Denoising Using Projection Onto Convex Sets (POCS) Based Framework

Mohammad Tofighi, Kivanc Kose*, A. Enis Cetin
Dept. of Electrical and Electronic Engineering, Bilkent University, Ankara, Turkey
*Dermatology Department, Memorial Sloan Kettering Cancer Center, New York, New York
tofighi@ee.bilkent.edu.tr, kosek@mskcc.org, cetin@bilkent.edu.tr

Abstract—Two new optimization techniques based on projections onto convex space (POCS) framework for solving convex optimization problems are presented. The dimension of the minimization problem is lifted by one and sets corresponding to the cost function are defined. If the cost function is a convex function in $\mathbb{R}^N$ the corresponding set is also a convex set in $\mathbb{R}^{N+1}$. The iterative optimization approach starts with an arbitrary initial estimate in $\mathbb{R}^{N+1}$ and an orthogonal projection is performed onto one of the sets in a sequential manner at each step of the optimization problem. The method provides globally optimal solutions in total-variation (TV), filtered variation (FV), $\ell_1$, and entropic cost functions. A new denoising algorithm using the TV framework is developed. The new algorithm does not require any of the regularization parameter adjustment. Simulation examples are presented.

Index Terms—Projection onto Convex Sets, Bregman Projections, Iterative Optimization, Lifting

I. INTRODUCTION

In many inverse signal and image processing problems and compressing sensing problems an optimization problem is solved to find a solution to the following problem:

$$\min_{w \in C} f(w),$$

(1)

where $C$ is a set in $\mathbb{R}^N$ and $f(w)$ is the cost function. Some commonly used cost functions are based on $\ell_1$, $\ell_2$, total-variation (TV), filtered variation, and entropic functions [1]–[5]. Bregman developed iterative methods based on the so-called Bregman distance to solve convex optimization problems which arise in signal and image processing [6]. In Bregman’s approach, it is necessary to perform a D-projection (or Bregman projection) at each step of the algorithm and it may not be easy to compute the Bregman distance in general [5], [7], [8].

In this article Bregman’s older projections onto convex sets (POCS) framework [9], [10] is used to solve convex and some non-convex optimization problems instead of the Bregman distance approach. Bregman’s POCS method has also been widely used for finding a common point of convex sets in many inverse signal and image processing problems [10]–[33]. In the ordinary POCS approach the goal is simply to find a vector which is in the intersection of convex sets. In each step of the iterative algorithm an orthogonal projection is performed onto one of the convex sets. Bregman showed that successive orthogonal projections converge to a vector which is in the intersection of all the convex sets. If the sets do not intersect iterates oscillate between members of the sets [34]–[36]. Since there is no need to compute the Bregman distance in standard POCS, it found applications in many practical problems.

In our approach the dimension of the minimization problem is lifted by one and sets corresponding to the cost function are defined. This approach is graphically illustrated in Fig. 1. If the cost function is a convex function in $\mathbb{R}^N$ the corresponding set is also a convex set in $\mathbb{R}^{N+1}$. As a result the convex minimization problem is reduced to finding a specific member (the optimal solution) of the set corresponding to the cost function. As in ordinary POCS approach the new iterative optimization method starts with an arbitrary initial estimate in $\mathbb{R}^{N+1}$ and an orthogonal projection is performed onto one of the sets. After this vector is calculated it is projected onto the other set. This process is continued in a sequential manner at each step of the optimization problem. The method provides globally optimal solutions in total-variation, filtered variation, $\ell_1$, and entropic function based cost functions because they are convex cost functions.

The article is organized as follows. In Section [II] the convex minimization method based on the POCS approach is introduced. In Section [III] a new denoising method based on the convex minimization approach introduced in Section [II] is presented. This new approach uses supporting hyperplanes of the TV function and it does not require a regularization parameter as in other TV based methods. Since it is very easy to perform an orthogonal projection onto a hyperplane this method is computationally implementable for many cost functions without solving any nonlinear equations. In Section [IV] we present the simulation results and some denoising examples.

II. CONVEX MINIMIZATION

Let us first consider a convex minimization problem

$$\min_{w \in \mathbb{R}^N} f(w),$$

(2)

where $f : \mathbb{R}^N \to \mathbb{R}$ is a convex function. We increase the dimension by one to define the following sets in $\mathbb{R}^{N+1}$ corresponding to the cost function $f(w)$ as follows:

$$C_f = \{w = [w^T \ y]^T : \ y \geq f(w)\},$$

(3)

which is the set of $N+1$ dimensional vectors whose $(N+1)^{st}$ component $y$ is greater than $f(w)$. This set $C_f$ is called the
epigraph of \( f \). We use bold face letters for \( N \) dimensional vectors and underlined bold face letters for \( N+1 \) dimensional vectors, respectively.

The second set is that is related with the cost function \( f(\mathbf{w}) \) is the level set:

\[
C_s = \{ \mathbf{w} = [\mathbf{w}^T y]^T : y \leq \alpha, \mathbf{w} \in \mathbb{R}^{N+1} \},
\]

where \( \alpha \) is a real number. Here it is assumed that \( f(\mathbf{w}) \geq \alpha \) for all \( f(\mathbf{w}) \in \mathbb{R} \) such that the sets \( C_f \) and \( C_s \) do not intersect. They are both closed and convex sets in \( \mathbb{R}^{N+1} \). Sets \( C_f \) and \( C_s \) are graphically illustrated in Fig. 1 in which \( \alpha = 0 \).

![Fig. 1. Two convex sets \( C_f \) and \( C_s \) corresponding to the cost function \( f \). We sequentially project an initial vector \( \mathbf{w}_0 \) onto \( C_s \) and \( C_f \) to find the global minimum which is located at \( \mathbf{w}^* \).](image)

The POCS based minimization algorithm starts with an arbitrary \( \mathbf{w}_0 = [\mathbf{w}_0^T y_0]^T \in \mathbb{R}^{N+1} \). We project \( \mathbf{w}_0 \) onto the set \( C_s \) to obtain the first iterate \( \mathbf{w}_1 \) which will be,

\[
\mathbf{w}_1 = [\mathbf{w}_0^T 0]^T,
\]

where \( \alpha = 0 \) is assumed as in Fig. 1. Then we project \( \mathbf{w}_1 \) onto the set \( C_f \). The new iterate \( \mathbf{w}_2 \) is determined by minimizing the distance between \( \mathbf{w}_1 \) and \( C_f \), i.e.,

\[
\mathbf{w}_2 = \arg \min_{\mathbf{w} \in C_f} \| \mathbf{w}_1 - \mathbf{w} \|.
\]

Equation 6 is the ordinary orthogonal projection operation onto the set \( C_f \in \mathbb{R}^{N+1} \). To solve the problem in Eq. 6 we do not need to compute the Bregman’s so-called D-projection. After finding \( \mathbf{w}_2 \), we perform the next projection onto the set \( C_s \) and obtain \( \mathbf{w}_3 \) etc. Eventually iterates oscillate between two nearest vectors of the two sets \( C_s \) and \( C_f \). As a result we obtain

\[
\lim_{n \to \infty} \mathbf{w}_{2n} = [\mathbf{w}^* f(\mathbf{w}^*)]^T,
\]

where \( \mathbf{w}^* \) is the \( N \) dimensional vector minimizing \( f(\mathbf{w}) \). The proof of Eq. 7 follows from Bregman’s POCS theorem [7], [4]. It was generalized to non-intersection case by Gubin et. al [12], [34], [35]. Since the two closed and convex sets \( C_s \) and \( C_f \) are closest to each other at the optimal solution case, iterations oscillate between the vectors \( [\mathbf{w}^* f(\mathbf{w}^*)]^T \) and \( [\mathbf{w}^* 0]^T \) in \( \mathbb{R}^{N+1} \) as \( n \) tends to infinity. It is possible to increase the speed of convergence by non-orthogonal projections [24].

If the cost function \( f \) is not convex and have more than one local minimum then the corresponding set \( C_f \) is not convex in \( \mathbb{R}^{N+1} \). In this case iterates may converge to one of the local minima.

### III. Denoising Using POCS

In this section, we present a new method of denoising, based on TV and FV. Let the noisy signal be \( \mathbf{y} \), and the original signal or image be \( \mathbf{w}_0 \). Suppose that the observation model is the additive noise model:

\[
\mathbf{y} = \mathbf{w}_0 + \mathbf{v},
\]

where \( \mathbf{v} \) is the additional noise. In this approach we solve the following problem for denoising:

\[
\mathbf{w}^* = \arg \min_{\mathbf{w} \in C_f} \| \mathbf{y} - \mathbf{w} \|^2,
\]

where, \( \mathbf{y} = [y^T 0] \) and \( C_f \) is the epigraph set of TV or FV in \( \mathbb{R}^{N+1} \). The minimization problem is essentially the orthogonal projection onto the set \( C_f \). This means that we select the nearest vector \( \mathbf{w}^* \) on the set \( C_f \) to \( \mathbf{y} \). This is graphically illustrated in Fig. 2.

![Fig. 2. Graphical representation of the minimization of Eq. 9. \( \mathbf{y} \) is projected onto the set \( C_f \). TV(\( \mathbf{w} \)) is zero for \( \mathbf{w} = [0, 0, \ldots, 0]^T \) or when it is a constant vector.](image)

In current TV based denoising methods [37], [38] the following cost function is used:

\[
\min_{\mathbf{w} \in C_f} \| \mathbf{y} - \mathbf{w} \|^2 + \lambda \text{TV}(\mathbf{w}).
\]

The solution of this problem can be obtained using the method that we discussed in Section II. One problem with this approach is the estimation of the regularization parameter \( \lambda \). One has to determine the \( \lambda \) in an ad hoc manner or by visual inspection. On the other hand we do not require any parameter adjustment in Eq. 9.

The denoising solution in Eq. 9 can be found by performing successive orthogonal projection onto supporting hyperplanes of the epigraph set \( C_f \). In the first step we calculated TV(\( \mathbf{y} \)). We also calculate the surface normal at \( \mathbf{y} = [y^T \ TV(\mathbf{y})] \) in \( \mathbb{R}^{N+1} \) and determine the equation of the supporting hyperplane at \( [y^T \ TV(\mathbf{y})] \). We project \( \mathbf{y} = [y^T 0] \) onto this
Fig. 3. Graphical representation of the minimization of (9), using projection onto the supporting hyperplanes of $C_f$.

In the second step we project $\mathbf{w}_1$ onto the set $C_s$ by simply making its last component zero. We calculate the TV of this vector and the surface normal, and the supporting hyperplane as in the previous step. We project $\mathbf{y}$ onto the new supporting hyperplane, etc.

The sequence of iterations obtained in this manner converges to a vector in the intersection of $C_s$ and $C_f$. In this problem the sets $C_s$ and $C_f$ intersect because $TV(\mathbf{w}) = 0$ for $\mathbf{w} = [0, 0, ..., 0]^T$ or for a constant vector. However, we do not want to find a trivial constant vector in the intersection of $C_s$ and $C_f$. We calculate the distance between $\mathbf{y}$ and $\mathbf{w}_i$ at each step of the iterative algorithm described in the previous paragraph. This distance $\|\mathbf{y} - \mathbf{w}_i\|^2_2$ initially decreases and starts increasing as $i$ increases. Once we detect the increase we perform some refinement projections to obtain the solution of the denoising problem. A typical convergence graph is shown in Fig. 4 for the “note” image. Simulation examples are presented in the next section.

IV. SIMULATION RESULTS

Consider the “Note” image shown in Fig. 5. This is corrupted by a zero mean Gaussian noise with $\lambda = 45$ in Fig. 6. The image is restored using our method and Chambolle’s algorithm [37] and the denoised images are shown in Fig. 7 and 8 respectively. The $\lambda$ parameter in [10] is manually adjusted to get the best possible results. Our algorithm not only produce a higher SNR, but also a visually better looking image. Solution results for other SNR levels are presented in Table I. We also corrupted this image with $\epsilon$-contaminated Gaussian noise (“salt-and-pepper noise”). Denoising results are summarized in Table II.

In Table III denoising results for 10 other images with different noise levels are presented. In almost all cases our method produces higher SNR results than the denoising results obtained using [37].

| Noise std | Input SNR | POCS | Chambolle |
|-----------|-----------|------|-----------|
| 5         | 21.12     | 30.63| 29.48     |
| 10        | 15.12     | 25.93| 24.20     |
| 15        | 11.56     | 22.91| 21.05     |
| 20        | 9.06      | 20.93| 18.90     |
| 25        | 7.14      | 19.27| 17.17     |
| 30        | 5.59      | 17.89| 15.78     |
| 35        | 4.21      | 16.68| 14.69     |
| 40        | 3.07      | 15.90| 13.70     |
| 45        | 2.05      | 15.08| 12.78     |
| 50        | 1.12      | 14.25| 12.25     |

V. CONCLUSION

A new denoising method based on the epigraph of the TV function is developed. The solution is obtained using POCS. The new algorithm does not need the optimization of the regularization parameter.
TABLE II
COMPARISON OF THE RESULTS FOR DENOISING ALGORITHMS FOR \( \epsilon \)-CONTAMINATION NOISE FOR NOTE IMAGE

| \( \epsilon \) | \( \sigma_1 \) | \( \sigma_2 \) | Input SNR | POCS | Chambolle |
|--------------|--------------|--------------|-----------|-------|-----------|
| 0.9          | 5            | 30           | 14.64     | 23.44 | 20.56     |
| 0.9          | 5            | 40           | 12.55     | 21.39 | 17.60     |
| 0.9          | 5            | 50           | 10.75     | 19.49 | 15.54     |
| 0.9          | 5            | 60           | 9.29      | 17.61 | 13.82     |
| 0.9          | 5            | 70           | 7.98      | 16.01 | 12.57     |
| 0.9          | 5            | 80           | 6.89      | 14.54 | 11.37     |
| 0.9          | 10           | 30           | 12.56     | 22.88 | 19.74     |
| 0.9          | 10           | 40           | 11.13     | 21.00 | 15.30     |
| 0.9          | 10           | 50           | 9.85      | 19.35 | 12.47     |
| 0.9          | 10           | 60           | 8.58      | 17.87 | 10.42     |
| 0.9          | 10           | 70           | 7.52      | 16.38 | 8.76      |
| 0.9          | 10           | 80           | 6.46      | 15.05 | 7.45      |
| 0.95         | 5            | 30           | 16.75     | 24.52 | 23.18     |
| 0.95         | 5            | 40           | 14.98     | 22.59 | 20.44     |
| 0.95         | 5            | 50           | 13.41     | 20.54 | 18.45     |
| 0.95         | 5            | 60           | 12.10     | 18.72 | 16.80     |
| 0.95         | 5            | 70           | 10.80     | 17.13 | 15.34     |
| 0.95         | 5            | 80           | 9.76      | 15.63 | 14.11     |
| 0.95         | 10           | 30           | 13.68     | 23.79 | 20.43     |
| 0.95         | 10           | 40           | 12.66     | 22.09 | 15.35     |
| 0.95         | 10           | 50           | 11.71     | 20.65 | 12.28     |
| 0.95         | 10           | 60           | 10.72     | 19.10 | 10.22     |
| 0.95         | 10           | 70           | 9.82      | 17.59 | 8.66      |
| 0.95         | 10           | 80           | 8.92      | 16.12 | 7.34      |

Fig. 5. Original “Note” image.

Fig. 6. “Note” image corrupted with Gaussian noise with \( \lambda = 45 \).

Fig. 7. Denoised image “Note” image, using \([1 -1]\) filter; SNR = 15.08.
level will there is insufficient should be made in a

Fig. 8. Denoised image "Note" image, using Chambolle’s algorithm; SNR = 12.78.

TABLE III

| Images      | Noise std | Input SNR | POCS       | Chambolle  |
|-------------|-----------|-----------|------------|------------|
| House       | 30        | 13.85     | 27.43      | 27.13      |
| House       | 50        | 9.45      | 24.20      | 24.36      |
| Lena        | 30        | 12.95     | 23.63      | 23.54      |
| Lena        | 50        | 8.50      | 21.46      | 21.37      |
| Mandrill    | 30        | 13.04     | 19.98      | 19.64      |
| Mandrill    | 50        | 8.61      | 17.94      | 17.92      |
| Living room | 30        | 12.65     | 21.21      | 20.88      |
| Living room | 50        | 8.20      | 19.25      | 19.05      |
| Lake        | 30        | 13.44     | 22.19      | 21.86      |
| Lake        | 50        | 8.97      | 20.03      | 19.90      |
| Jet plane   | 30        | 15.57     | 26.28      | 25.91      |
| Jet plane   | 50        | 11.33     | 23.91      | 23.54      |
| Peppers     | 30        | 12.65     | 23.57      | 23.59      |
| Peppers     | 50        | 8.20      | 21.48      | 21.36      |
| Pirate      | 30        | 12.13     | 21.39      | 21.30      |
| Pirate      | 50        | 7.71      | 19.37      | 19.43      |
| Cameraman   | 30        | 12.97     | 24.13      | 23.67      |
| Cameraman   | 50        | 8.55      | 21.55      | 21.22      |
| Flower      | 30        | 11.84     | 21.97      | 20.89      |
| Flower      | 50        | 7.42      | 19.00      | 18.88      |
| Average     | 30        | 13.11     | 23.18      | 22.84      |
| Average     | 50        | 8.69      | 20.82      | 20.70      |

Fig. 9. Sample images used in our experiments (a) House, (b) Jet plane, (c) Lake, (d) Lena, (e) Living room, (f) Mandrill, (g) Peppers, (h) Pirate, (i) Flower, (j) Cameraman.
