Bregman iteration. Numerical tests in semi-supervised learning, image inpainting and image colorization demonstrate that WNTV is an effective and efficient method in image processing and data analysis.

REFERENCES

[1] A. Buades, B. Coll, and J.-M. Morel, *A review of image denoising algorithms, with a new one*, Multiscale Model. Simul., 4 (2005), pp. 490–530.

[2] A. Buades, B. Coll, and J.-M. Morel, *Neighborhood filters and PDE's*, Numer. Math., 105 (2006), pp. 1–34.

[3] F. R. K. Chung, *Spectral Graph Theory*, American Mathematical Society, 1997.



(a) Original Color Image.

(b) Gray Style Image.

(c) GL (24.87dB).

(d) NTV (24.75dB).

(e) WNLL (28.57dB).

(f) WNTV (28.94dB).

Fig. 10: *Results of Graph Laplacian (GL), nonlocal TV (NTV), weighted nonlocal Laplacian (WNLL) and weighted nonlocal TV (WNTV) in the F-16 image colorization from 1% samples.*

[4] G. Gilboa and S. Osher, *Nonlocal linear image regularization and supervised segmentation*, Multiscale Model. Simul., 6 (2007), pp. 595–630.

[5] G. Gilboa and S. Osher, *Nonlocal operators with applications to image processing*, Multiscale Model. Simul., 7 (2008), pp. 1005–1028.

[6] T. Goldstein and S. Osher, *The Split Bregman Method for L1-Regularized Problems*, SIAM Journal on Imaging Sciences, 2:2 (2009), pp. 323–343.

[7] Y. LeCun, C. Cortes, and C. J. Burges, Mnist database.

[8] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Modeling & Simulation, 4:2 (2005), pp. 460–489.

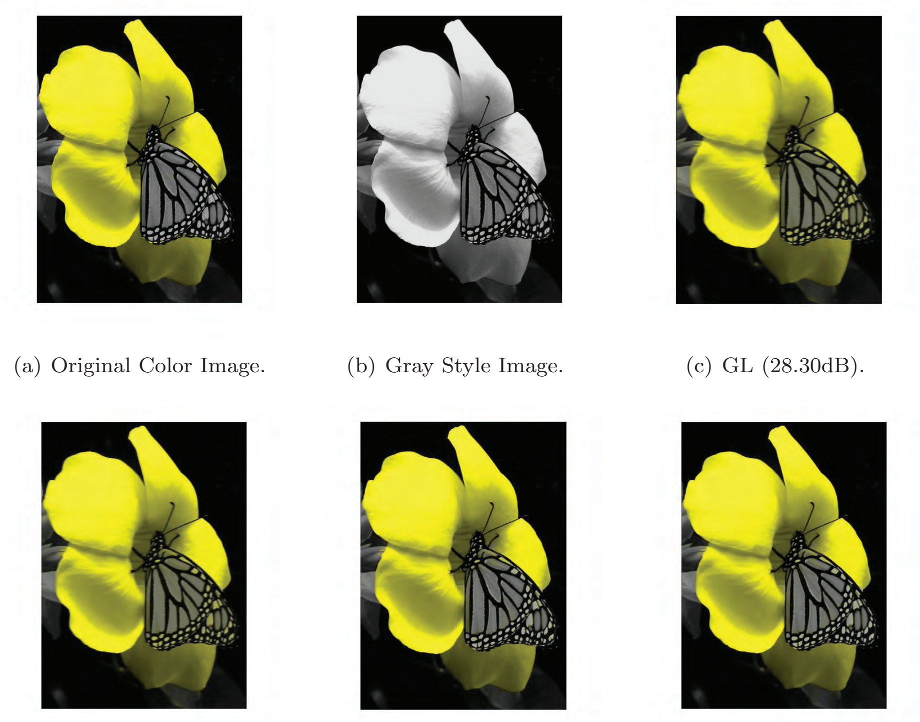
[9] S. Osher, Z. Shi, and W. Zhu, *Low dimensional manifold model for image processing*, SIAM Journal on Imaging Sciences, 10:4 (2017), pp. 1669–1690.

[10] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D: Nonlinear Phenomena, 60:1-4 (1992), pp. 259–268.

[11] Z. Shi, S. Osher, and W. Zhu, *Weighted nonlocal Laplacian on interpolation from sparse data*, Journal of Scientific Computing, 73 (2017), pp. 1164–1177.

[12] Z. Shi, J. Sun, and M. Tian, *Harmonic extension on the point cloud*, Multiscale Model. Simul., 16 (2018), pp. 215–247.

[13] X. Zhu, Z. Ghahramani, and J. D. Lafferty, *Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions*, in “Machine Learning, Proceedings of the Twentieth International Conference ICML 2003), August 21-24, 2003, Washington, DC, USA”, pp. 912–919, 2003.



(a) Original Color Image.

(b) Gray Style Image.

(c) GL (28.30dB).

(d) NTV (28.43dB).

(e) WNLL (31.98dB).

(f) WNTV (32.35dB).

Fig. 11: *Results of Graph Laplacian (GL), nonlocal TV (NTV), weighted nonlocal Laplacian (WNLL) and weighted nonlocal TV (WNTV) in the butterflyflower image colorization from 1% samples.*