Alternating Direction Method of Multipliers

Lecturer: Ryan Tibshirani
Convex Optimization 10-725/36-725

Last time: dual methods

Consider the problem

$$\min_x \; f(x) \;\; \text{subject to} \;\; Ax = b$$

where f is strictly convex and closed. Denote Lagrangian

$$L(x, u) = f(x) + u^T (Ax - b)$$

Dual gradient ascent repeats, for $k = 1, 2, 3, \ldots$

$$\begin{aligned} x^{(k)} &= \operatorname*{argmin}_x \; L(x, u^{(k-1)}) \\ u^{(k)} &= u^{(k-1)} + t_k (Ax^{(k)} - b) \end{aligned}$$

Good: x update decomposes when f does. Bad: require stringent assumptions (strong convexity of f) to ensure convergence

Augmented Lagrangian method (also called method of multipliers) considers the modified problem, for a parameter $\rho > 0$,

$$\begin{aligned} \min_x \quad & f(x) + \frac{\rho}{2}\|Ax - b\|_2^2 \\ \text{subject to} \quad & Ax = b \end{aligned}$$

uses modified Lagrangian

$$L_\rho(x, u) = f(x) + u^T(Ax - b) + \frac{\rho}{2}\|Ax - b\|_2^2$$

and repeats, for $k = 1, 2, 3, \ldots$

$$\begin{aligned} x^{(k)} &= \operatorname*{argmin}_x \; L_\rho(x, u^{(k-1)}) \\ u^{(k)} &= u^{(k-1)} + \rho(Ax^{(k)} - b) \end{aligned}$$

Good: better convergence properties. Bad: lose decomposability

Alternating direction method of multipliers

Alternating direction method of multipliers or ADMM: combines the best of both methods. Consider a problem of the form:

$$\min_{x,z} \; f(x) + g(z) \;\; \text{subject to} \;\; Ax + Bz = c$$

We define augmented Lagrangian, for a parameter $\rho > 0$,

$$L_\rho(x, z, u) = f(x) + g(z) + u^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2$$

We repeat, for $k = 1, 2, 3, \ldots$

$$\begin{aligned}
x^{(k)} &= \underset{x}{\operatorname{argmin}} \; L_\rho(x, z^{(k-1)}, u^{(k-1)}) \\
z^{(k)} &= \underset{z}{\operatorname{argmin}} \; L_\rho(x^{(k)}, z, u^{(k-1)}) \\
u^{(k)} &= u^{(k-1)} + \rho(Ax^{(k)} + Bz^{(k)} - c)
\end{aligned}$$

Convergence guarantees

Under modest assumptions on f, g (these do not require A, B to be full rank), the ADMM iterates satisfy, for any $\rho > 0$:

- Residual convergence: $r^{(k)} = Ax^{(k)} - Bz^{(k)} - c \to 0$ as $k \to \infty$, i.e., primal iterates approach feasibility
- Objective convergence: $f(x^{(k)}) + g(z^{(k)}) \to f^\star + g^\star$, where $f^\star + g^\star$ is the optimal primal objective value
- Dual convergence: $u^{(k)} \to u^\star$, where $u^\star$ is a dual solution

For details, see Boyd et al. (2010). Note that we do not generically get primal convergence, but this is true under more assumptions

Convergence rate: not known in general, theory is currently being developed, e.g., in Hong and Luo (2012), Deng and Yin (2012), Iutzeler et al. (2014), Nishihara et al. (2015). Roughly, it behaves like a first-order method (or a bit faster)

ADMM in scaled form

It is often easier to express the ADMM algorithm in a scaled form, where we replace the dual variable u by a scaled variable $w = u/\rho$. In this parametrization, the ADMM steps are:

$$\begin{aligned} x^{(k)} &= \underset{x}{\operatorname{argmin}} \ f(x) + \frac{\rho}{2}\|Ax + Bz^{(k-1)} - c + w^{(k-1)}\|_2^2 \\ z^{(k)} &= \underset{z}{\operatorname{argmin}} \ g(z) + \frac{\rho}{2}\|Ax^{(k)} + Bz - c + w^{(k-1)}\|_2^2 \\ w^{(k)} &= w^{(k-1)} + Ax^{(k)} + Bz^{(k)} - c \end{aligned}$$

Note that here the kth iterate $w^{(k)}$ is just given by a running sum of residuals:

$$w^{(k)} = w^{(0)} + \sum_{i=1}^{k} \left(Ax^{(i)} + Bz^{(i)} - c\right)$$

Outline

Today:

- Examples, practicalities
- Consensus ADMM
- Faster convergence?

Connection to proximal operators

Consider

$$\min_x \; f(x) + g(x) \quad \Longleftrightarrow \quad \min_{x,z} \; f(x) + g(z) \;\; \text{subject to} \;\; x = z$$

ADMM steps (equivalent to Douglas-Rachford, here):

$$\begin{aligned} x^{(k)} &= \text{prox}_{f,1/\rho}(z^{(k-1)} - w^{(k-1)}) \\ z^{(k)} &= \text{prox}_{g,1/\rho}(x^{(k)} + w^{(k-1)}) \\ w^{(k)} &= w^{(k-1)} + x^{(k)} - z^{(k)} \end{aligned}$$

where $\text{prox}_{f,1/\rho}$ is the proximal operator for f at parameter $1/\rho$, and similarly for $\text{prox}_{g,1/\rho}$

In general, the update for block of variables reduces to prox update whenever the corresponding linear transformation is the identity

Example: lasso regression

Given $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$, recall the lasso problem:

$$\min_{\beta} \ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

We can rewrite this as:

$$\min_{\beta,\alpha} \ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\alpha\|_1 \ \text{ subject to } \ \beta - \alpha = 0$$

ADMM gives us a simple algorithm:

$$\begin{aligned}
\beta^{(k)} &= (X^T X + \rho I)^{-1}\big(X^T y + \rho(\alpha^{(k-1)} - w^{(k-1)})\big) \\
\alpha^{(k)} &= S_{\lambda/\rho}(\beta^{(k)} + w^{(k-1)}) \\
w^{(k)} &= w^{(k-1)} + \beta^{(k)} - \alpha^{(k)}
\end{aligned}$$

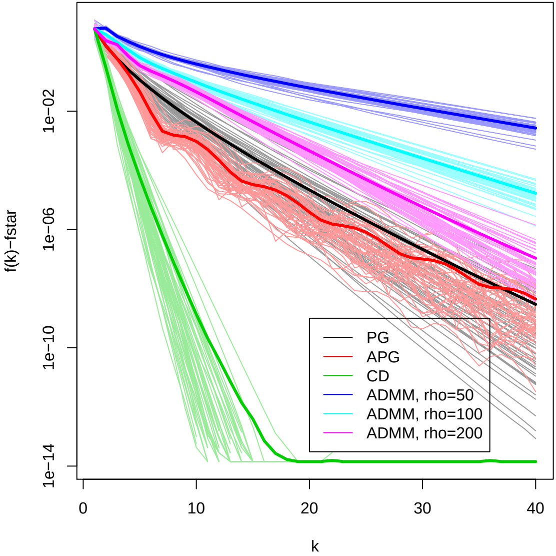
Notes:

- The matrix $X^T X + \rho I$ is always invertible, regardless of X
- If we compute a factorization (say Cholesky) in $O(p^3)$ flops, then each β update takes $O(p^2)$ flops
- The α update applies the soft-thresolding operator S_t, which recall is defined as

$$[S_t(x)]_j = \begin{cases} x_j - t & x > t \\ 0 & -t \leq x \leq t \\ x_j + t & x < -t \end{cases}, \quad j = 1, \ldots p$$

- ADMM steps are "almost" like repeated soft-thresholding of ridge regression coefficients

Comparison of various algorithms for lasso regression: 50 instances with $n = 100$, $p = 20$



Practicalities

In practice, ADMM usually obtains a relatively accurate solution in a handful of iterations, but it requires a large number of iterations for a highly accurate solution (generally behaves like a first-order method)

Choice of ρ can greatly influence practical convergence of ADMM:

- ρ too large $\rightarrow$ not enough emphasis on minimizing $f + g$
- ρ too small $\rightarrow$ not enough emphasis on feasibility

Boyd et al. (2010) give a strategy for varying ρ; can be useful, but does not have convergence guarantees

Like deriving duals, transforming a problem into one that ADMM can handle is sometimes a bit subtle, since different forms can lead to different algorithms

Example: group lasso regression

Given $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$, recall the group lasso problem:

$$\min_{\beta} \; \frac{1}{2}\|y - X\beta\|_2^2 + \lambda \sum_{g=1}^{G} c_g \|\beta_{(g)}\|_2$$

Rewrite as

$$\min_{\beta,\alpha} \; \frac{1}{2}\|y - X\beta\|_2^2 + \lambda \sum_{g=1}^{G} c_g \|\alpha_{(g)}\|_2 \;\; \text{subject to} \;\; \beta - \alpha = 0$$

ADMM steps are:

$$\begin{aligned}
\beta^{(k)} &= (X^T X + \rho I)^{-1}\big(X^T y + \rho(\alpha^{(k-1)} - w^{(k-1)})\big) \\
\alpha_{(g)}^{(k)} &= R_{c_g \lambda/\rho}\big(\beta_{(g)}^{(k)} + w_{(g)}^{(k-1)}\big), \quad g = 1, \ldots G \\
w^{(k)} &= w^{(k-1)} + \beta^{(k)} - \alpha^{(k)}
\end{aligned}$$

Notes:

- The matrix $X^T X + \rho I$ is always invertible, regardless of X
- If we compute a factorization (say Cholesky) in $O(p^3)$ flops, then each β update takes $O(p^2)$ flops
- The α update applies the group soft-thresolding operator R_t, which recall is defined as

$$R_t(x) = \left(1 - \frac{t}{\|x\|_2}\right)_+ x$$

- Similar ADMM steps follow for a sum of arbitrary norms of as regularizer, provided we know prox operator of each norm
- ADMM algorithm can be rederived when groups have overlap (hard problem to optimize in general!). See Boyd et al. (2010)

Example: sparse subspace estimation

Given $S \in \mathbb{S}_p$ (typically $S \succeq 0$ is a covariance matrix), consider the sparse subspace estimation problem (Vu et al., 2013):

$$\max_Y \ \mathrm{tr}(SY) - \lambda \|Y\|_1 \ \text{ subject to } \ Y \in \mathcal{F}_k$$

where $\mathcal{F}_k$ is the Fantope of order k, namely

$$\mathcal{F}_k = \{Y \in \mathbb{S}^p : 0 \preceq Y \preceq I, \ \mathrm{tr}(Y) = k\}$$

Note that when $\lambda = 0$, the above problem is equivalent to ordinary principal component analysis (PCA)

This above is an SDP and in principle solveable with interior point methods, though these can be complicated to implement and quite slow for large problem sizes

We rewrite the problem as:

$$\min_{Y,Z} -\mathrm{tr}(SY) + I_{\mathcal{F}_k}(Y) + \lambda\|Z\|_1 \quad \text{subject to} \quad Y = Z$$

ADMM steps are:

$$\begin{aligned} Y^{(k)} &= P_{\mathcal{F}_k}(Z^{(k-1)} - W^{(k-1)} + S/\rho) \\ Z^{(k)} &= S_{\lambda/\rho}(Y^{(k)} + W^{(k-1)}) \\ W^{(k)} &= W^{(k-1)} + Y^{(k)} - Z^{(k)} \end{aligned}$$

Here $P_{\mathcal{F}_k}$ is Fantope projection operator, computed by clipping the eigendecomposition $A = U\Sigma U^T$, $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$:

$$P_{\mathcal{F}_k}(A) = U\Sigma_\theta U^T, \quad \Sigma_\theta = \mathrm{diag}(\sigma_1(\theta), \ldots, \sigma_p(\theta))$$

where each $\sigma_i(\theta) = \min\{\max\{\sigma_i - \theta, 0\}, 1\}$, and $\sum_{i=1}^p \sigma_i(\theta) = k$

Example: sparse plus low rank decomposition

Given $M \in \mathbb{R}^{n \times m}$, consider the sparse plus low rank decomposition problem (Candes et al., 2009):

$$\begin{array}{ll} \min_{L,S} & \|L\|_{\mathrm{tr}} + \lambda \|S\|_1 \\ \text{subject to} & L + S = M \end{array}$$

ADMM steps:

$$\begin{aligned} L^{(k)} &= S^{\mathrm{tr}}_{1/\rho}(M - S^{(k-1)} + W^{(k-1)}) \\ S^{(k)} &= S^{\ell_1}_{\lambda/\rho}(M - L^{(k)} + W^{(k-1)}) \\ W^{(k)} &= W^{(k-1)} + M - L^{(k)} - S^{(k)} \end{aligned}$$

where, to distinguish them, we use $S^{\mathrm{tr}}_{\lambda/\rho}$ for matrix soft-thresolding and $S^{\ell_1}_{\lambda/\rho}$ for elementwise soft-thresolding

Example from Candes et al. (2009):

(a) Original frames (b) Low-rank $\hat{L}$ (c) Sparse $\hat{S}$

Consensus ADMM

Consider a problem of the form: $\min_x \sum_{i=1}^B f_i(x)$

The consensus ADMM approach begins by reparametrizing:

$$\min_{x_1,\ldots x_B,x} \sum_{i=1}^B f_i(x_i) \text{ subject to } x_i = x, \; i = 1,\ldots B$$

This yields the decomposable ADMM steps:

$$x_i^{(k)} = \underset{x_i}{\operatorname{argmin}} \; f_i(x_i) + \frac{\rho}{2}\|x_i - x^{(k-1)} + w_i^{(k-1)}\|_2^2, \quad i = 1,\ldots B$$

$$x^{(k)} = \frac{1}{B}\sum_{i=1}^B \left(x_i^{(k)} + w_i^{(k-1)}\right)$$

$$w_i^{(k)} = w_i^{(k-1)} + x_i^{(k)} - x^{(k)}, \quad i = 1,\ldots B$$

Write $\bar{x} = \frac{1}{B}\sum_{i=1}^B x_i$ and similarly for other variables. Not hard to see that $\bar{w}^{(k)} = 0$ for all iterations $k \geq 1$

Hence ADMM steps can be simplified, by taking $x^{(k)} = \bar{x}^{(k)}$:

$$x_i^{(k)} = \underset{x_i}{\operatorname{argmin}} \; f_i(x_i) + \frac{\rho}{2}\|x_i - \bar{x}^{(k-1)} + w_i^{(k-1)}\|_2^2, \quad i = 1, \ldots B$$
$$w_i^{(k)} = w_i^{(k-1)} + x_i^{(k)} - \bar{x}^{(k)}, \quad i = 1, \ldots B$$

To reiterate, the x_i, $i = 1, \ldots B$ updates here are done in parallel

Intuition:

- Try to minimize each $f_i(x_i)$, use (squared) ℓ_2 regularization to pull each x_i towards the average $\bar{x}$
- If a variable x_i is bigger than the average, then w_i is increased
- So the regularization in the next step pulls x_i even closer

General consensus ADMM with regularization

Consider a problem of the form: $\min_x \sum_{i=1}^B f_i(a_i^T x + b_i) + g(x)$

For consensus ADMM, we again reparametrize:

$$\min_{x_1,\ldots x_B, x} \sum_{i=1}^B f_i(a_i^T x_i + b_i) + g(x) \text{ subject to } x_i = x, \ i = 1, \ldots B$$

This yields the decomposable ADMM updates:

$$x_i^{(k)} = \underset{x_i}{\operatorname{argmin}} \ f_i(a_i^T x_i + b_i) + \frac{\rho}{2}\|x_i - x^{(k-1)} + w_i^{(k-1)}\|_2^2, \quad i = 1, \ldots B$$

$$x^{(k)} = \underset{x}{\operatorname{argmin}} \ \frac{B\rho}{2}\|x - \bar{x}^{(k)} - \bar{w}^{(k-1)}\|_2^2 + g(x)$$

$$w_i^{(k)} = w_i^{(k-1)} + x_i^{(k)} - x^{(k)}, \quad i = 1, \ldots B$$

Notes:

- It is no longer true that $\bar{w}^{(k)} = 0$ at a general iteration k, so ADMM steps do not simplify as before
- To reiterate, the x_i, $i = 1, \ldots B$ updates are done in parallel
- Each x_i update can be thought of as a loss minimization on part of the data, with ℓ_2 regularization
- The x update is a proximal operation in regularizer g
- The w update drives the individual variables into consensus
- A different initial reparametrization will give rise to a different ADMM algorithm

See Boyd et al. (2010), Parikh and Boyd (2013) for more details on consensus ADMM, strategies for splitting up into subproblems, and implementation tips

Faster convergence?

ADMM can exhibit much faster convergence than usual, when we parametrize subproblems in a "special way"

- ADMM updates relate closely to block coordinate descent, in which we optimize a criterion in an alternating fashion across blocks of variables
- With this in mind, get fastest convergence when minimizing over blocks of variables leads to updates in nearly orthogonal directions
- Suggests we should design ADMM form (auxiliary constraints) so that primal updates de-correlate as best as possible
- This is done in, e.g., Ramdas and Tibshirani (2014), Wytock et al. (2014), Barbero and Sra (2014)

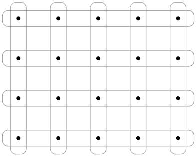
Example: 2d fused lasso

Given an image $Y \in \mathbb{R}^{d \times d}$, equivalently written as $y \in \mathbb{R}^n$, recall the 2d fused lasso or 2d total variation denoising problem:

$$\min_{\Theta} \ \frac{1}{2}\|Y - \Theta\|_F^2 + \lambda \sum_{i,j} \Big(|\Theta_{i,j} - \Theta_{i+1,j}| + |\Theta_{i,j} - \Theta_{i,j+1}| \Big)$$

$$\iff \ \min_{\theta} \ \frac{1}{2}\|y - \theta\|_2^2 + \lambda \|D\theta\|_1$$

Here $D \in \mathbb{R}^{m \times n}$ is a 2d difference operator giving the appropriate differences (across horizontally and vertically adjacent positions)



First way to setup ADMM:

$$\min_{\theta,z} \ \frac{1}{2}\|y-\theta\|_2^2 + \lambda\|z\|_1 \ \ \text{subject to} \ \ \theta = Dz$$

Leads to ADMM steps:

$$\begin{aligned}
\theta^{(k)} &= (I+\rho D^T D)^{-1}\big(y + \rho D^T(z^{(k-1)} + w^{(k-1)})\big) \\
z^{(k)} &= S_{\lambda/\rho}(D\theta^{(k)} - w^{(k-1)}) \\
w^{(k)} &= w^{(k-1)} + z^{(k-1)} - D\theta^{(k)}
\end{aligned}$$

Notes:

- The θ update solves linear system in $I+\rho L$, with $L = D^T D$ the graph Laplacian matrix of the 2d grid, so this can be done efficiently, in roughly $O(n)$ operations
- The z update applies soft thresholding operator S_t
- Hence one entire ADMM cycle uses roughly $O(n)$ operations

Second way to setup ADMM:

$$\min_{\Theta,Z} \quad \frac{1}{2}\|Y-\Theta\|_F^2 + \lambda \sum_{i,j} \Big(|\Theta_{i,j} - \Theta_{i+1,j}| + |Z_{i,j} - Z_{i,j+1}| \Big)$$

$$\text{subject to } \; \Theta = Z$$

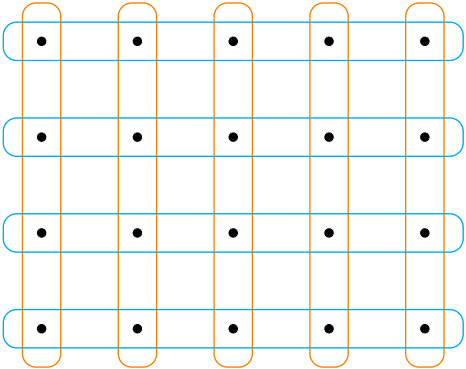
Leads to ADMM steps:

$$\begin{aligned}
\Theta_{\cdot,j}^{(k)} &= \mathrm{FL}_{\lambda/(1+\rho)}^{\mathrm{1d}}\bigg(\frac{Y + \rho(Z_{\cdot,j}^{(k-1)} - W_{\cdot,j}^{(k-1)})}{1+\rho} \bigg), \quad j = 1,\ldots,d \\
Z_{i,\cdot}^{(k)} &= \mathrm{FL}_{\lambda/\rho}^{\mathrm{1d}}\Big(\Theta_{i,\cdot}^{(k)} + W_{i,\cdot}^{(k-1)} \Big), \quad i = 1,\ldots,d \\
W^{(k)} &= W^{(k-1)} + \Theta^{(k)} - Z^{(k)}
\end{aligned}$$

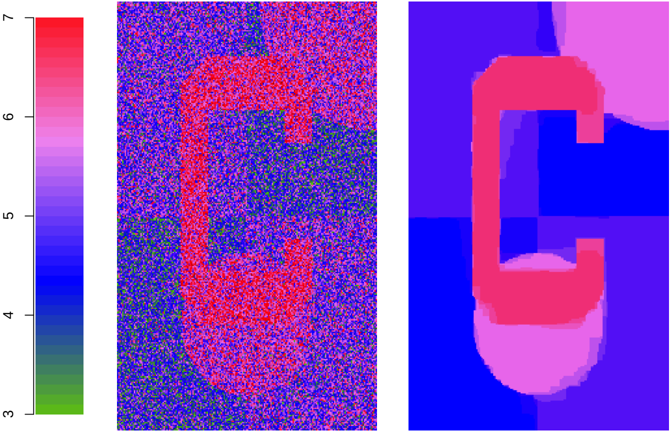
Notes:

- Both Θ, Z updates solve (sequence of) 1d fused lassos, where we write $\mathrm{FL}_\tau^{\mathrm{1d}}(a) = \mathrm{argmin}_x \; \frac{1}{2}\|a-x\|_2^2 + \tau \sum_{i=1}^{d-1} |x_i - x_{i+1}|$

- Critical: each 1d fused lasso solution can be computed exactly in $O(d)$ operations with specialized algorithms (e.g., Johnson, 2013; Davies and Kovac, 2001)
- Hence one entire ADMM cycle again uses $O(n)$ operations



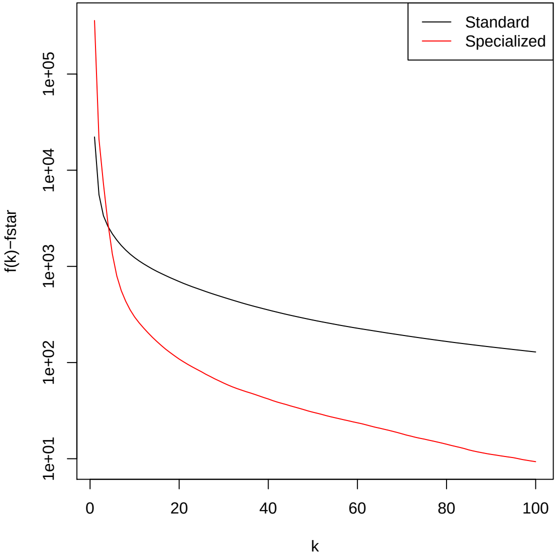
Comparison of 2d fused lasso algorithms: an image of dimension 300×200 (so $n = 60,000$)



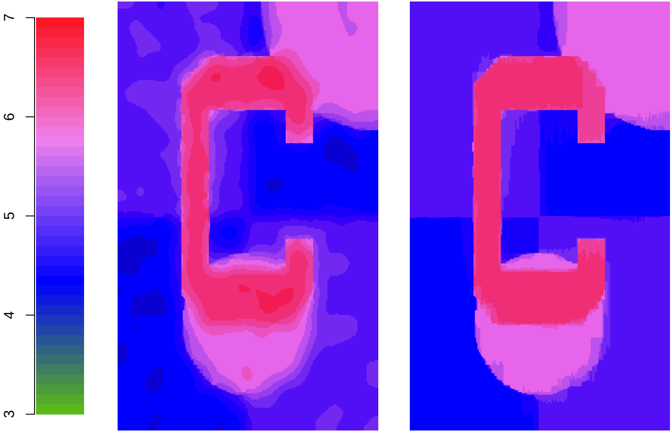
Data

Exact solution

Two ADMM algorithms, let's call them standard and specialized ADMM, convergence of criterions:



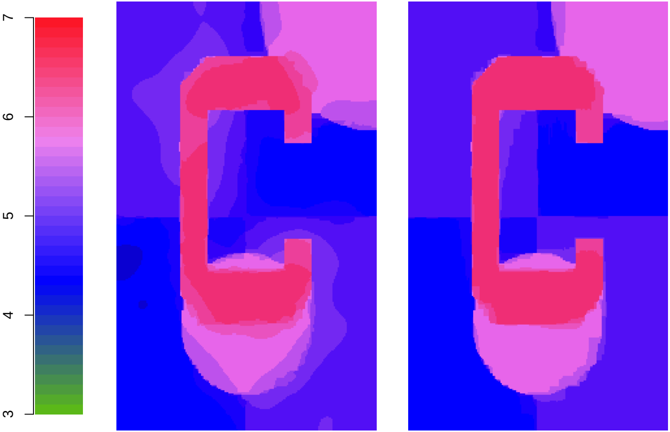
ADMM iterates visualized after $k = 10, 30, 50, 100$ iterations:



Standard ADMM
10 iterations

Specialized ADMM
10 iterations

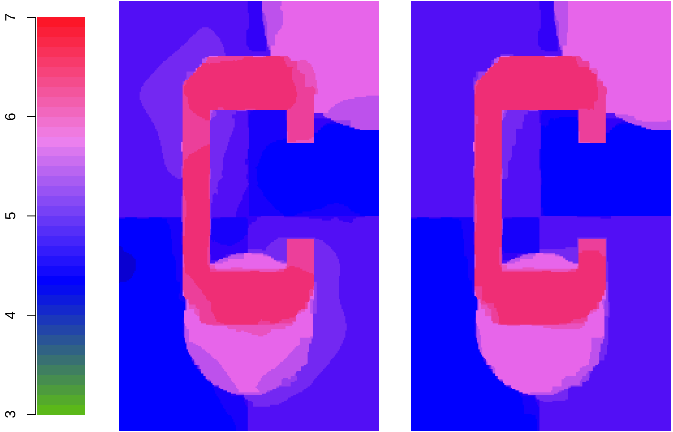
ADMM iterates visualized after $k = 10, 30, 50, 100$ iterations:



Standard ADMM
30 iterations

Specialized ADMM
30 iterations

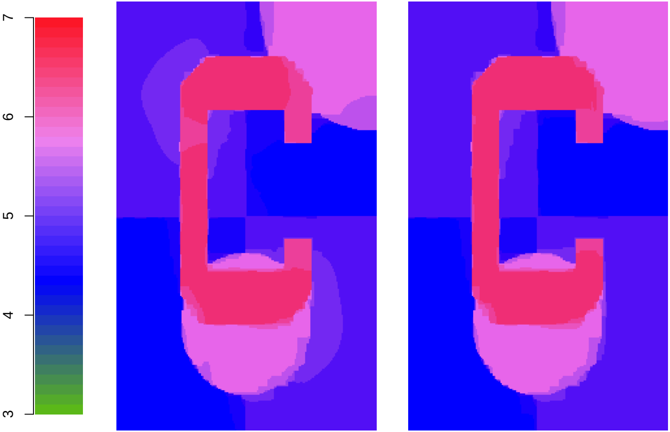
ADMM iterates visualized after $k = 10, 30, 50, 100$ iterations:



Standard ADMM
50 iterations

Specialized ADMM
50 iterations

ADMM iterates visualized after $k = 10, 30, 50, 100$ iterations:



Standard ADMM
100 iterations

Specialized ADMM
100 iterations

References

- A. Barbero and S. Sra (2014), "Modular proximal optimization for multidimensional total-variation regularization"
- S. Boyd and N. Parikh and E. Chu and B. Peleato and J. Eckstein (2010), "Distributed optimization and statistical learning via the alternating direction method of multipliers"
- E. Candes and X. Li and Y. Ma and J. Wright (2009), "Robust principal component analysis?"
- N. Parikh and S. Boyd (2013), "Proximal algorithms"
- A. Ramdas and R. Tibshirani (2014), "Fast and flexible ADMM algorithms for trend filtering"
- V. Vu and J. Cho and J. Lei and K. Rohe (2013),"Fantope projection and selection: a near-optimal convex relaxation of sparse PCA"
- M. Wytock and S. Sra. and Z. Kolter (2014), "Fast Newton methods for the group fused lasso"