The Split Bregman Method for L_1 Regularized Problems: An Overview

Pardis Noorzad[1]

[1]Department of Computer Engineering and IT
Amirkabir University of Technology

Khordad 1389

Image Restoration and Variational Models

- Fundamental problem in image restoration: **denoising**
- Denoising is an important step in machine vision tasks
- Concern is to *preserve* important image features
  - edges, texture

  while *removing* noise
- Variational models have been very successful

TV-based Image Restoration

- **Total variation based** image restoration models first introduced by Rudin, Osher, and Fatemi [ROF92]
- An early example of PDE based **edge preserving** denoising
- Has been extended and solved in a variety of ways
- Here, the **Split Bregman** method is introduced

Denoising

Decomposition

$f = u + v$

- $f : \Omega \to \mathbb{R}$ is the noisy image
- Ω is the bounded open subset of $\mathbb{R}^2$
- u is the true signal
- $v \sim N(0, \sigma^2)$ is the white Gaussian noise

Conventional Variational Model

Easy to solve — results are dissapointing

$$\min \int_{\Omega} (u_{xx} + u_{yy})^2 dxdy$$

such that

$$\int_{\Omega} u dxdy = \int_{\Omega} f dxdy$$

(white noise is of zero mean)

$$\int_{0} \frac{1}{2}(u-f)^2 dxdy = \sigma^2,$$

(a priori information about v)

The ROF Model

Difficult to solve — successful for denoising

$$\min_{u \in \mathrm{BV}(\Omega)} \{\|u\|_{\mathrm{BV}} + \lambda \|f - u\|_2^2\}$$

- $\lambda > 0$: scale parameter
- $\mathrm{BV}(\Omega)$: space of functions with **bounded variation** on Ω
- $\|.\|$: **BV seminorm** or **total variation** given by,

$$\|u\|_{\mathrm{BV}} = \int_\Omega |\nabla u|$$

BV seminorm

- It's use is essential — allows image recovery with edges
- What if first term were replaced by $\int_\Omega |\nabla u|^p$?
  - Which is both **differentiable** and **strictly convex**
- No good! For $p > 1$, its derivative has **smoothing effect** in the optimality condition
- For **TV** however, the operator is degenerate, and affects only **level lines** of the image

Iterative Regularization

Adding back the noise

- In the ROF model, $u - f$ is treated as **error** and discarded
- In the decomposition of f into signal u and additive noise v
  - There exists some **signal** in v
  - And some smoothing of textures in u
- Osher et al. [OBG+05] propose an iterated procedure to **add the noise back**

Iterative Regularization

The iteration

Step 1: Solve the ROF model to obtain:

$$u_1 = \underset{u \in BV(\Omega)}{\arg\min} \left\{ \int |\nabla u| + \lambda \int (f - u)^2 \right\}$$

Step 2: Perform a correction step:

$$u_2 = \underset{u \in BV(\Omega)}{\arg\min} \left\{ \int |\nabla u| + \lambda \int (f + v_1 - u)^2 \right\}$$

(v_1 is the noise estimated by the first step, $f = u_1 + v_1$)

Definition

L_1 regularized optimization

$$\min_u \|\Phi(u)\|_1 + H(u)$$

- Many important problems in imaging science (and other problems in engineering) can be posed as L_1 **regularized optimization** problems
- $\|.\|_1$: the L_1 norm
- both $\|\Phi(u)\|_1$ and $H(u)$ are convex functions

Easy Instances

$$\arg\min_u \|Au - f\|_2^2 \quad \text{differentiable}$$

$$\arg\min_u \|u\|_1 + \|u - f\|_2^2 \quad \text{solvable by shrinkage}$$

Shrinkage

or Soft Thresholding

Solves the L_1 problem of the form (H(.) is convex and differentiable):

$$\arg\min_u \mu\|u\|_1 + H(u)$$

Based on this iterative scheme

$$u^{k+1} \rightarrow \arg\min_u \mu\|u\|_1 + \frac{1}{2\delta^k}\|u - (u^k - \delta^k \nabla H(u^k))\|^2$$

Shrinkage

Continued

Since unknown u is **componentwise separable**, each component can be independently obtained:

$$u_i^{k+1} = \operatorname{shrink}((u^k - \delta^k \nabla H(u^k))_i, \mu\delta^k),\ i = 1, \ldots, n,$$

$$\operatorname{shrink}(y, \alpha) := \operatorname{sgn}(y) \max\{|y| - \alpha, 0\} = \begin{cases} y - \alpha, & y \in (\alpha, \infty), \\ 0, & y \in [-\alpha, \alpha], \\ y + \alpha, & y \in (-\infty, -\alpha). \end{cases}$$

Hard Instances

$$\arg\min_u \|\Phi(u)\|_1 + \|u - f\|_2^2$$

$$\arg\min_u \|u\|_1 + \|Au - f\|_2^2$$

What makes these problems hard?
The **coupling** between the L_1 and L_2 terms.

Split the L_1 and L_2 components

To solve the general regularization problem:

$$\arg\min_{u} \|\Phi(u)\|_1 + H(u)$$

Introduce $d = \Phi(u)$ and solve the constrained problem

$$\arg\min_{u,d} \|d\|_1 + H(u) \quad \text{such that} \quad d = \Phi(u)$$

Split the L_1 and L_2 components

Continued

Add an L_2 penalty term to get an unconstrained problem

$$\arg\min_{u,d} \|d\|_1 + H(u) + \frac{\lambda}{2}\|d - \Phi(u)\|^2$$

- Obvious way is to use the **penalty method** to solve this
- However, as $\lambda_k \to \infty$, the **condition number of the Hessian** approaches infinity, making it impractical to use fast iterative methods like Conjugate Gradient to approximate the inverse of the Hessian.

Analog of adding the noise back

The optimization problem is solved by iterating

$$
\begin{aligned}
(u^{k+1}, d^{k+1}) &= \arg\min_{u,d} \|d\|_1 + H(u) + \frac{\lambda}{2}\|d - \Phi(u) - b^k\|^2 \\
b^{k+1} &= b^k + (\Phi(u) - d^k)
\end{aligned}
$$

The iteration in the first line can be done separately for u and d.

3-step Algorithm

Step 1: $u^{k+1} = \arg\min_u H(u) + \frac{\lambda}{2}\|d^k - \Phi(u) - b^k\|_2^2$

Step 2: $d^{k+1} = \arg\min_d \|d\|_1 + \frac{\lambda}{2}\|d^k - \Phi(u) - b^k\|_2^2$

Step 3: $b^{k+1} = b^k + \Phi(u^{k+1}) - d^{k+1}$

- Step 1 is now a differentiable optimization problem, we'll solve with **Gauss Seidel**
- Step 2 can be solved efficiently with shrinkage
- Step 3 is an explicit evaluation

Anisotropic TV

$$\arg\min_u |\nabla_x u| + |\nabla_y u| + \frac{\mu}{2}\|u - f\|_2^2$$

Anisotropic TV

The steps

Step 1: $u^{k+1} = G(u^k)$

Step 2: $d_x^{k+1} = \text{shrink}(\nabla_x u^{k+1} + b_x^k, \frac{1}{\lambda})$

Step 3: $d_y^{k+1} = \text{shrink}(\nabla_y u^{k+1} + b_y^k, \frac{1}{\lambda})$

Step 4: $b_x^{k+1} = b_x^k + (\nabla_x u - x)$

Step 5: $b_y^{k+1} = b_y^k + (\nabla_y u - y)$

- $G(u^k)$: result of one Gauss-Seidel sweep for the corresponding L_2 optimization
- This algorithm is cheap — each step is a few operations per pixel

Isotropic TV

With similar steps

$$\arg\min_u \sum_i \sqrt{(\nabla_x u)_i^2 + (\nabla_y u)_i^2} + \frac{\mu}{2}\|u - f\|_2^2$$

Split Bregman is fast

Intel Core 2 Duo desktop (3 GHz), compiled with g++

Anisotropic		
Image	**Time/cycle (sec)**	**Time Total (sec)**
256×256 Blocks	0.0013	0.068
512×512 Lena	0.0054	0.27

Isotropic		
Image	**Time/cycle (sec)**	**Time Total (sec)**
256×256 Blocks	0.0018	0.0876
512×512 Lena	0.011	0.55

Split Bregman is fast

Compared to Graph Cuts

Image	Split Bregman	Graph Cuts(4 point)	Graph Cuts(16 point)
256×256 Blocks	0.0732	0.214	0.468
512×512 Lena	0.2412	0.709	1.51

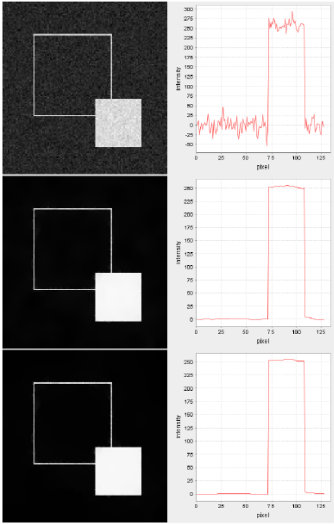
Intermediate images are smooth

Intermediate images are smooth

Noisy (sigma=15)

10 Iterations

50 Iterations



References

- Tom Goldstein and Stanley Osher, *The split bregman method for l1-regularized problems*, SIAM Journal on Imaging Sciences **2** (2009), no. 2, 323–343.
- Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Modeling and Simulation **4** (2005), 460–489.
- Leonid I. Rudin, Stanley Osher, and Emad Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D **60** (1992), no. 1-4, 259–268.

Thank you for your attention. Any questions?