# Anderson Accelerated Douglas-Rachford Splitting

Anqi Fu

Department of Electrical Engineering, Stanford University

Joint work with Junzi Zhang and Stephen P. Boyd

EURO 2021: European Conference on Operational Research
Athens, Greece

July 13, 2021

# Overview

# Prox-Affine Optimization Problem

Consider the **prox-affine** form of a convex optimization problem:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^N f_i(x_i) \\ \text{subject to} & \sum_{i=1}^N A_i x_i = b \end{array}$$

with variables $x_i \in \mathbf{R}^{n_i}$ for $i = 1, \ldots, N$.

# Prox-Affine Optimization Problem

Consider the **prox-affine** form of a convex optimization problem:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^N f_i(x_i) \\ \text{subject to} & \sum_{i=1}^N A_i x_i = b \end{array}$$

with variables $x_i \in \mathbf{R}^{n_i}$ for $i = 1, \ldots, N$.

- $A_i \in \mathbf{R}^{m \times n_i}$ and $b \in \mathbf{R}^m$ are given data.
- $f_i : \mathbf{R}^{n_i} \to \mathbf{R} \cup \{+\infty\}$ are closed, convex and proper (CCP).
- Each $f_i$ can only be accessed via its proximal operator

$$\mathbf{prox}_{tf_i}(v_i) = \operatorname{argmin}_{x_i} \left( f_i(x_i) + \frac{1}{2t} \|x_i - v_i\|_2^2 \right),$$

where $t > 0$ is a parameter.

# Prox-Affine Optimization Problem

Why **prox-affine** form?

$$
\begin{array}{ll}
\text{minimize} & \sum_{i=1}^N f_i(x_i) \\
\text{subject to} & \sum_{i=1}^N A_i x_i = b.
\end{array}
$$

# Prox-Affine Optimization Problem

Why **prox-affine** form?

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^N f_i(x_i) \\ \text{subject to} & \sum_{i=1}^N A_i x_i = b. \end{array}$$

- **Generic Formulation**: encompasses many classes of convex problems (cone programs, consensus optimization, etc).

# Prox-Affine Optimization Problem

Why **prox-affine** form?

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^N f_i(x_i) \\ \text{subject to} & \sum_{i=1}^N A_i x_i = b. \end{array}$$

- **Generic Formulation**: encompasses many classes of convex problems (cone programs, consensus optimization, etc).
- **Block Separable**: ideal for parallel/distributed implementation.

# Prox-Affine Optimization Problem

Why **prox-affine** form?

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^N f_i(x_i) \\ \text{subject to} & \sum_{i=1}^N A_i x_i = b. \end{array}$$

- **Generic Formulation**: encompasses many classes of convex problems (cone programs, consensus optimization, etc).
- **Block Separable**: ideal for parallel/distributed implementation.
- **Black Box Proximal Operator**: preserves privacy in peer-to-peer optimization settings.

# Previous Work

- Common methods for distributed optimization:
  - Alternating direction method of multipliers (ADMM).
  - Douglas-Rachford splitting (DRS).
  - Augmented Lagrangian method (ALM).

# Previous Work

- Common methods for distributed optimization:
  - Alternating direction method of multipliers (ADMM).
  - Douglas-Rachford splitting (DRS).
  - Augmented Lagrangian method (ALM).
- These are typically slow to converge, so researchers employ acceleration techniques:
  - Adaptive penalty parameters.
  - Momentum methods.
  - Quasi-Newton/Newton-type method with line search.

# Our Method

A2DR: Anderson acceleration (AA) applied to DRS.

- First type-II AA variant that **converges globally** in non-smooth, potentially pathological settings.
- Produces primal and dual solutions, or a certificate of infeasibility/unboundedness.
- Consistently converges faster with no parameter tuning.
- Memory efficient $\Rightarrow$ little extra cost per iteration.
- Scales to large problems and is easily parallelized.
- Available as an open-source Python solver:

  `https://github.com/cvxgrp/a2dr`

# DRS Algorithm

- Rewrite problem using $\mathcal{I}_S$ as indicator of set $S$:

$$\text{minimize} \quad \overbrace{\sum_{i=1}^{N} f_i(x_i)}^{f(x)} + \overbrace{\mathcal{I}_{Ax=b}(x)}^{g(x)},$$

where $A = [A_1 \ \ldots \ A_n]$ and $x = (x_1, \ldots, x_N)$.

# DRS Algorithm

- Rewrite problem using $\mathcal{I}_S$ as indicator of set $S$:

$$\text{minimize} \quad \overbrace{\sum_{i=1}^{N} f_i(x_i)}^{f(x)} + \overbrace{\mathcal{I}_{Ax=b}(x)}^{g(x)},$$

  where $A = [A_1 \ \ldots \ A_n]$ and $x = (x_1, \ldots, x_N)$.

- DRS iterates for $k = 1, 2, \ldots$,

$$\begin{aligned}
x_i^{k+1/2} &= \mathbf{prox}_{tf_i}(v^k), \quad i = 1, \ldots, N \\
v^{k+1/2} &= 2x^{k+1/2} - v^k \\
x^{k+1} &= \Pi_{Av=b}(v^{k+1/2}) \\
v^{k+1} &= v^k + x^{k+1} - x^{k+1/2}.
\end{aligned}$$

  $\Pi_S(v)$ is Euclidean projection of $v$ onto $S$.

# Convergence of DRS

- DRS iterations can be conceived as a fixed point (FP) mapping

$$v^{k+1} = F(v^k).$$

- $F$ is firmly non-expansive.
- $v^k$ converges to a fixed point of $F$ (if it exists).
- $x^k$ and $x^{k+1/2}$ converge to a solution of our problem.

# Convergence of DRS

- DRS iterations can be conceived as a fixed point (FP) mapping

$$v^{k+1} = F(v^k).$$

- $F$ is firmly non-expansive.
- $v^k$ converges to a fixed point of $F$ (if it exists).
- $x^k$ and $x^{k+1/2}$ converge to a solution of our problem.

In practice, this convergence is often rather slow.

# Type-II AA

- Quasi-Newton method for accelerating fixed point iterations.
- **Extrapolates** next iterate using $M+1$ most recent iterates

$$v^{k+1} = \sum_{j=0}^{M} \alpha_j^k F(v^{k-M+j}).$$

- Let $G(v) = v - F(v)$, then $\alpha^k \in \mathbf{R}^{M+1}$ is solution to

$$\begin{array}{ll} \text{minimize} & \|\sum_{j=0}^{M} \alpha_j^k G(v^{k-M+j})\|_2^2 \\ \text{subject to} & \sum_{j=0}^{M} \alpha_j^k = 1. \end{array}$$

- Typically only need $M \approx 10$ for good performance.

# Adaptive Regularization

- Type-II AA is unstable (Scieur et al, 2016) and can provably diverge (Mai & Johansson 2019).
- Add adaptive regularization term to unconstrained formulation.

# Adaptive Regularization

- Type-II AA is unstable (Scieur et al, 2016) and can provably diverge (Mai & Johansson 2019).
- Add adaptive regularization term to unconstrained formulation.
- Change variables to $\gamma^k \in \mathbf{R}^M$ and define

$$\alpha_0^k = \gamma_0^k, \quad \alpha_i^k = \gamma_i^k - \gamma_{i-1}^k \ \forall i = 1, \ldots, M-1, \quad \alpha_M^k = 1 - \gamma_{M-1}^k.$$

# Adaptive Regularization

- Type-II AA is unstable (Scieur et al, 2016) and can provably diverge (Mai & Johansson 2019).
- Add adaptive regularization term to unconstrained formulation.
- Change variables to $\gamma^k \in \mathbf{R}^M$ and define

$$\alpha_0^k = \gamma_0^k, \quad \alpha_i^k = \gamma_i^k - \gamma_{i-1}^k \ \forall i = 1, \dots, M-1, \quad \alpha_M^k = 1 - \gamma_{M-1}^k.$$

- Unconstrained AA problem:

$$\text{minimize} \quad \|g^k - Y_k \gamma^k\|_2^2,$$

where we define

$$g^k = G(v^k), \quad y^k = g^{k+1} - g^k, \quad Y_k = [y^{k-M} \ \dots \ y^{k-1}].$$

# Adaptive Regularization

- Type-II AA is unstable (Scieur et al, 2016) and can provably diverge (Mai & Johansson 2019).
- Add adaptive regularization term to unconstrained formulation.
- Change variables to $\gamma^k \in \mathbf{R}^M$ and define

$$\alpha_0^k = \gamma_0^k, \quad \alpha_i^k = \gamma_i^k - \gamma_{i-1}^k \ \forall i = 1, \ldots, M-1, \quad \alpha_M^k = 1 - \gamma_{M-1}^k.$$

- Stabilized AA problem with quadratic regularization:

$$\text{minimize} \quad \|g^k - Y_k \gamma^k\|_2^2 + \eta \left( \|S_k\|_F^2 + \|Y_k\|_F^2 \right) \|\gamma^k\|_2^2,$$

where $\eta \geq 0$ is a parameter and

$$g^k = G(v^k), \quad y^k = g^{k+1} - g^k, \quad Y_k = [y^{k-M} \ \ldots \ y^{k-1}],$$
$$s^k = v^{k+1} - v^k, \quad S_k = [s^{k-M} \ \ldots \ s^{k-1}].$$

# A2DR

- Let $\epsilon > 0$, $M$ positive integer.
- A2DR iterates for $k = 1, 2, \ldots,$

1. Compute $v_{\text{DRS}}^{k+1} = F(v^k), \quad g^k = v^k - v_{\text{DRS}}^{k+1}$.
2. Solve stabilized AA problem for $\gamma^k \Rightarrow$ calculate $\alpha^k$.
3. Compute $v_{\text{AA}}^{k+1} = \sum_{j=0}^{M} \alpha_j^k v_{\text{DRS}}^{k-M+j+1}$.
4. Safeguard: If residual $\|G(v^k)\|_2 = O(1/n_{\text{AA}}^{1+\epsilon})$, adopt

$$v^{k+i} = v_{\text{AA}}^{k+i} \text{ for } i = 1, \ldots, M,$$

where $n_{\text{AA}} = \#$ of adopted AA candidates.

Otherwise, take $v^{k+1} = v_{\text{DRS}}^{k+1}$.

(This step ensures convergence in pathological cases).

# Stopping Criterion of A2DR

- Stop and output $x^{k+1/2}$ when $\|r^k\|_2 \leq \epsilon_{\text{tol}}$:

$$\begin{aligned} r^k_{\text{prim}} &= Ax^{k+1/2} - b, \\ r^k_{\text{dual}} &= \tfrac{1}{t}(v^k - x^{k+1/2}) + A^T\lambda^k, \\ r^k &= (r^k_{\text{prim}}, r^k_{\text{dual}}). \end{aligned}$$

- Dual variable is minimizer of dual residual norm

$$\lambda^k = \text{argmin}_\lambda \ \|\tfrac{1}{t}(v^k - x^{k+1/2}) + A^T\lambda\|_2,$$

which we obtain by solving a simple least-squares problem.

# Stopping Criterion of A2DR

- Stop and output $x^{k+1/2}$ when $\|r^k\|_2 \leq \epsilon_{\text{tol}}$:

$$
\begin{aligned}
r^k_{\text{prim}} &= Ax^{k+1/2} - b, \\
r^k_{\text{dual}} &= \tfrac{1}{t}(v^k - x^{k+1/2}) + A^T \lambda^k, \\
r^k &= (r^k_{\text{prim}}, r^k_{\text{dual}}).
\end{aligned}
$$

- Dual variable is minimizer of dual residual norm

$$
\lambda^k = \text{argmin}_\lambda \ \|\tfrac{1}{t}(v^k - x^{k+1/2}) + A^T \lambda\|_2,
$$

which we obtain by solving a simple least-squares problem.

- Notice we get a proximal point $\frac{1}{t}(v^k - x^{k+1/2}) \in \partial f(x^{k+1/2})$.

**Theorem (Solvable Case)**

*If the problem is solvable (e.g., feasible and bounded), then*

$$\liminf_{k\to\infty} \|r^k\|_2 = 0$$

*and the AA candidates are adopted infinitely often. Furthermore, if $F$ has a fixed point, then*

$$\lim_{k\to\infty} v^k = v^\star \text{ and } \lim_{k\to\infty} x^{k+1/2} = x^\star,$$

*where $v^\star$ is a fixed-point of $F$ and $x^\star$ is a solution to our problem.*

# Convergence of A2DR: Theorems

## Theorem (Pathological Case)

*If the problem is pathological (strongly primal infeasible or strongly dual infeasible), then*

$$\lim_{k\to\infty}\left(v^k - v^{k+1}\right) = \delta v \neq 0.$$

*Furthermore, if* $\lim_{k\to\infty} Ax^{k+1/2} = b$*, then the problem is unbounded and* $\|\delta v\|_2 = t\,\mathrm{dist}(\mathrm{dom}\, f^*, \mathrm{range}(A^T))$*. Otherwise, it is infeasible and* $\|\delta v\|_2 \geq \mathrm{dist}(\mathrm{dom}\, f, \{x : Ax = b\})$ *with equality when the dual problem is feasible.*

# Preconditioning

- Convergence greatly improved by rescaling problem.
- Replace original $A$, $b$, $f_i$ with

$$\hat{A} = DAE, \quad \hat{b} = Db, \quad \hat{f}_i(\hat{x}_i) = f_i(e_i \hat{x}_i).$$

- $D$ and $E$ are diagonal positive ($e_i > 0$ corresponds to $i$th block diagonal entry of $E$) and chosen by equilibrating $A$.
- Proximal operator of $\hat{f}_i$ can be evaluated using proximal operator of $f_i$

$$\mathbf{prox}_{t\hat{f}_i}(\hat{v}_i) = \frac{1}{e_i}\mathbf{prox}_{(e_i^2 t)f_i}(e_i \hat{v}_i).$$

- Stopping criterion checked on rescaled problem.

# a2dr Solver Interface

```
result = a2dr(p_list, A_list, b)
```

Input arguments:

- p_list is list of function handles for $\mathbf{prox}_{tf_i}(v_i)$, e.g.,

$$f_i(x_i) = x_i \;\Rightarrow\; \texttt{p\_list[i] = lambda v,t: v - t}$$

- A_list is list of matrices $A_i$, b is vector $b$.

Output dictionary keys:

- solve_time is total runtime.
- num_iters is total number of iterations $K$.
- x_vals is list of final values $x_i^K$.
- primal and dual are vectors of $r_{\text{prim}}^k$ and $r_{\text{dual}}^k$ for $k = 1, \ldots, K$.

# Nonnegative Least Squares (NNLS)

$$\begin{array}{ll} \text{minimize} & \|Fz - g\|_2^2 \\ \text{subject to} & z \geq 0 \end{array}$$

with respect to $z \in \mathbf{R}^q$.

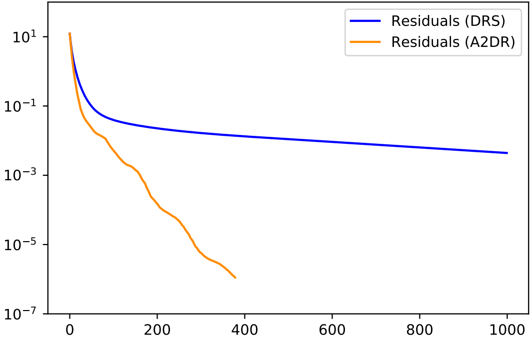
- Problem data: $F \in \mathbf{R}^{p \times q}$ and $g \in \mathbf{R}^p$.
- Can be written in prox-affine form with

$$f_1(x_1) = \|Fx_1 - g\|_2^2, \quad f_2(x_2) = \mathcal{I}_{\mathrm{R}_+^n}(x_2),$$
$$A_1 = I, \quad A_2 = -I, \quad b = 0.$$

- We evaluate the proximal operator of $f_1$ using LSQR.

# NNLS: Convergence of $\|r^k\|_2$

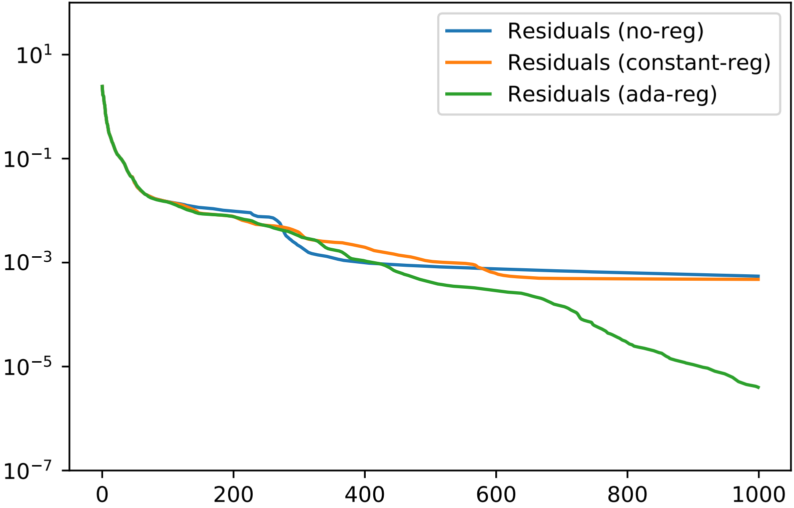
$p = 10^4$, $q = 8000$, $F$ has 0.1% nonzeros




A2DR took only 55 seconds, while OSQP and SCS took respectively 349 and 327 seconds.

# NNLS: Effect of Regularization on $\|r^k\|_2$

$p = 300,\ q = 500,\ F$ has 0.1% nonzeros



# Sparse Inverse Covariance Estimation

- Samples $z_1, \dots, z_p$ IID from $\mathcal{N}(0, \Sigma)$.
- Know covariance $\Sigma \in \mathbf{S}_+^q$ has **sparse** inverse $S = \Sigma^{-1}$.
- One way to estimate $S$ is by solving the penalized log-likelihood problem

$$\text{minimize} \quad -\log\det(S) + \mathrm{tr}(SQ) + \alpha \textstyle\sum_{i,j} |S_{ij}|,$$

  where $Q$ is the sample covariance, $\alpha \geq 0$ is a parameter.
- Note $\log\det(S) = -\infty$ when $S \not\succ 0$.

# Sparse Inverse Covariance Estimation

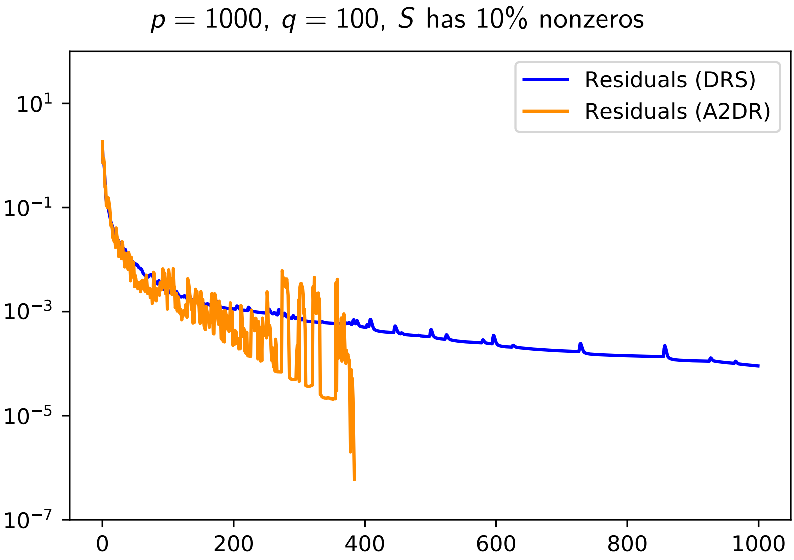
- Problem can be written in prox-affine form with

$$f_1(S_1) = -\log\det(S_1) + \operatorname{tr}(S_1 Q), \quad f_2(S_2) = \alpha \sum_{i,j} |(S_2)_{ij}|,$$

$$A_1 = I, \quad A_2 = -I, \quad b = 0.$$

- Both proximal operators have closed-form solutions.

# Covariance Estimation: Convergence of $\|r^k\|_2$



# Covariance Estimation: Larger Examples

Compared performance between A2DR and SCS on large $S$ with vectorizations of $O(10^6)$.

- $q = 1200$: A2DR took 1 hour to converge to a tolerance of $10^{-3}$, while SCS took 11 hours to achieve a tolerance of $10^{-1}$ and yielded a much worse objective value.
- $q = 2000$: A2DR converged in 2.6 hours to a tolerance of $10^{-3}$, while SCS failed immediately with an out-of-memory error.

# Multi-Task Logistic Regression

$$\text{minimize} \quad \phi(W\theta, Y) + \alpha \sum_{l=1}^{L} \|\theta_l\|_2 + \beta\|\theta\|_*$$

with respect to $\theta = [\theta_1 \cdots \theta_L] \in \mathbf{R}^{s \times L}$.

- Problem data: $W \in \mathbf{R}^{p \times s}$ and $Y = [y_1 \cdots y_L] \in \mathbf{R}^{p \times L}$.
- Regularization parameters: $\alpha \geq 0, \beta \geq 0$.
- Logistic loss function

$$\phi(Z, Y) = \sum_{l=1}^{L} \sum_{i=1}^{p} \log\left(1 + \exp(-Y_{il} Z_{il})\right).$$

# Multi-Task Logistic Regression

- Rewrite problem in prox-affine form with

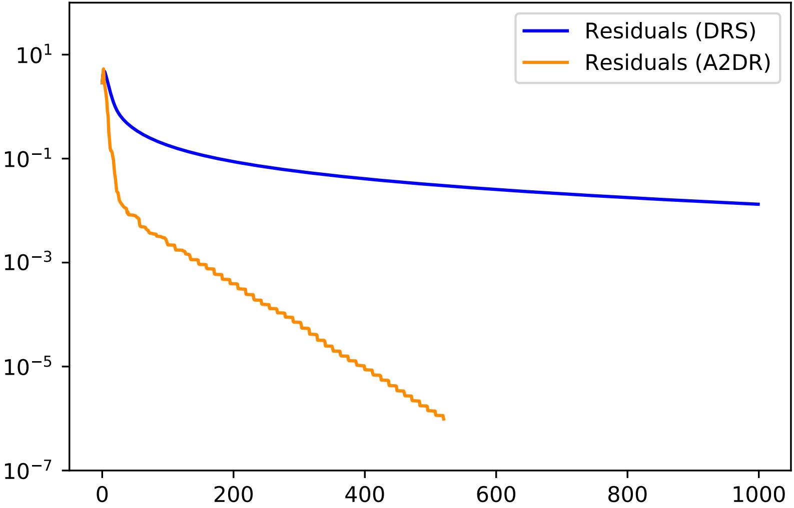
$$f_1(Z) = \phi(Z, Y), \quad f_2(\theta) = \alpha \sum_{l=1}^{L} \|\theta_l\|_2, \quad f_3(\tilde{\theta}) = \beta \|\tilde{\theta}\|_*,$$

$$A = \begin{bmatrix} I & -W & 0 \\ 0 & I & -I \end{bmatrix}, \quad x = \begin{bmatrix} Z \\ \theta \\ \tilde{\theta} \end{bmatrix}, \quad b = 0.$$

- We evaluate the proximal operator of $f_1$ using the Newton-CG method, and the rest with closed-form formulae.

# Multi-Task Logistic: Convergence of $\|r^k\|_2$

$p = 300,\ s = 500,\ L = 10,\ \alpha = \beta = 0.1$



# Other Examples

A2DR can be applied to many other problems (see paper for details):

- $l_1$ trend filtering.
- Stratified models.
- Single commodity flow optimization.
- Optimal control.
- Coupled quadratic program.

# Conclusion

- A2DR is a fast, robust algorithm for solving generic convex optimization problems in prox-affine form.
- Highly scalable, parallelizable, and memory-efficient.
- Consistently fast convergence with no parameter tuning.
- Produces primal and dual solutions, or a certificate of infeasibility/unboundedness.
- Open-source Python library:

  `https://github.com/cvxgrp/a2dr`

# References

A. Fu.*, J. Zhang*, S. Boyd. "Anderson Accelerated Douglas-Rachford Splitting." *SIAM Journal on Scientific Computing*, vol. 42 (6): A3560–A3583, November 2020.
(*equal contribution)