The Split Bregman Method for L1-Regularized Problems

Tom Goldstein

May 22, 2008

Some Common L1 Regularized Problems

TV Denoising: $$\min_u \|u\|_{BV} + \frac{\mu}{2}\|u - f\|_2^2$$

De-Blurring/Deconvolution: $$\min_u \|u\|_{BV} + \frac{\mu}{2}\|Ku - f\|_2^2$$

Basis Pursuit/Compressed Sensing MRI: $$\min_u \|u\|_{BV} + \frac{\mu}{2}\|\mathcal{F}u - f\|_2^2$$

What Makes these Problems Hard??

- Some "easy" problems...

$$\arg\min_u \|Au - f\|_2^2 \text{ (Differentiable)}$$

$$\arg\min_u |u|_1 + \|u - f\|_2^2 \text{ (Solvable by shrinkage)}$$

- Some "hard" problems

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

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

- What makes these problems hard is the "coupling" between L1 and L2 terms

A Better Formulation

- We want to solve the general L1 regularization problem:

$$\arg\min_{u} |\Phi u| + \|Ku - f\|^2$$

- We need to "split" the L1 and L2 components of this energy
- Introduce a new variable

$$\text{let } d = \Phi u$$

- We wish to solve the constrained problem

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

Solving the Constrained Problem

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

- We add an L2 penalty term to get an unconstrained problem

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

- This splitting was independently introduced by Wang and Dr. Yin Zhang (FTVd)
- We need a way of modifying this problem to get exact enforcement of the constraint
- The most obvious way is to use continuation: let $\lambda_n \to \infty$
- Continuation makes the condition number bad

A Better Solution: Use Bregman Iteration

- We group the first two energy terms together:

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

- to get...

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

- We now define the "Bregman Distance" of this convex functional as

$$D_E^p(u,d,u^k,d^k) = E(u,d) - \langle p_u^k, u - u^k\rangle + \langle p_d^k, d - d^k\rangle$$

A Better Solution: Use Bregman Iteration

- Rather than solve $\min E(u,d) + \frac{\lambda}{2}\|d - \Phi(u)\|^2$ we recursively solve

$$(u^{k+1}, d^{k+1}) = \arg\min_{u,d} D_E^p(u, d, u^k, d^k) + \frac{\lambda}{2}\|d - \Phi(u)\|_2^2$$

- or

$$\arg\min_{u,d} E(u,d) - \langle p_u^k, u - u^k \rangle + \langle p_d^k, d - d^k \rangle + \frac{\lambda}{2}\|d - \Phi(u)\|_2^2$$

- Where p_u and p_d are in the subgradient of E with respect to the variables u and d

Why does this work?

- Because of the convexity of the functionals we are using, it can be shown that

$$\|d - \Phi u\| \to 0 \text{ as } k \to \infty$$

- Furthermore, is can be shown that the limiting values, $u^* = \lim_{k\to\infty} u^k$ and $d^* = \lim_{k\to\infty} d^k$ satisfy the original constrained optimization problem

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

- It therefore follows that u^* is a solution to the original L1 constrained problem

$$u^* = \arg\min_{u} |\Phi u| + \|Ku - f\|^2$$

Don't Worry! This isn't as complicated as it looks

- As is done for Bregman iterative denoising, we can get explicit formulas for p_u and p_d, and use them to simplify the iteration
- This gives us the simplified iteration

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

- This is the analog of "adding the noise back" when we use Bregman for denoising

Summary of what we have so far

- We began with an L1-constrained problem

$$u^* = \arg\min |\Phi u| + \|Ku - f\|^2$$

- We form the "Split Bregman" formulation

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

- For some optimal value $b^* = b$ of the Bregman parameter, these two problems are equivalent
- We solve the optimization problem by iterating

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

Why is this better?

- We can break this algorithm down into three easy steps

$$
\begin{aligned}
\text{Step 1} &: \quad u^{k+1} = \arg\min_u H(u) + \frac{\lambda}{2}\|d - \Phi(u) - b^k\|_2^2 \\
\text{Step 2} &: \quad d^{k+1} = \arg\min_d |d|_1 + \frac{\lambda}{2}\|d - \Phi(u) - b^k\|_2^2 \\
\text{Step 3} &: \quad b^{k+1} = b^k + \Phi(u^{k+1}) - d^{k+1}
\end{aligned}
$$

- Because of the decoupled form, step 1 is now a *differentiable* optimization problem - we can directly solve it with tools like Fourier Transform, Gauss-Seidel, CG, etc...
- Step 2 can be solved efficiently using shrinkage

$$d^{k+1} = \mathit{shrink}(\Phi(u^{k+1}) + b^k, 1/\lambda)$$

- Step 3 is explicit, and easy to evaluate

Example: Fast TV Denoising

- We begin by considering the Anisotropic ROF denoising problem
$$\arg\min_u |\nabla_x u| + |\nabla_y u| + \frac{\mu}{2}\|u - f\|_2^2$$
- We then write down the *Split Bregman* formulation

$$\begin{aligned}\arg\min_{x,y,u} \quad & |d_x| + |d_y| + \frac{\mu}{2}\|u - f\|_2^2 \\ + \quad & \frac{\lambda}{2}\|d_x - \nabla_x u - b_x\|_2^2 \\ + \quad & \frac{\lambda}{2}\|d_y - \nabla_y u - b_y\|_2^2\end{aligned}$$

Example: Fast TV Denoising

- The TV algorithm then breaks down into these steps:

$$
\begin{aligned}
\text{Step 1} &: \quad u^{k+1} = G(u^k) \\
\text{Step 2} &: \quad d_x^{k+1} = shrink(\nabla_x u^{k+1} + b_x^k, 1/\lambda) \\
\text{Step 3} &: \quad d_y^{k+1} = shrink(\nabla_y u^{k+1} + b_y^k, 1/\lambda) \\
\text{Step 4} &: \quad b_x^{k+1} = b_x^k + (\nabla_x u - x) \\
\text{Step 5} &: \quad b_y^{k+1} = b_y^k + (\nabla_y u - y)
\end{aligned}
$$

  where $G(u^k)$ represents the results of one Gauss Seidel sweep for the corresponding L2 optimization problem.
- This is very cheap – each step is only a few operations per pixel

Isotropic TV

- This method can do isotropic TV using the following decoupled formulation

$$
\begin{aligned}
\arg\min \quad & \sqrt{d_x^2 + d_y^2} + \frac{\mu}{2}\|u - f\|_2^2 \\
+ \quad & \frac{\lambda}{2}\|d_x - \nabla_x u - b_x\|_2^2 + \frac{\lambda}{2}\|d_y - \nabla_y u - b_y\|_2^2
\end{aligned}
$$

- We now have to solve for (d_x, d_y) using the generalized shrinkage formula (Yin et. al.)

$$
\begin{aligned}
d_x^{k+1} &= \max(s^k - 1/\lambda, 0)\frac{\nabla_x u^k + b_x^k}{s^k} \\
d_y^{k+1} &= \max(s^k - 1/\lambda, 0)\frac{\nabla_y u^k + b_y^k}{s^k}
\end{aligned}
$$

where

$$
s^k = \sqrt{(\nabla_x u^k + b_x^k)^2 + (\nabla_y u^k + b_y^k)^2}
$$

Time Trials

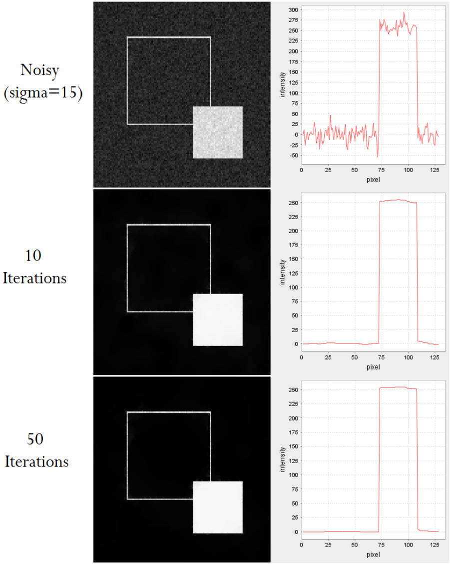
- Time trials were done on a Intel Core 2 Due desktop (3 GHz)
- Linux Platform, compiled with g++

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

This can be made even faster...

- Most of the denoising takes place in first 10 iterations



This can be made even faster...

- Most of the denoising takes place in first 10 iterations
- "Staircases" form quickly, but then take some time to flatten out
- If we are willing to accept a "visual" convergence criteria, we can denoise in about 10 iterations (0.054 sec) for Lena, and 20 iterations (0.024 sec) for the blocky image.

This can be made even faster...

Original

Noisy (sigma=25)

10 Iterations

50 Iterations

Compressed Sensing for MRI

- Many authors (Donoho, Yin, etc...) get superior reconstruction using both TV and Besov regularizers
- We wish to solve

$$\arg\min_u |\nabla u| + |Wu| + \frac{\mu}{2}\|R\mathcal{F}u - f^k\|_2^2$$

where R comprises a subset of rows of the identity, and W is an orthogonal wavelet transform (Haar).

- Apply the "Split Bregman" method: Let $w \leftarrow Wu$, $d_x \leftarrow \nabla_x u$, and $d_y \leftarrow \nabla_y u$

$$\arg\min_{u,d_x,d_y,w} \sqrt{d_x^2 + d_y^2} + |w| + \frac{\mu}{2}\|R\mathcal{F}u - f\|_2^2$$
$$+\frac{\lambda}{2}\|d_x - \nabla_x u - b_x\|_2^2 + \frac{\lambda}{2}\|d_y - \nabla_y u - b_y\|_2^2$$
$$+\frac{\gamma}{2}\|w - Wu - b_w\|_2^2$$

Compressed Sensing for MRI

- The optimality condition for u is circulant:

$$(\mu\mathcal{F}^T R^T R\mathcal{F} - \lambda\Delta + \gamma I)u^{k+1} = rhs^k$$

- The resulting algorithm is

Unconstrained CS Optimization Algorithm

$u^{k+1} = \mathcal{F}^{-1}K^{-1}\mathcal{F}rhs^k$

$(d_x^{k+1}, d_y^{k+1}) = shrink(\nabla_x u + b_x, \nabla_y u + b_y, 1/\lambda)$

$w^{k+1} = shrink(Wu + b_w, 1/\gamma)$

$b_x^{k+1} = b_x^k + (\nabla_x u - d_x)$

$b_y^{k+1} = b_y^k + (\nabla_y u - d_y)$

$b_w^{k+1} = b_w^k + (Wu - w)$

Compressed Sensing for MRI

- To solve the constrained problem

$$\arg\min_u |\nabla u| + |Wu| \text{ such that } \|R\mathcal{F}u - f\|_2 < \sigma$$

  we use "double Bregman"
- First, solve the unconstrained problem

$$\arg\min_u |\nabla u| + |Wu| + \frac{\mu}{2}\|R\mathcal{F}u - f^k\|_2^2$$

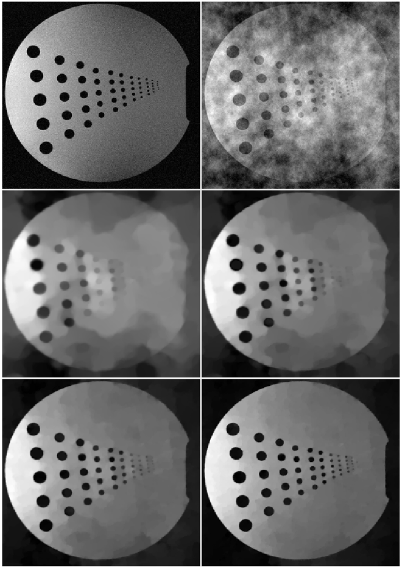
  by performing "inner" iterations
- Then, update

$$f^{k+1} = f^k + f - R\mathcal{F}u^{k+1}$$

  this is an "outer iteration"

Compressed Sensing

- 256 x 256 MRI of phantom, 30%



Bregman Iteration vs Continuation

- As $\lambda \to \infty$, the condition number of each sub-problem goes to ∞
- This is okay if we have a direct solver for each sub-problem (such as FFT)
- Drawback: Direct solvers are slower than iterative solvers, or may not be available
- With Bregman iteration, condition number stays constant - we can use efficient iterative solvers

Example: Direct Solvers May be Inefficient

- TV-L1:

$$\arg\min_{u} |\nabla u| + \mu|u - f|$$

- Split-Bregman formulation

$$\arg\min_{u,d} |d| + \mu|v - f| + \frac{\lambda}{2}\|d - \nabla u - b_d\|_2^2 + \frac{\gamma}{2}\|u - v - b_v\|_2^2$$

- We must solve the sub-problem

$$(\mu I - \lambda\Delta)u = RHS$$

- If $\lambda \approx \mu$, then this is strongly diagonally dominant: use Gauss-Seidel (cheap)
- If $\lambda >> \mu$, then we must use a direct solver: 2 FFT's per iteration (expensive)

Example: Direct Solvers May Not Exist

- Total-Variation based Inpainting:

$$\arg\min_u \int_\Omega |\nabla u| + \mu \int_{\Omega/D} (u-f)^2$$

$$\arg\min_u |\nabla u| + \mu \|Ru - f\|^2$$

  where R consists of rows of the identity matrix.

- The optimization sub-problem is

$$(\mu R^T R - \lambda\Delta)u = RHS$$

- Not Circulant! - We have to use an iterative solver (e.g. Gauss-Seidel)

Generalizations

- Bregman Iteration can be used to solve a wide range of non-L1 problems

$$\arg\min J(u) \text{ such that } A(u) = 0$$

where J and $\|A(\cdot)\|^2$ are convex.

- We can use a Bregman-like penalty function

$$u^{k+1} = \arg\min J(u) + \frac{\lambda}{2}\|A(u) - b^k\|^2$$

$$b^{k+1} = b^k - A(u)$$

- Theorem: Any fixed point of the above algorithm is solution to the original constrained problem
- Convergence can be proved for a broad class of problems: If J is strictly convex and twice differentiable, then $\exists\lambda_0 > 0$ such that the algorithm converges for any

$$\lambda < \lambda_0$$

Conclusion

- The Split Bregman formulation is a fast tool that can solve almost any L1 regularization problem
- Small memory footprint
- This method is easily parallelized for large problems
- Easy to code

Conclusion

Acknowledgment

We thank Jie Zheng for his helpful discussions regarding MR image processing. This publication was made possible by the support of the National Science Foundation's GRFP program, as well as ONR grant N000140710810 and the Department of Defense.