Alternating Direction Method of Multipliers

Yu-Xiang Wang
CS292F

Last time: modern stochastic gradient methods

- SGD has slow convergence, we can solve the finite sum problem faster.
- SAG, SAGA, variance reduction
- SVRG and its convergence analysis — more explicit variance reduction and how it helps the algorithm to converge faster
- Adaptive gradient methods: Adagrad, Adam and SignSGD

Reminder: conjugate functions

Recall that given $f : \mathbb{R}^n \to \mathbb{R}$, the function

$$f^*(y) = \max_x \; y^T x - f(x)$$

is called its conjugate

- Conjugates appear frequently in dual programs, since

$$-f^*(y) = \min_x \; f(x) - y^T x$$

- If f is closed and convex, then $f^{**} = f$. Also,

$$x \in \partial f^*(y) \iff y \in \partial f(x) \iff x \in \operatorname*{argmin}_z \; f(z) - y^T z$$

- If f is strictly convex, then $\nabla f^*(y) = \operatorname*{argmin}_z \; f(z) - y^T z$

Dual ascent

Even if we can't derive dual (conjugate) in closed form, we can still use dual-based gradient or subgradient methods

Consider the problem

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

Its dual problem is

$$\max_u \ -f^*(-A^T u) - b^T u$$

where f^* is conjugate of f. Defining $g(u) = -f^*(-A^T u) - b^T u$, note that

$$\partial g(u) = A\partial f^*(-A^T u) - b$$

Therefore, using what we know about conjugates

$$\partial g(u) = Ax - b \quad \text{where} \quad x \in \operatorname*{argmin}_z \; f(z) + u^T Az$$

The dual subgradient method (for maximizing the dual objective) starts with an initial dual guess $u^{(0)}$, and repeats for $k = 1, 2, 3, \ldots$

$$x^{(k)} \in \operatorname*{argmin}_x \; f(x) + (u^{(k-1)})^T Ax$$
$$u^{(k)} = u^{(k-1)} + t_k(Ax^{(k)} - b)$$

Step sizes t_k, $k = 1, 2, 3, \ldots$, are chosen in standard ways

Recall that if f is strictly convex, then f^* is differentiable, and so this becomes dual gradient ascent, which repeats for $k = 1, 2, 3, \ldots$

$$x^{(k)} = \operatorname*{argmin}_x \; f(x) + (u^{(k-1)})^T Ax$$
$$u^{(k)} = u^{(k-1)} + t_k(Ax^{(k)} - b)$$

(Difference is that each $x^{(k)}$ is unique, here.) Again, step sizes t_k, $k = 1, 2, 3, \ldots$ are chosen in standard ways

Lastly, proximal gradients and acceleration can be applied as they would usually

Duality between Strong Convexity and Strong Smoothness

Assume that f is a closed and convex function. Then f is strongly convex with parameter $m \iff \nabla f^*$ Lipschitz with parameter $1/m$

Proof of "$\Longrightarrow$": Recall, if g strongly convex with minimizer x, then

$$g(y) \geq g(x) + \frac{m}{2}\|y - x\|_2, \quad \text{for all } y$$

Hence defining $x_u = \nabla f^*(u)$, $x_v = \nabla f^*(v)$,

$$f(x_v) - u^T x_v \geq f(x_u) - u^T x_u + \frac{m}{2}\|x_u - x_v\|_2^2$$
$$f(x_u) - v^T x_u \geq f(x_v) - v^T x_v + \frac{m}{2}\|x_u - x_v\|_2^2$$

Adding these together, using Cauchy-Schwartz, rearranging shows that $\|x_u - x_v\|_2 \leq \|u - v\|_2/m$

Proof of "$\Longleftarrow$": Exercise!

Convergence guarantees

The following results hold from combining the last fact with what we already know about gradient descent:

- If f is strongly convex with parameter m, then dual gradient ascent with constant step sizes $t_k = m$ converges at sublinear rate $O(1/\epsilon)$
- If f is strongly convex with parameter m and ∇f is Lipschitz with parameter L, then dual gradient ascent with step sizes $t_k = 2/(1/m + 1/L)$ converges at linear rate $O(\log(1/\epsilon))$

Note that these results describe convergence of the dual objective to its optimal value

Dual decomposition

Consider

$$\min_x \sum_{i=1}^B f_i(x_i) \quad \text{subject to} \quad Ax = b$$

Here $x = (x_1, \ldots x_B) \in \mathbb{R}^n$ divides into B blocks of variables, with each $x_i \in \mathbb{R}^{n_i}$. We can also partition A accordingly

$$A = [A_1 \ldots A_B], \quad \text{where } A_i \in \mathbb{R}^{m \times n_i}$$

Simple but powerful observation, in calculation of (sub)gradient, is that the minimization decomposes into B separate problems:

$$x^+ \in \operatorname*{argmin}_x \sum_{i=1}^B f_i(x_i) + u^T Ax$$

$$\iff x_i^+ \in \operatorname*{argmin}_{x_i} f_i(x_i) + u^T A_i x_i, \quad i = 1, \ldots B$$

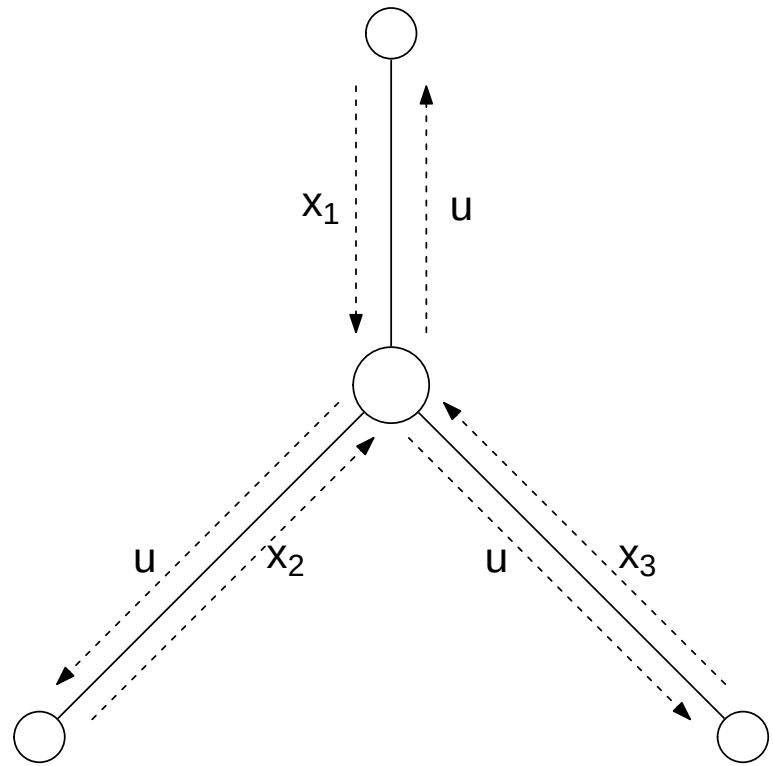
Dual decomposition algorithm: repeat for $k = 1, 2, 3, \ldots$

$$x_i^{(k)} \in \operatorname*{argmin}_{x_i} \; f_i(x_i) + (u^{(k-1)})^T A_i x_i, \quad i = 1, \ldots B$$

$$u^{(k)} = u^{(k-1)} + t_k \left(\sum_{i=1}^{B} A_i x_i^{(k)} - b \right)$$

Can think of these steps as:

- Broadcast: send u to each of the B processors, each optimizes in parallel to find x_i
- Gather: collect $A_i x_i$ from each processor, update the global dual variable u



Dual decomposition with inequality constraints

Consider

$$\min_x \sum_{i=1}^B f_i(x_i) \quad \text{subject to} \quad \sum_{i=1}^B A_i x_i \leq b$$

Dual decomposition, i.e., projected subgradient method:

$$x_i^{(k)} \in \operatorname*{argmin}_{x_i} \; f_i(x_i) + (u^{(k-1)})^T A_i x_i, \quad i = 1, \ldots B$$

$$u^{(k)} = \left(u^{(k-1)} + t_k \left(\sum_{i=1}^B A_i x_i^{(k)} - b \right) \right)_+$$

where u_+ denotes the positive part of u, i.e., $(u_+)_i = \max\{0, u_i\}$, $i = 1, \ldots, m$

Price coordination interpretation (Vandenberghe):

- Have B units in a system, each unit chooses its own decision variable x_i (how to allocate its goods)
- Constraints are limits on shared resources (rows of A), each component of dual variable u_j is price of resource j
- Dual update:

$$u_j^+ = (u_j - ts_j)_+, \quad j = 1, \ldots m$$

  where $s = b - \sum_{i=1}^{B} A_i x_i$ are slacks
  - Increase price u_j if resource j is over-utilized, $s_j < 0$
  - Decrease price u_j if resource j is under-utilized, $s_j > 0$
  - Never let prices get negative

Augmented Lagrangian method

also known as: method of multipliers

Disadvantage of dual ascent: require strong conditions to ensure convergence. Improved by augmented Lagrangian method, also called method of multipliers. We transform the primal problem:

$$\begin{array}{ll} \min\limits_{x} & f(x) + \dfrac{\rho}{2}\|Ax - b\|_2^2 \\ \text{subject to} & Ax = b \end{array}$$

where $\rho > 0$ is a parameter. Clearly equivalent to original problem, and objective is strongly convex when A has full column rank. Use dual gradient ascent:

$$\begin{aligned} x^{(k)} &= \operatorname*{argmin}_x \; f(x) + (u^{(k-1)})^T Ax + \frac{\rho}{2}\|Ax - b\|_2^2 \\ u^{(k)} &= u^{(k-1)} + \rho(Ax^{(k)} - b) \end{aligned}$$

Notice step size choice $t_k = \rho$ in dual algorithm. Why? Since $x^{(k)}$ minimizes $f(x) + (u^{(k-1)})^T Ax + \frac{\rho}{2}\|Ax - b\|_2^2$ over x, we have

$$\begin{aligned} 0 &\in \partial f(x^{(k)}) + A^T\Big(u^{(k-1)} + \rho(Ax^{(k)} - b)\Big) \\ &= \partial f(x^{(k)}) + A^T u^{(k)} \end{aligned}$$

This is the stationarity condition for original primal problem; under mild conditions $Ax^{(k)} - b \to 0$ as $k \to \infty$ (primal iterates become feasible), so KKT conditions are satisfied in the limit and $x^{(k)}, u^{(k)}$ converge to solutions

- Advantage: much better convergence properties
- Disadvantage: lose decomposability! (Separability is ruined by augmented Lagrangian ...)

Alternating direction method of multipliers

Alternating direction method of multipliers or ADMM: try for best of both worlds. Consider the problem

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

As before, we augment the objective

$$\begin{aligned} \min_{x} \quad & f(x) + g(z) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2 \\ \text{subject to} \quad & Ax + Bz = c \end{aligned}$$

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

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

ADMM repeats the steps, for $k = 1, 2, 3, \ldots$

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

Note that the usual method of multipliers would have replaced the first two steps by a joint minimization

$$
(x^{(k)}, z^{(k)}) = \operatorname*{argmin}_{x,z} \; L_\rho(x, z, u^{(k-1)})
$$

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: roughly, ADMM behaves like first-order method. Theory still being developed, see, e.g., in Hong and Luo (2012), Deng and Yin (2012), Iutzeler et al. (2014), Nishihara et al. (2015)

Scaled form ADMM

Scaled form: denote $w = u/\rho$, so augmented Lagrangian becomes

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

and ADMM updates become

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

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

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

Remainder of the lecture

- Examples, practicalities
- Consensus ADMM
- Special decompositions

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):

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

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

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 steps:

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

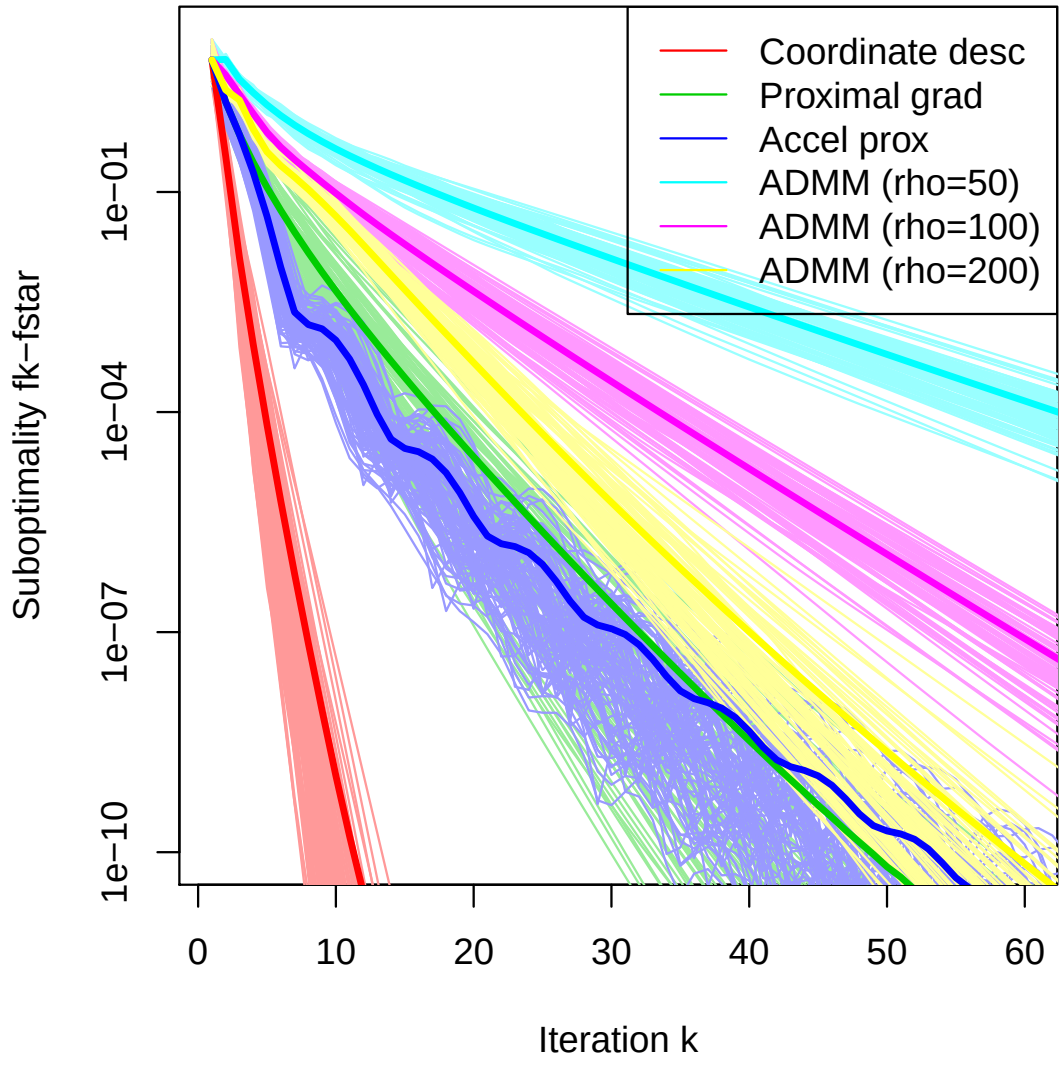
Notes:

- 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: 100 random instances with $n = 200$, $p = 50$



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 (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; it can work well in practice, 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:

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

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

Rewrite as:

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

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 + low rank decomposition

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

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

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:

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

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

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

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

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:

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

This yields the decomposable ADMM updates:

$$x_i^{(k)} = \operatorname*{argmin}_{x_i} f_i(a_i^T x_i + b_i) + \frac{\rho}{2}\|x_i - x^{(k-1)} + w_i^{(k-1)}\|_2^2,$$

$$i = 1,\dots B$$

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

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

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

Special decompositions

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), W., Sharpnack, Tibshirani and Smola (2016)

Example: Trend Filtering

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

where we can construct the discrete difference operators recursively

$$D^{(k+1)} = D^{(1)}D^{(k)}$$

There is an alternative decomposition that results in Fast ADMM updates (Ramdas and Tibshirani, 2014)

$$D^{(k+1)} = D^{(k)}D^{(1)}$$

$$\min_{\theta} \frac{1}{2}\|y-\theta\|_2^2 + \lambda\|D^{(1)}z\|_1 \quad \text{subject to} \quad D^{(k)}\theta = z$$

Generalization possible to Trend Filtering on Graphs! Leverage fast Laplacian solvers for the linear system (Multi-grids methods / Graph-Sparsifiers), and graph-cut (Boykov and Kolmogorov) / parametric maxflow (Champolle an Darbon) for the prox operator.

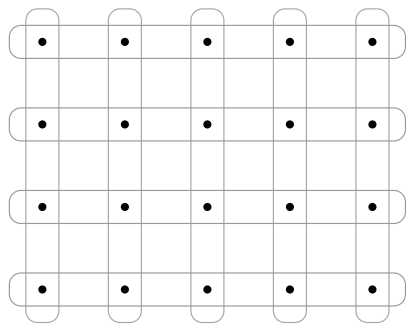
Example: 2d fused lasso

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:

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

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

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 rewrite:

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

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 rewrite:

$$\min_{H,V} \quad \frac{1}{2}\|Y - H\|_F^2 + \lambda \sum_{i,j} \Big(|H_{i,j} - H_{i+1,j}| + |V_{i,j} - V_{i,j+1}| \Big)$$

$$\text{subject to} \;\; H = V$$

Leads to ADMM steps:

$$H_{\cdot,j}^{(k)} = \mathrm{FL}_{\lambda/(1+\rho)}^{\mathrm{1d}} \left(\frac{Y + \rho(V_{\cdot,j}^{(k-1)} - W_{\cdot,j}^{(k-1)})}{1+\rho} \right), \quad j = 1, \ldots, d$$

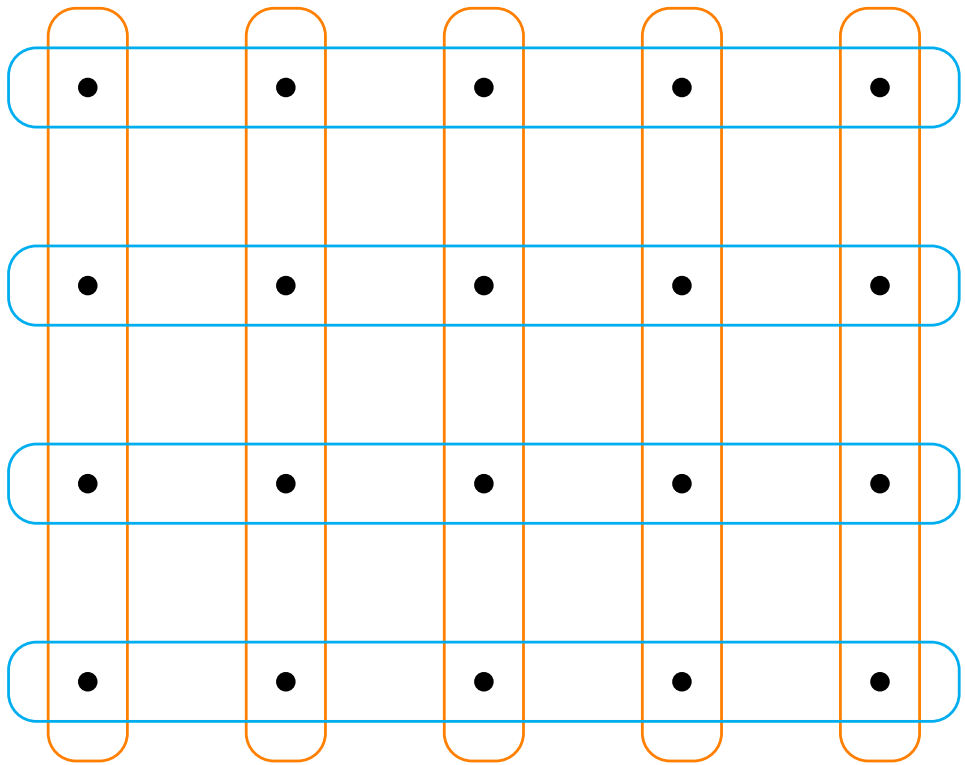
$$V_{i,\cdot}^{(k)} = \mathrm{FL}_{\lambda/\rho}^{\mathrm{1d}} \Big(H_{i,\cdot}^{(k)} + W_{i,\cdot}^{(k-1)} \Big), \quad i = 1, \ldots, d$$

$$W^{(k)} = W^{(k-1)} + H^{(k)} - V^{(k)}$$

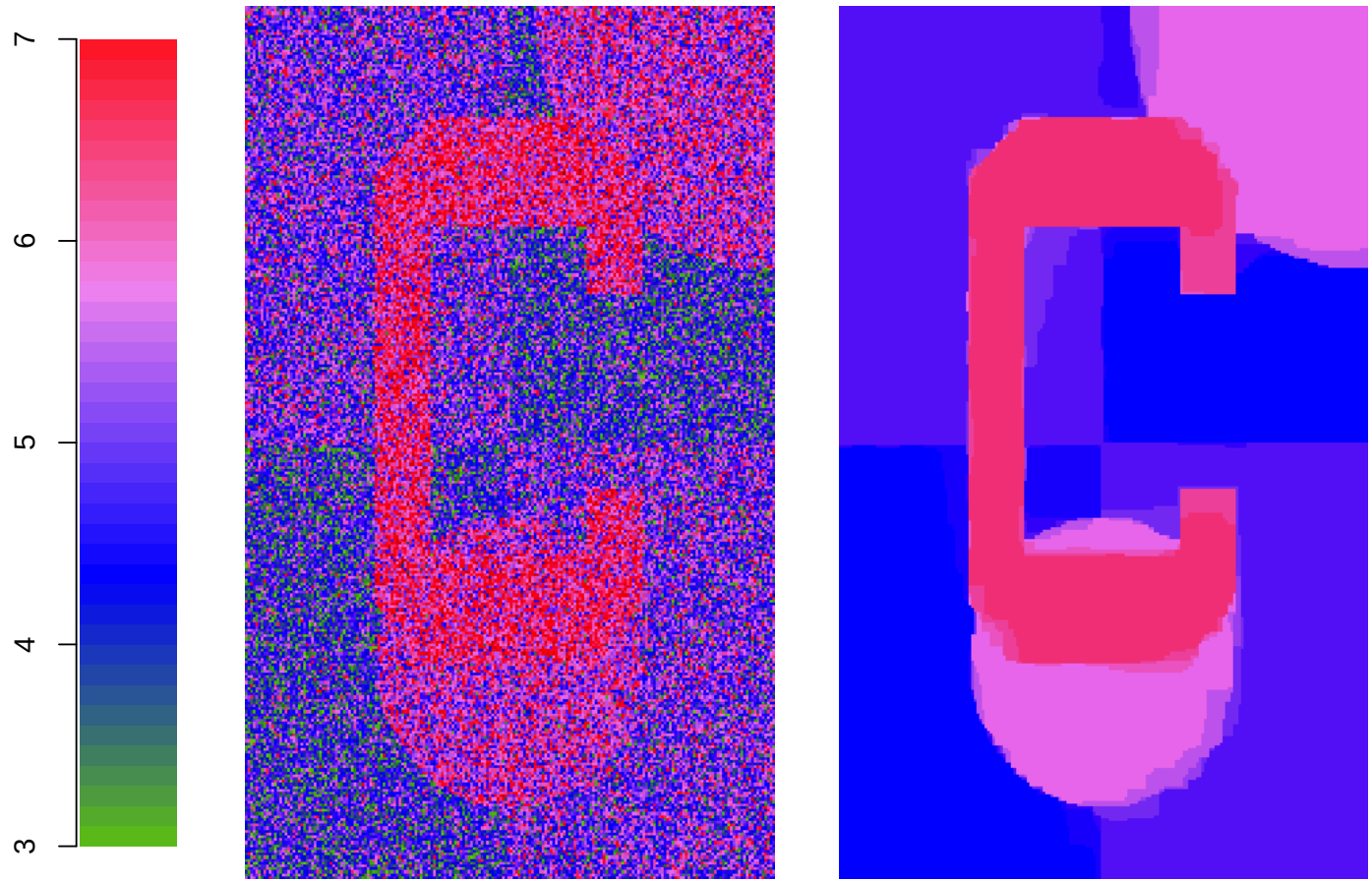
Notes:

- Both H, V 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



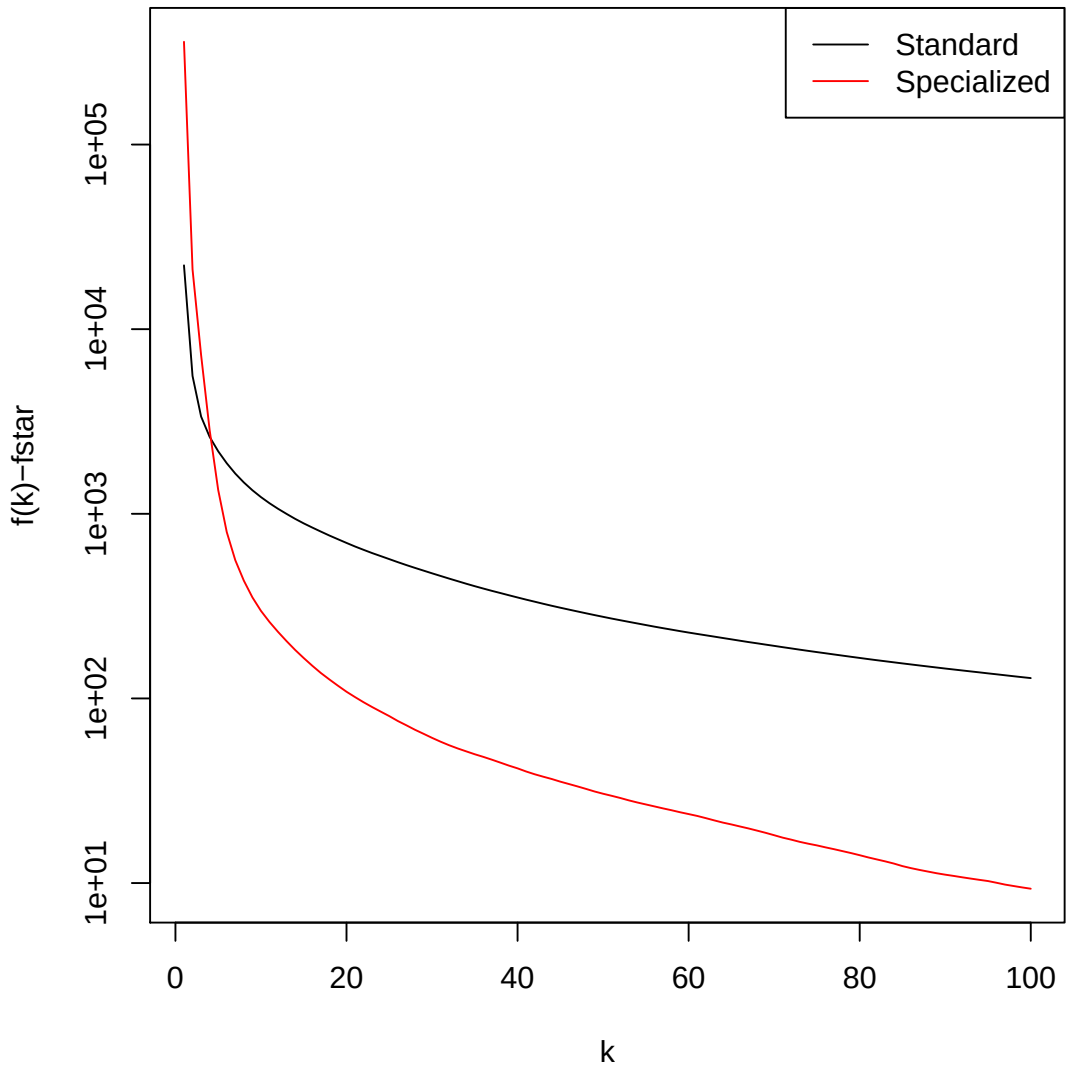
Comparison of 2d fused lasso algorithms: an image of dimension 300×200 (so $n = 60,000$)



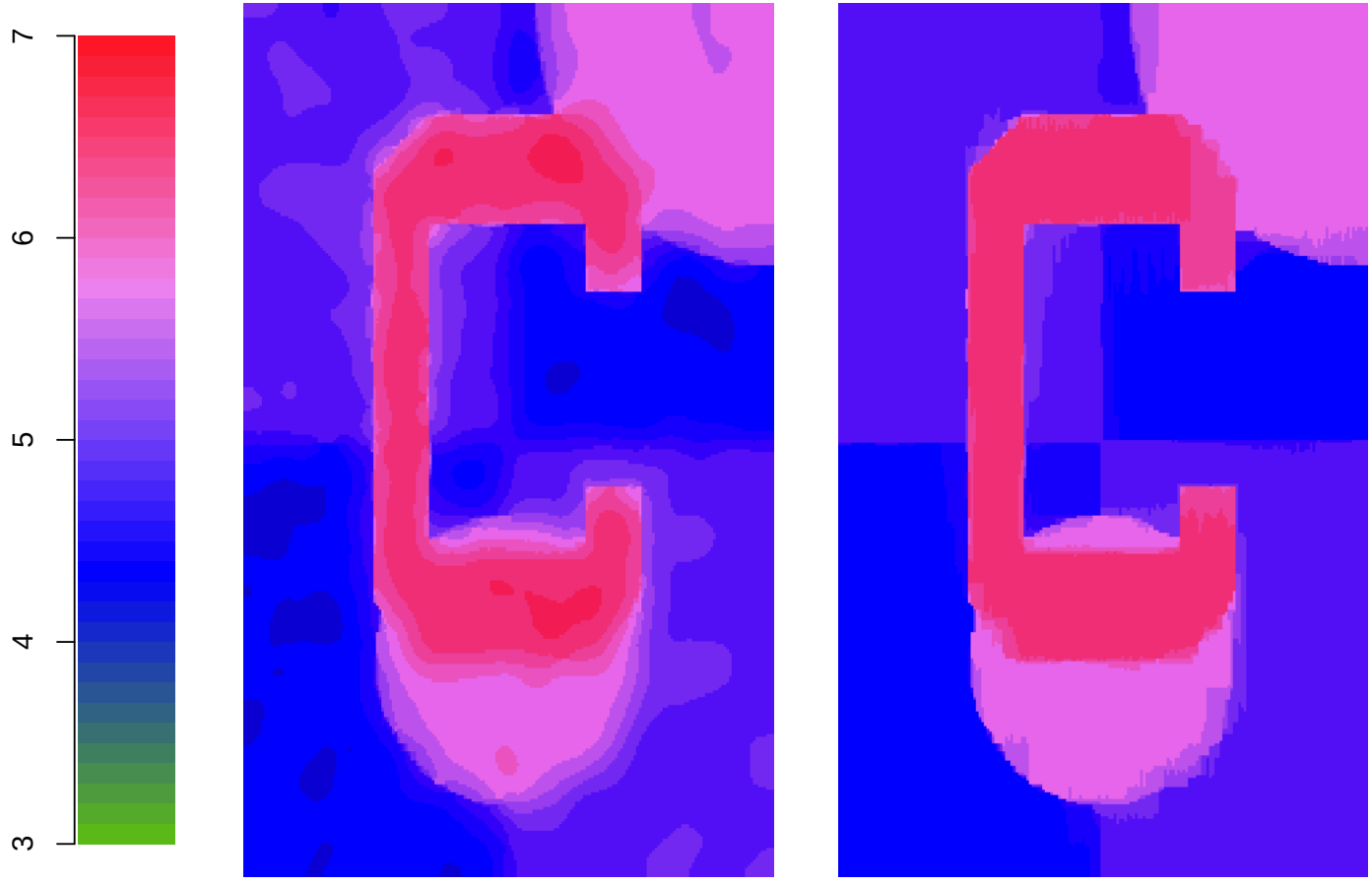
Data

Exact solution

Two ADMM algorithms, (say) standard and specialized ADMM:



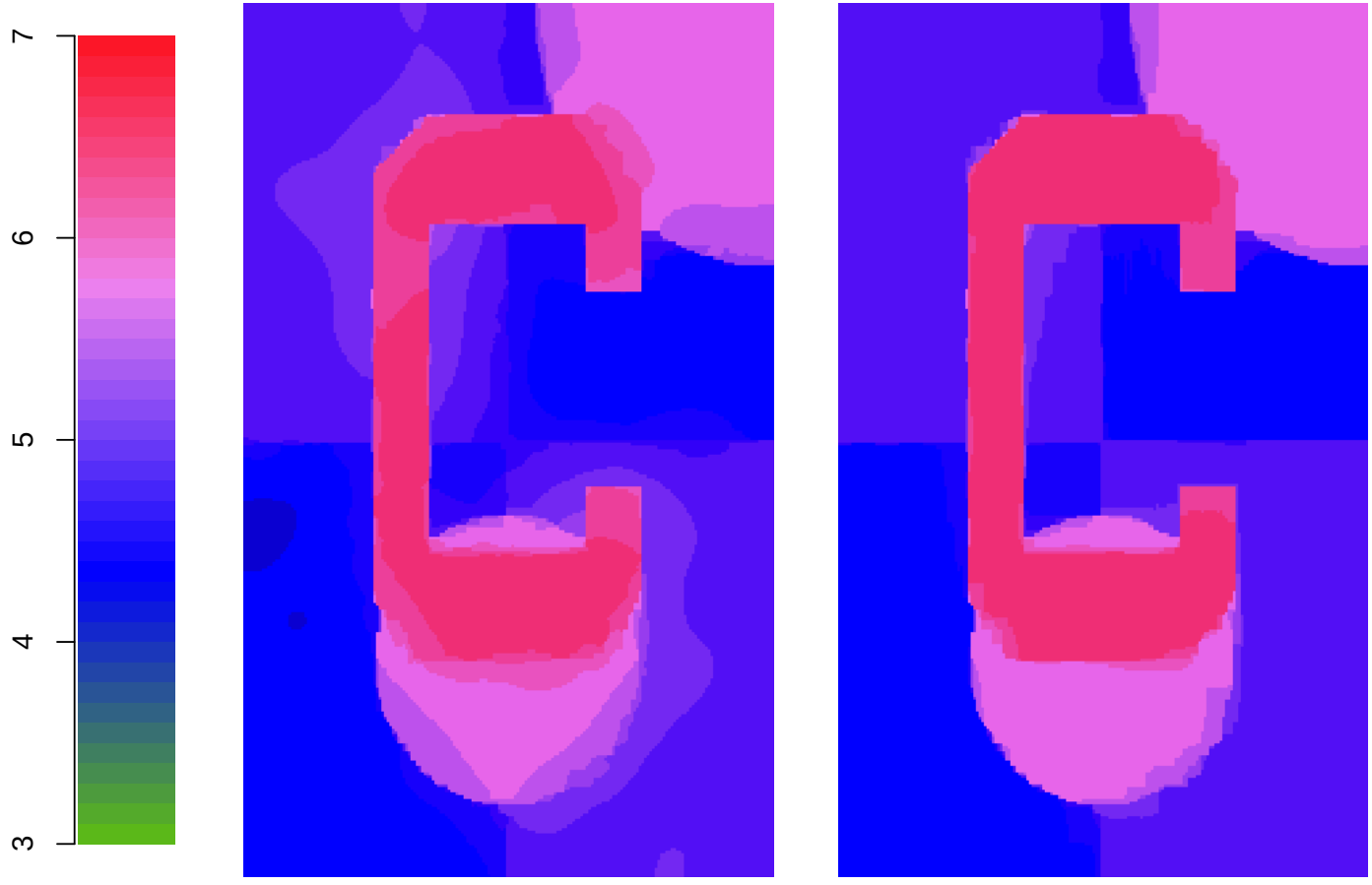
ADMM iterates visualized after $k = 10, 30, 50, 100$ iterations:



Standard ADMM
10 iterations

Specialized ADMM
10 iterations

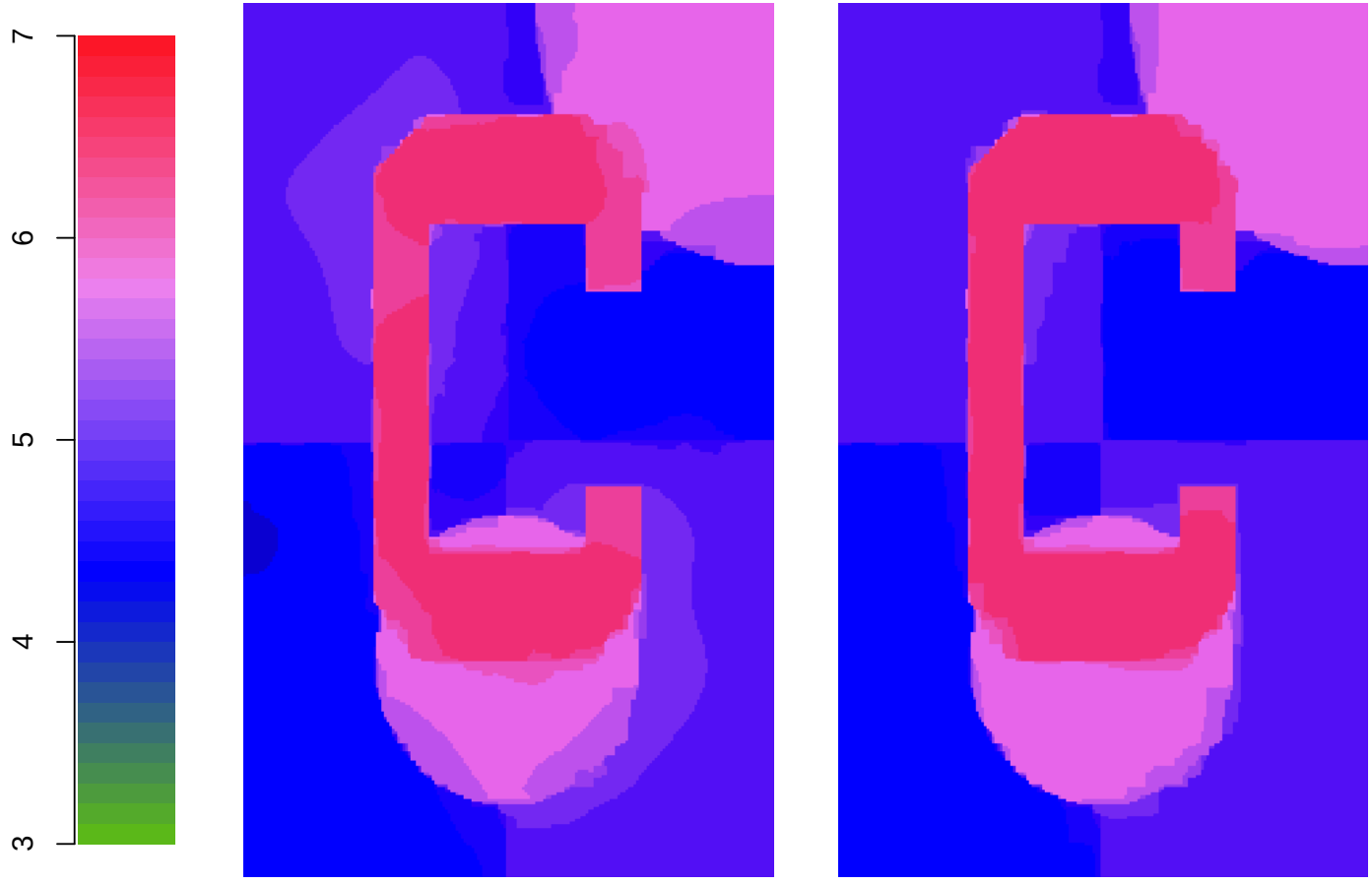
ADMM iterates visualized after $k = 10, 30, 50, 100$ iterations:



Standard ADMM
30 iterations

Specialized ADMM
30 iterations

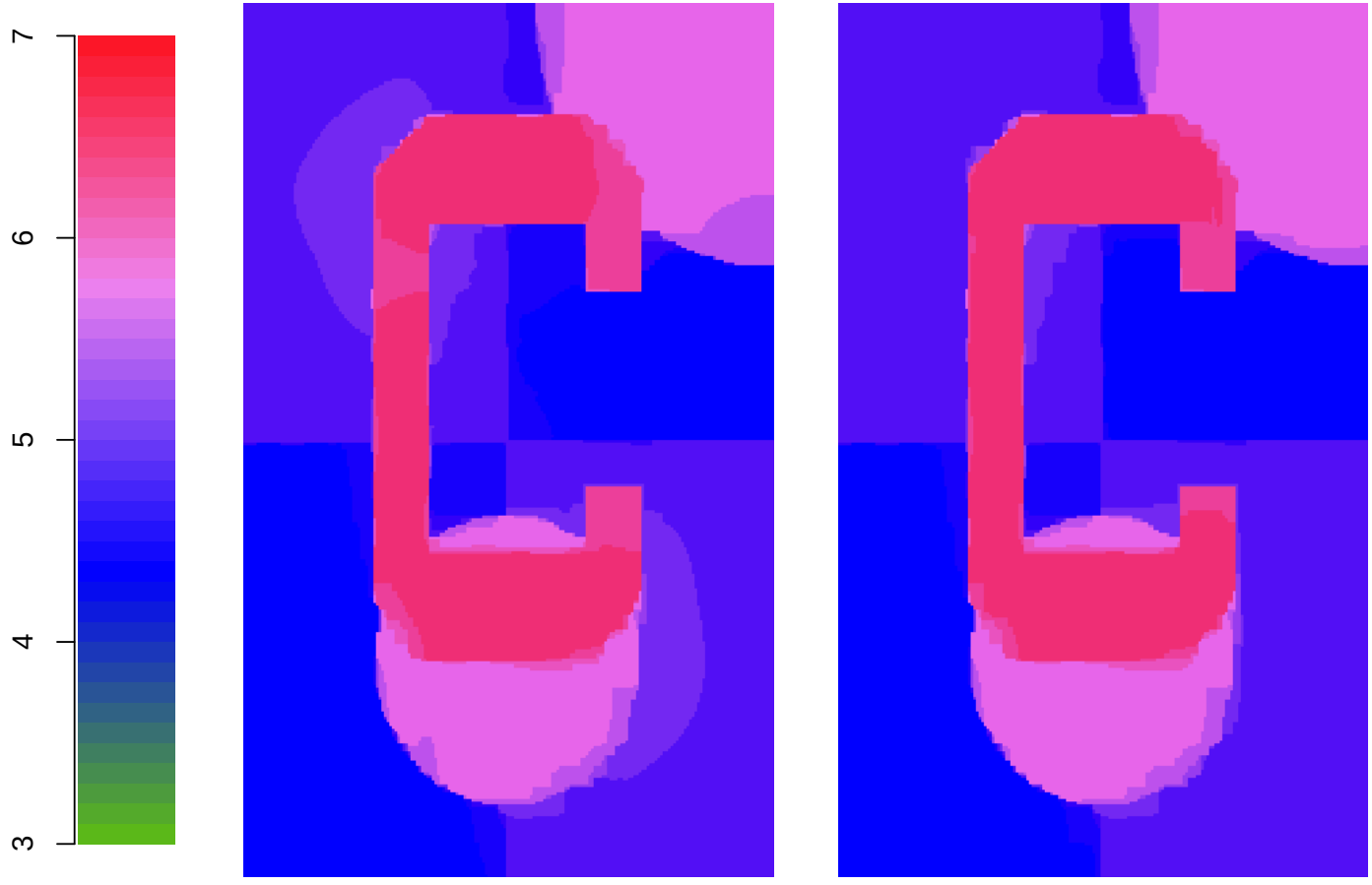
ADMM iterates visualized after $k = 10, 30, 50, 100$ iterations:



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"
- 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"

More references

- A. Johnson (2013). "A dynamic programming algorithm for the fused lasso and l 0-segmentation".
- A. Ramdas, and R. Tibshirani. (2014) "Fast and flexible ADMM algorithms for trend filtering"
- Y.-X. Wang, J. Sharpnack, A. Smola, R. Tibshirani (2016). "Trend filtering on graphs".
- A. Chambolle and J. Darbon (2009)"On total variation minimization and surface evolution using parametric maximum flows"
- D. S. Hochbaum (2001) "An efficient algorithm for image segmentation, Markov random fields and related problems"
- Y. Boykov and V. Kolmogorov (2004) "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision."