Accelerated Proximal Gradient Method for "Garbage Bin" Optimization Problem

Shijie Zhou

November 25, 2020

In this note we are trying to develop an efficient algorithm to solve the "garbage bin" optimization problem in SECM. This algorithm is based on accelerated proximal gradient method. Here we will introduce the solver step by step, illustrate the simulation experiments on synthetic data, and also discuss the things to be improved in practical use.

1 Introduction

So far we have developed the algorithm to handle multi-motif shapes which are not rotational invariant, like triangles. Now we are prototyping a more flexible algorithm to handle various feature shapes and are able to correctly reconstruct arbitrary images. The arbitrary image can be represented as the convolution sum of structured motifs dictionary $\boldsymbol{D}$ and sparse maps $\boldsymbol{X}$, plus the image for capturing features that are hard to model explicitly, as is shown in Figure 1.

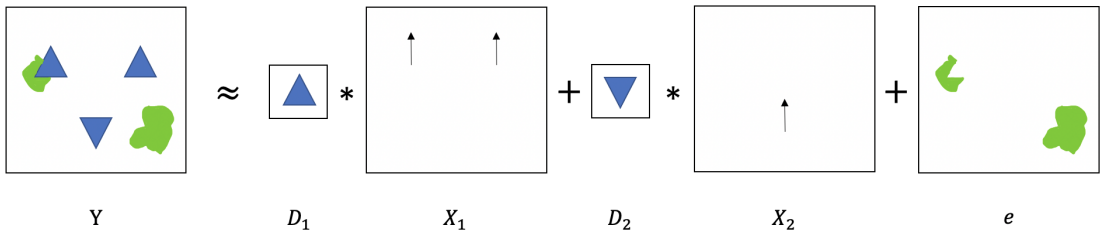

Figure 1: Sparse Representation for arbitrary images

We are incorporating extra terms of samll L1 norm and small total variation which capture the structured noise $\boldsymbol{e}$. We assume this structured noise has gradient sparsity property so the total variation (TV) term is suitable for this problem formulation. So *what is the structured noise* $\boldsymbol{e}$ *in SECM and why it has gradient sparsity property*?

Basically, the structured noise $\boldsymbol{e}$ can be regarded as a kind of "garbage bin", it could pick up everything that motifs didn't capture, just like the blobby thing in SECM (a blob of the reactive substance). If it is a few pixels that are different, then a sparse model can be appropriate. But if it is a blob, then the main property of the blob is "continuous", so its boundary is small comparing to its area. If we treat the blob as sparse, we would pay the area. If we treat the blob as having small total variation, then we pay the boundary, that is the number of pixels where the gradient is nonzero. Therefore, if we think the residual in SECM which is hard to model explicitly is blobby, then the total variation is appropriate.

2 Problem formulation

We are interested in the following optimization problem:

$$\min_{x,e} \frac{1}{2}\|\mathcal{L}[D[x]+e]-R\|_2^2+\alpha\|x\|_1+\gamma\|e\|_{TV} \tag{2.1}$$

$$=\min_{x,e} \frac{1}{2}\|\mathcal{L}[D[x]+e]-R\|_2^2+\alpha\|x\|_1+\gamma\sum_i\|\nabla e_i\|_2 \tag{2.2}$$

$$=\min_{x,e} \frac{1}{2}\|\mathcal{L}[D[x]]+\mathcal{L}[e]-R\|_2^2+\alpha\|x\|_1+\gamma\|\nabla e\|_2 \tag{2.3}$$

where $x \in \mathbb{R}^{n^2}$ is the vectorized sparse map (to make the notation simple, here we use $D[x]$ to represent the convolution sum of the dictionary and sparse maps), $e \in \mathbb{R}^{n^2}$ is the vectorized "garbage bin" image, $\nabla e = \begin{bmatrix} \nabla_x e \\ \nabla_y e \end{bmatrix} \in \mathbb{R}^{2n^2}$ is the discrete gradient of the "garbage bin" image, and $\nabla e_i = \begin{bmatrix} \nabla_x e_i \\ \nabla_y e_i \end{bmatrix} \in \mathbb{R}^2$ denotes the discrete gradient of e at pixel i.

To be more precise, the formal ideal optimization problem in our SECM setting can be written as follow:

$$\boldsymbol{X},\boldsymbol{e}=\underset{\boldsymbol{X_1},\ldots,\boldsymbol{X_K},\boldsymbol{e}}{\operatorname{argmin}} \frac{1}{2}||\mathcal{S}\{\Psi*\mathcal{L}_\Theta[\sum_{i=1}^{K}\boldsymbol{D}_i*\boldsymbol{X}_i+\boldsymbol{e}]\}-\boldsymbol{R}||_2^2+\alpha\sum_{i=1}^{K}\|\boldsymbol{X}_i\|_1+\gamma\|\boldsymbol{e}\|_{TV}, \tag{2.4}$$

3 Auxiliary variable and penalty techniques

An auxiliary variable $v \in \mathbb{R}^{2n^2}$ is introduced to transfer ∇e out of the nondifferentiable term $\|\cdot\|_2$, and the difference between v and ∇e is penalized[2][3]:

$$\min_{x,e,v} \frac{1}{2}\|\mathcal{L}[D[x]]+\mathcal{L}[e]-R\|_2^2+\alpha\|x\|_1+\gamma\|v\|_2 \tag{3.1}$$

$$\text{s.t.} \quad v=\nabla e \in \mathbb{R}^{2n^2}$$

$\Leftrightarrow$

$$\min_{x,e,v} \frac{1}{2}\|\mathcal{L}[D[x]]+\mathcal{L}[e]-R\|_2^2+\frac{\beta}{2}\|v-\nabla e\|_2^2+\alpha\|x\|_1+\gamma\|v\|_2 \tag{3.2}$$

with a sufficiently large penalty parameter β.

Set variable $w=(x,e,v)$, and rewrite the linear operator $\mathcal{L}[D[\cdot]]$ as $\mathcal{A}[\cdot]$, (3.2) becomes:

$$\min_{(x,e,v)} \underbrace{\frac{1}{2}\|\mathcal{A}[x]+\mathcal{L}[e]-R\|_2^2+\frac{\beta}{2}\|v-\nabla e\|_2^2}_{f(w)}+\underbrace{\alpha\|x\|_1+\gamma\|v\|_2}_{g(w)} \tag{3.3}$$

$$=\min_w f(w)+g(w) \tag{3.4}$$

where

$f(w)=\frac{1}{2}\|\mathcal{A}[x]+\mathcal{L}[e]-R\|_2^2+\frac{\beta}{2}\|v-\nabla e\|_2^2$, which is convex and smooth.

$g(w)=\alpha\|x\|_1+\gamma\|v\|_2$, which is convex and non-smooth.

So now we can apply the proximal gradient method to (3.4) without alternating the variables.

4 Proximal gradient method

To make the derivation simpler and clearer, let's change the notation of the linear maps to operators, that is: $\mathcal{A}[x] = \mathcal{A}x$ and $\mathcal{A}^*$ is the adjoint of $\mathcal{A}$, $\mathcal{L}[e] = \mathcal{L}e$ and $\mathcal{L}^*$ is the adjoint of $\mathcal{L}$. Additionally, ∇^* is the adjoint of the dicrete derivative operator ∇.

$$\min_{(x,e,v)} \underbrace{\frac{1}{2}\|\mathcal{A}x + \mathcal{L}e - R\|_2^2 + \frac{\beta}{2}\|v - \nabla e\|_2^2}_{f(w)} + \underbrace{\alpha\|x\|_1 + \gamma\|v\|_2}_{g(w)} \tag{4.1}$$

$$= \min_w f(w) + g(w) \tag{4.2}$$

Now let's apply the proximal gradient method:
Step 1:

$$w_k' = w_k - \frac{1}{L}\nabla f(w) \tag{4.3}$$

$$\Rightarrow \begin{cases} x_k' &= x_k - \frac{1}{L_x}\nabla_x f(w) \\ v_k' &= v_k - \frac{1}{L_v}\nabla_v f(w) \\ e_k' &= e_k - \frac{1}{L_e}\nabla_e f(w) \end{cases} \tag{4.4}$$

Let's calculate the gradients first:

$$\nabla_x f(w) = \frac{\partial}{\partial x}\frac{1}{2}\|\mathcal{A}x - (R - \mathcal{L}e)\|_2^2 \tag{4.5}$$

$$= \mathcal{A}^*[\mathcal{A}x - (R - \mathcal{L}e)] \tag{4.6}$$

$$= \mathcal{A}^*\mathcal{A}x - \mathcal{A}^*(R - \mathcal{L}e) \tag{4.7}$$

$$\nabla_v f(w) = \frac{\partial}{\partial v}\frac{\beta}{2}\|v - \nabla e\|_2^2 \tag{4.8}$$

$$= \beta(v - \nabla e) \tag{4.9}$$

$$\nabla_e f(w) = \frac{\partial}{\partial e}\frac{1}{2}\|\mathcal{L}e - (R - \mathcal{A}x)\|_2^2 + \frac{\beta}{2}\|v - \nabla e\|_2^2 \tag{4.10}$$

$$= \mathcal{L}^*[\mathcal{L}e - (R - \mathcal{A}x)] - \beta\nabla^*(v - \nabla e) \tag{4.11}$$

$$= \mathcal{L}^*\mathcal{L}e - \mathcal{L}^*(R - \mathcal{A}x) - \beta\nabla^* v + \beta\nabla^*\nabla e \tag{4.12}$$

$$= (\mathcal{L}^*\mathcal{L} + \beta\nabla^*\nabla)e - \mathcal{L}^*(R - \mathcal{A}x) - \beta\nabla^* v \tag{4.13}$$

Then by the definition of the Lipschitz constant, we can easily find out three Lipschitz constants:

$$L_x = \|\mathcal{A}^*\mathcal{A}\| \tag{4.14}$$

$$L_v = \beta \tag{4.15}$$

$$L_e = \|\mathcal{L}^*\mathcal{L} + \beta\nabla^*\nabla\| \tag{4.16}$$

However, it is difficult to calculate the Lipschitz constant L_x and L_e in practice because of too much computational cost. We will discuss how to approximate it and choose the step size later.

Ideally, now we can finish the first step of the proximal gradient method using the above results:

$$\begin{cases} x'_k &= x_k - \frac{1}{\|\mathcal{A}^*\mathcal{A}\|}[\mathcal{A}^*\mathcal{A}x_k - \mathcal{A}^*(R - \mathcal{L}e_k)] \\ v'_k &= v_k - \frac{1}{\beta}[\beta(v_k - \nabla e_k)] = \nabla e_k \\ e'_k &= e_k - \frac{1}{\|\mathcal{L}^*\mathcal{L} + \beta\nabla^*\nabla\|}[(\mathcal{L}^*\mathcal{L} + \beta\nabla^*\nabla)e_k - \mathcal{L}^*(R - \mathcal{A}x_k) - \beta\nabla^* v_k] \end{cases} \tag{4.17}$$

Step 2:

$$w_{k+1} = prox(w'_k) = \underset{w}{\operatorname{argmin}} \frac{L}{2}\|w - w'_k\|_F^2 + g(w) \tag{4.18}$$

$$= \underset{x,v,e}{\operatorname{argmin}} \frac{L_x}{2}\|x - x'_k\|_2^2 + \frac{L_v}{2}\|v - v'_k\|_2^2 + \frac{L_e}{2}\|e - e'_k\|_2^2 + \alpha\|x\|_1 + \gamma\|v\|_2 \tag{4.19}$$

which is equivalent to:

$$x_{k+1} = \underset{x}{\operatorname{argmin}} \frac{L_x}{2}\|x - x'_k\|_2^2 + \alpha\|x\|_1 \tag{4.20}$$

$$v_{k+1} = \underset{v}{\operatorname{argmin}} \frac{L_v}{2}\|v - v'_k\|_2^2 + \gamma\|v\|_2 \tag{4.21}$$

$$e_{k+1} = \underset{e}{\operatorname{argmin}} \frac{L_e}{2}\|e - e'_k\|_2^2 \tag{4.22}$$

The solution will be:

$$x_{k+1} = soft(x'_k, \frac{\alpha}{L_x}) = soft(x'_k, \frac{\alpha}{\|\mathcal{A}^*\mathcal{A}\|}) \tag{4.23}$$

$$v_{k+1} = shrink(v'_k, \frac{\gamma}{L_v}) = shrink(v'_k, \frac{\gamma}{\beta}) \tag{4.24}$$

$$e_{k+1} = e'_k \tag{4.25}$$

where $soft(\cdot,\cdot)$ is the soft-thresholding operator

$$soft(x,\tau) \doteq \max\{|x| - \tau, 0\} \cdot \operatorname{sgn}(x), \quad x \in \mathbb{R}$$

applied to the vector entry-wise.
And $shrink(\cdot,\cdot)$ is the 2D shrinkage operator

$$shrink(\boldsymbol{x},\tau) \doteq \max\{\|\boldsymbol{x}\| - \tau, 0\} \cdot \boldsymbol{x}/\|\boldsymbol{x}\|, \quad \boldsymbol{x} \in \mathbb{R}^{2n^2}$$

5 Accelerated proximal gradient method

Now we organize our algorithm in Accelerated Proximal Gradient (APG) scheme:

Algorithm 1 Accelerated Proximal Gradient (APG) for Garbage Bin Problem

Problem: $\min_{(x,e,v)} \underbrace{\frac{1}{2}\|\mathcal{A}x + \mathcal{L}e - R\|_2^2 + \frac{\beta}{2}\|v - \nabla e\|_2^2}_{f(w)} + \underbrace{\alpha\|x\|_1 + \gamma\|v\|_2}_{g(w)} = \min_w f(w) + g(w)$

Input: $x_0 \in \mathbb{R}^{n^2}$, $v_0 \in \mathbb{R}^{2n^2}$, $e_0 \in \mathbb{R}^{n^2}$, $p_1 \leftarrow x_0$, $q_1 \leftarrow v_0$, $r_1 \leftarrow e_0$, $t_1 \leftarrow 1$, step sizes: ρ_x, ρ_v, ρ_e

1: **while** x_k, v_k, e_k not converged $(k = 1, 2, ...)$ **do**

2:
$$\begin{cases} x'_k & \leftarrow p_k - \rho_x \nabla_{p_k} f(w) & = p_k - \rho_x \mathcal{A}^*[\mathcal{A}p_k - (R - \mathcal{L}e)] \\ v'_k & \leftarrow q_k - \rho_v \nabla_{q_k} f(w) & = q_k - \rho_v \beta(q_k - \nabla e) \\ e'_k & \leftarrow r_k - \rho_e \nabla_{r_k} f(w) & = r_k - \rho_e \mathcal{L}^*[\mathcal{L}r_k - (R - \mathcal{A}x)] - \beta\nabla^*(v - \nabla r_k) \end{cases}$$

3:
$$\begin{cases} x_{k+1} & \leftarrow & soft(x'_k, \alpha\rho_x) \\ v_{k+1} & \leftarrow & shrink(v'_k, \frac{\gamma}{\beta}) \\ e_{k+1} & \leftarrow & e'_k \end{cases}$$

4: $t_{k+1} \leftarrow \frac{1+\sqrt{1+4t_k^2}}{2}$, $b_{k+1} \leftarrow \frac{t_k - 1}{t_{k+1}}$

5:
$$\begin{cases} p_{k+1} & \leftarrow & x_{k+1} + b_{k+1}(x_{k+1} - x_k) \\ q_{k+1} & \leftarrow & v_{k+1} + b_{k+1}(v_{k+1} - v_k) \\ r_{k+1} & \leftarrow & e_{k+1} + b_{k+1}(e_{k+1} - e_k) \end{cases}$$

6: **end while**

Output: $x_* \leftarrow x_k$, $v_* \leftarrow v_k$, $e_* \leftarrow e_k$

The comparison experiments using MATLAB show that the convergence of APG is much better than that of PG.

6 Simulation experiments

We developed a package in MATLAB and finished the simulation experiments using synthetic data. As is shown in Figure.2, we generate a random sparse map and do the convolution with triangle dictionary to get the true triangles image. As we stated in the previous section, the purpose of this basic version of simulation experiment is to verify the feasibility of our proposed model and algorithm. So here we use the dictionary of triangle with one orientation instead of the multi-motif array. The true blob image is created by loading a self-designed matrix, an .mat file in MATLAB. The ground truth image is the combination of both true triangle image and true blob image.

To make computation more efficient, every linear operator in our problem is defined as a matrix in our code, so its adjoint is simply the matrix's conjugate transpose. To be specific, the linear operator $\mathcal{L}[D[\cdot]] = \mathcal{A}[\cdot]$ is defined as a random matrix $\boldsymbol{A}$, the line scan operator is defined as another random matrix $\boldsymbol{L}$, the dicrete derivative operator ∇ is defined as a first-order finite difference matrix $\nabla \in \mathbb{R}^{2n^2 \times n^2}$, which can directly work with the vectoried garbage bin (blob) image $e \in \mathbb{R}^{n^2}$ and generate $\nabla e = \begin{bmatrix} \nabla_x e \\ \nabla_y e \end{bmatrix} \in \mathbb{R}^{2n^2}$. We utilize only 5% of the measurement to do the reconstruction.

For the parameter choices, $\alpha = 100$, $\beta = 5000$, $\gamma = 2e + 10$, and the step sizes $\rho_x = 1e - 5$, $\rho_v = 1/\beta$, $\rho_e = 1e - 6$. Since the Lipschitz constants L_x and L_e are difficult to compute in practice according to its definition, these choices can be tricky but the combination of our choices here can result in a relative good reconstruction result, as is shown in Figure 3.

We will discuss about the parameters selection in the next section.

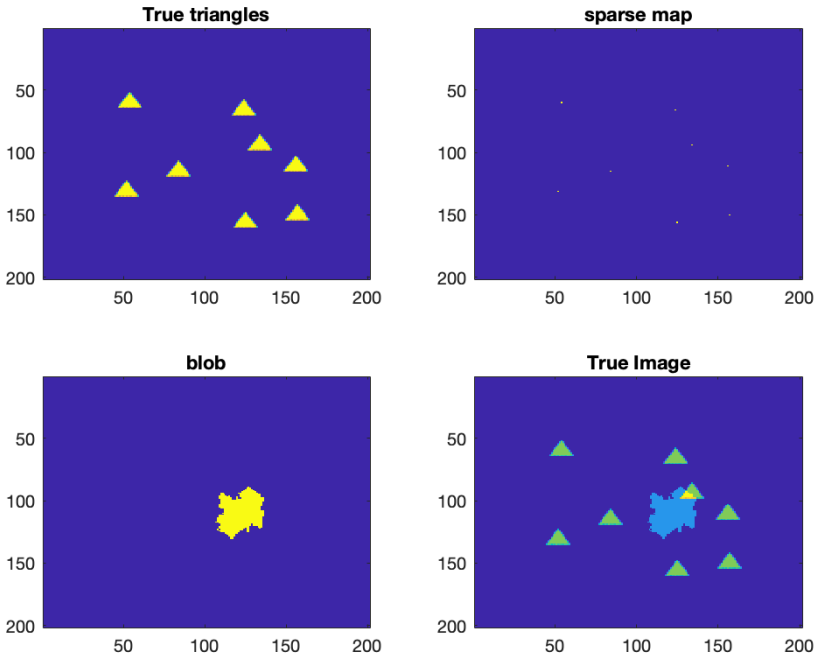

Figure 2: Synthetic data (ground truth)

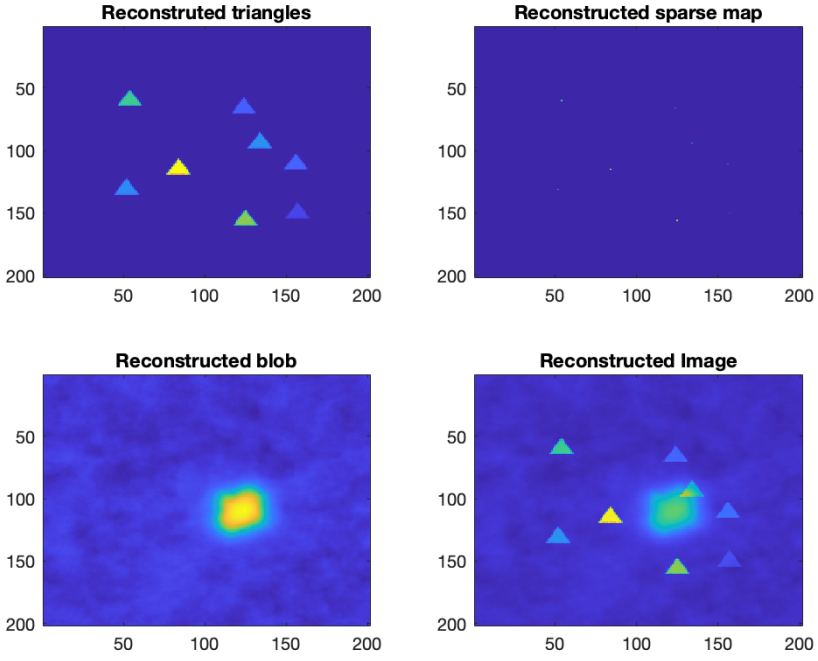

Figure 3: Reconstruction results

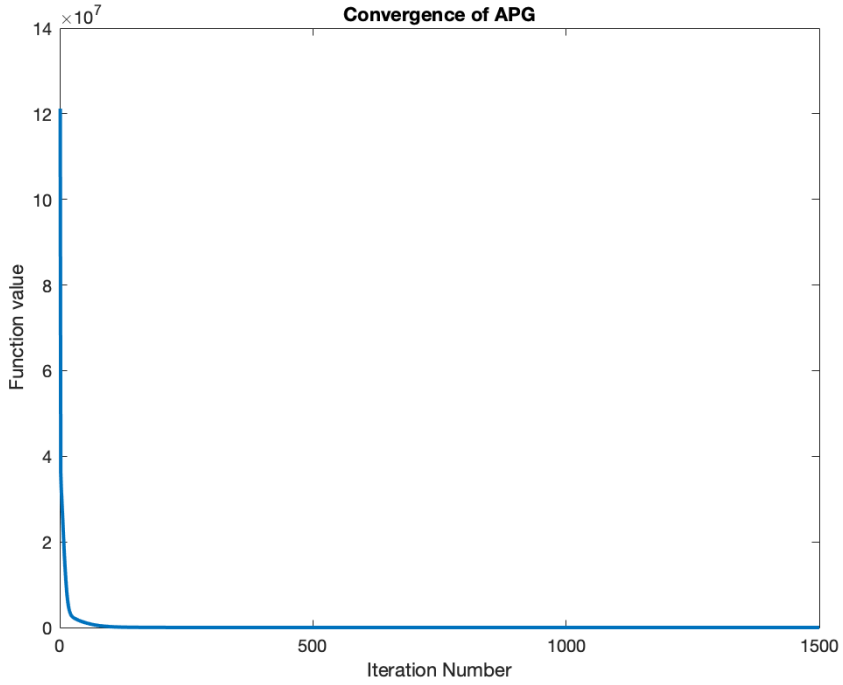

Figure 4: Objective function value

7 Discussion

Although our reconstruction results are relatively good, they are still not perfect. Based on my experience of dealing with real data, the choices for parameters can be a serious issue and generally a fixed constant will not work so good. As it is stated in the previous sections, the choices for step size and penalty parameters can be both serious issues. Here we propose two ideas that may be helpful.

7.1 Step size choice: One-step line search + Barzilai-Borwein (BB) method

To avoid calculating the Lipschitz constants and using fixed step size, choosing step sizes by one-step line search + Barzilai-Borwein (BB) method may make things better. Recall the first step of proximal gradient method, where step sizes $\frac{1}{L_x}$ and $\frac{1}{L_e}$ are hard to compute directly:

$$\begin{cases} x'_k &= x_k - \frac{1}{L_x}\nabla_x f(w) \\ e'_k &= e_k - \frac{1}{L_e}\nabla_e f(w) \end{cases} \tag{7.1}$$

Let's rewrite them using clearer notations:

$$\begin{cases} x'_k &= x_k - \rho_x \nabla f(x) \\ e'_k &= e_k - \rho_e \nabla f(e) \end{cases} \tag{7.2}$$

where

$$f(x) = \frac{1}{2}\|\mathcal{A}x - \underbrace{(R - \mathcal{L}e)}_{b}\|_2^2 = \frac{1}{2}\|\mathcal{A}x - b\|_2^2 \tag{7.3}$$

$$f(e) = \frac{1}{2}\|\mathcal{L}e - \underbrace{(R - \mathcal{A}x)}_{c}\|_2^2 + \frac{\beta}{2}\|v - \nabla e\|_2^2 = \frac{1}{2}\|\mathcal{L}e - c\|_2^2 + \frac{\beta}{2}\|v - \nabla e\|_2^2 \tag{7.4}$$

For the first iteration, find the first step size $\rho_x^{(0)}$ and $\rho_e^{(0)}$ by line search and do steepest descent.
Given x_0:

$$\nabla f(x_0) = \mathcal{A}^*(\mathcal{A}x_0 - b) = d_x \tag{7.5}$$

$$\rho_x^{(0)} = \underset{\rho}{\operatorname{argmin}} f(x_0 - \rho\nabla f(x_0)) = \underset{\rho}{\operatorname{argmin}} \underbrace{\frac{1}{2}\|\mathcal{A}(x_0 - \rho d_x) - b\|_2^2}_{\phi(\rho)} \tag{7.6}$$

$\Leftrightarrow$

$$\frac{\partial\phi(\rho)}{\partial\rho} = 0 \tag{7.7}$$

$$-(\mathcal{A}d_x)^T(-\rho_x^{(0)}\mathcal{A}d_x + \mathcal{A}x_0 - b) = 0$$

$$\rho_x^{(0)}(\mathcal{A}d_x)^T(\mathcal{A}d_x) - (\mathcal{A}d_x)^T(\mathcal{A}x_0 - b) = 0$$

$$\rho_x^{(0)} = \frac{(\mathcal{A}d_x)^T(\mathcal{A}x_0 - b)}{(\mathcal{A}d_x)^T(\mathcal{A}d_x)} \tag{7.8}$$

Given e_0:

$$\nabla f(e_0) = \mathcal{L}^*(\mathcal{L}e_0 - c) - \beta\nabla^*(v - \nabla e_0) = d_e \tag{7.9}$$

$$\rho_e^{(0)} = \underset{\rho}{\operatorname{argmin}} f(e_0 - \rho\nabla f(e_0)) = \underset{\rho}{\operatorname{argmin}} \underbrace{\frac{1}{2}\|\mathcal{L}(e_0 - \rho_e^{(0)}d_e) - c\|_2^2 + \frac{\beta}{2}\left\|v - \nabla(e_0 - \rho_e^{(0)}d_e)\right\|_2^2}_{\phi(\rho)} \tag{7.10}$$

$\Leftrightarrow$

$$\frac{\partial\phi(\rho)}{\partial\rho} = 0 \tag{7.11}$$

$$\rho_e^{(0)} = \frac{(\mathcal{L}d_e)^T(\mathcal{L}e_0 - c) - \beta(\nabla d_e)^T(v - \nabla e_0)}{(\mathcal{L}d_e)^T(\mathcal{L}d_e) + \beta(\nabla d_e)^T(\nabla d_e)} \tag{7.12}$$

After getting $\rho_x^{(0)}$ and $\rho_e^{(0)}$, we apply the Barzilai-Borwein (BB) method for choosing step sizes for the following iterations[1][4].
Let

$$s^{k-1} = x_k - x_{k-1}, \qquad y^{k-1} = d_x^k - d_x^{k-1} \tag{7.13}$$

The BB step is defined so that it corresponds to premultiplying the negative gradient by a multiple of identity that has a quasi-Newton property, specifically,

$$\rho_x^{(k),BB1} = \frac{(s^{k-1})^T s^{k-1}}{(s^{k-1})^T y^{k-1}} \quad or \quad \rho_x^{(k),BB2} = \frac{(s^{k-1})^T y^{k-1}}{(y^{k-1})^T y^{k-1}} \tag{7.14}$$

Same for e.

7.2 Continuation scheme on penalty parameters

The coordination of three penalty parameters (α, β and γ) is also a serious issue in practical. We did an experiment that only reconstruct triangles with exactly the same α in Section 6, the reconstruction results are shown as follow:

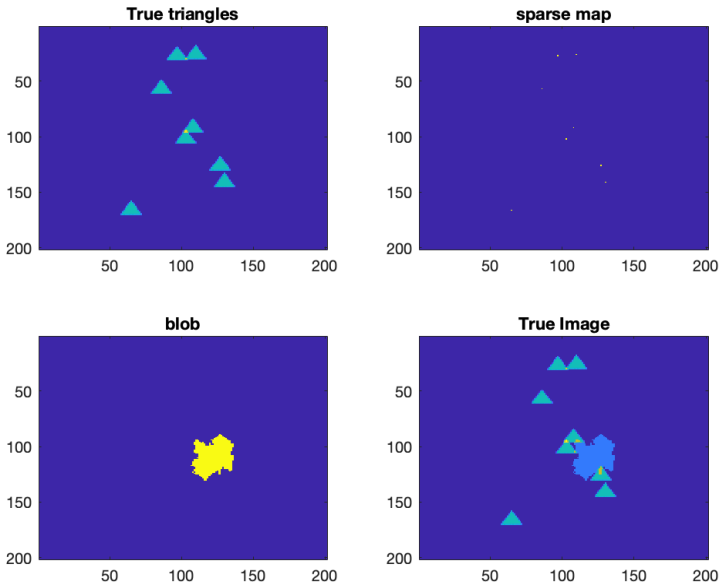

Figure 5: True triangles

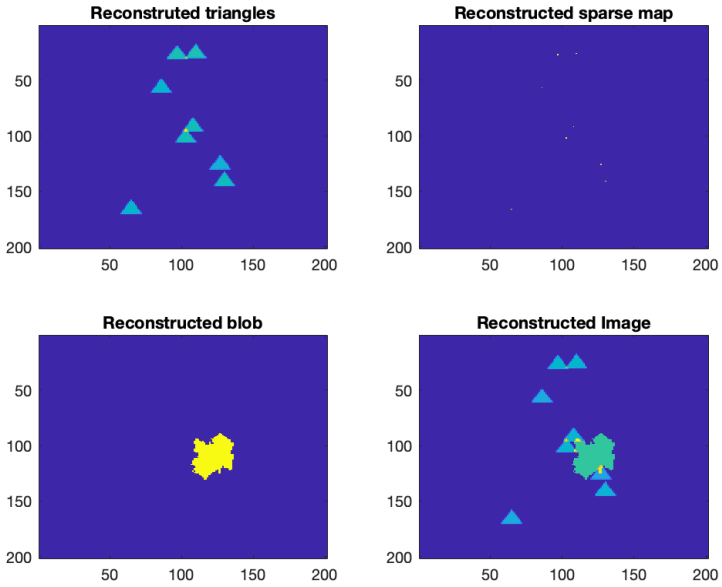

Figure 6: Reconstruction results

Notice that we don't consider about blob image here, just plug in the ground truth blob image for visualization.

We can see that the results in Section 6 are worse than here. That is to say, when we put things together, the choices for β and γ will definitely affect the performance of sparse signal recovery. Theoretically, β is used to penalize the error from auxiliary variable, which is supposed to be as large as possible. However, its ratio with γ, i.e., $\frac{\gamma}{\beta}$, also plays a role for the 2D shrinkage operator, thereby affecting the reconstruction result of "garbage bin" image e. So if β is too large, the reconstruction of "garbage bin" image e can be very poor.

An idea is to use a continuation scheme on penalty parameters instead of three fixed constants. Obviously, fixed penalty parameters are not invariant to different data. The specific continuation scheme for our problem is still to be discussed, in a sense, we can refer to the continuation scheme for β in [3]:

Algorithm 2 Continuation scheme for β

Input: all input variables for Algorithm 1, $\beta_0 > 0$, and $\beta_{max} > \beta_0$
Initialize: all input variables for Algorithm 1, $\beta = \beta_0$

1: **while** $\beta \leq \beta_{max}$ **do**
2: Run Algorithm 1 until stopping criteria is met
3: $\beta \leftarrow 2 * \beta$
4: **end while**

References

[1] Jonathan Barzilai and Jonathan M Borwein. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1):141–148, 1988.

[2] Richard Courant et al. Variational methods for the solution of problems of equilibrium and vibrations. *Lecture notes in pure and applied mathematics*, pages 1–1, 1994.

[3] Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.

[4] Zaiwen Wen, Wotao Yin, Donald Goldfarb, and Yin Zhang. A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation. *SIAM Journal on Scientific Computing*, 32(4):1832–1857, 2010.