Distributed Optimization over Network

Wotao Yin (UCLA, Math Department)

2014 IEEE SAM

Goal of this talk

Build optimization algorithms that run on networks from basic operators:

- forward operator: $\mathbf{fwd}_f := (I - \nabla f)$
- backward operator: $\mathbf{prox}_f$ (define later)
- reflection operator: $\mathbf{refl}_f := \mathbf{prox}_f + (\mathbf{prox}_f - I)$
- averaging operator: W where $W\mathbf{1} = \mathbf{1}$.

Goal of this talk

Build optimization algorithms that run on networks from basic operators:

- forward operator: $\mathbf{fwd}_f := (I - \nabla f)$
- backward operator: $\mathbf{prox}_f$ (define later)
- reflection operator: $\mathbf{refl}_f := \mathbf{prox}_f + (\mathbf{prox}_f - I)$
- averaging operator: W where $W\mathbf{1} = \mathbf{1}$.

We do not cover

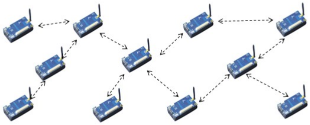
- Nonconvex optimization
- Asynchronous computation or communication
- Dynamic topology, control problem.

Roughly speaking

- **first-order algorithms are simple**
- **convergence requires very few conditions**
- **convergence rates can be derived**
- combined with **duality** and **splitting**, they are **very versatile:**
  - as simple as gradient descent and alternating projection
  - but also handles complicated objective terms and constraints
  - give rise to parallel, distributed, decentralized algorithms
- **focus: decentralized consensus**

Consensus optimization

- A connected network of n agents



- Each agent i has function f_i
- Find a *consensus solution* $x^* \in \mathbb{R}^p$ to

$$\underset{x \in \mathbb{R}^p}{\text{minimize}}\, f(x) := \sum_{i=1}^{n} f_i(x)$$

For analysis, define $\bar{f}(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x)$.

Existing decentralized approaches

- **(sub)gradient descent**: Nedic-Ozdaglar'09, diminishing step-size by Jakovetic-Xavier-Moura'13, fixed step-size by Yuan-Ling-Y.'13

Existing decentralized approaches

- **(sub)gradient descent**: Nedic-Ozdaglar'09, diminishing step-size by Jakovetic-Xavier-Moura'13, fixed step-size by Yuan-Ling-Y.'13
- **Decentralized ADMM**: Bertsekas-Tsitsiklis'97, Giannakis et al, Schizas et al'08, linear convergence Shi-Ling-Y.'13

Existing decentralized approaches

- **(sub)gradient descent**: Nedic-Ozdaglar'09, diminishing step-size by Jakovetic-Xavier-Moura'13, fixed step-size by Yuan-Ling-Y.'13
- **Decentralized ADMM**: Bertsekas-Tsitsiklis'97, Giannakis et al, Schizas et al'08, linear convergence Shi-Ling-Y.'13
- Related to **gossip algorithms** (Tsitsiklis et al'86, Boyd et al'06) and **diffusion algorithms** (Lopes-Sayed'08, Tkahashi-Yamada'10)

Existing decentralized approaches

- **(sub)gradient descent**: Nedic-Ozdaglar'09, diminishing step-size by Jakovetic-Xavier-Moura'13, fixed step-size by Yuan-Ling-Y.'13
- **Decentralized ADMM**: Bertsekas-Tsitsiklis'97, Giannakis et al, Schizas et al'08, linear convergence Shi-Ling-Y.'13
- Related to **gossip algorithms** (Tsitsiklis et al'86, Boyd et al'06) and **diffusion algorithms** (Lopes-Sayed'08, Tkahashi-Yamada'10)
- **Belief propagation** (Cetin et al'06)
- **Incremental optimization** (Rabbat et al'04)
- ... more ...

Compact notation

- Each node i has variable $x_{(i)} \in \mathbb{R}^p$, placed on the ith row of $\mathbf{x}$.

$$\mathbf{x} \triangleq \begin{pmatrix} — & x_{(1)}^{\mathrm{T}} & — \\ — & x_{(2)}^{\mathrm{T}} & — \\ & \vdots & \\ — & x_{(n)}^{\mathrm{T}} & — \end{pmatrix} \in \mathbb{R}^{n \times p}$$

- $\mathbf{x}$ is consensus if all rows are equal: $x_{(i)}^T = x_{(j)}^T,\ \forall i \neq j$.

Compact notation

- Each node i has variable $x_{(i)} \in \mathbb{R}^p$, placed on the ith row of $\mathbf{x}$.

$$\mathbf{x} \triangleq \begin{pmatrix} — & x_{(1)}^{\mathrm{T}} & — \\ — & x_{(2)}^{\mathrm{T}} & — \\ & \vdots & \\ — & x_{(n)}^{\mathrm{T}} & — \end{pmatrix} \in \mathbb{R}^{n \times p}$$

- $\mathbf{x}$ is consensus if all rows are equal: $x_{(i)}^T = x_{(j)}^T, \ \forall i \neq j$.

$$\mathbf{f}(\mathbf{x}) \triangleq \begin{pmatrix} f(x_{(1)}) \\ f(x_{(2)}) \\ \vdots \\ f(x_{(n)}) \end{pmatrix} \in \mathbb{R}^n, \quad \nabla \mathbf{f}(\mathbf{x}) \triangleq \begin{pmatrix} — & \nabla f_1(x_{(1)})^{\mathrm{T}} & — \\ — & \nabla f_2(x_{(2)})^{\mathrm{T}} & — \\ & \vdots & \\ — & \nabla f_n(x_{(n)})^{\mathrm{T}} & — \end{pmatrix} \in \mathbb{R}^{n \times p}.$$

- original problem $\Longleftrightarrow$

$$\text{minimize } \mathbf{1}^T \mathbf{f}(\mathbf{x}), \text{ subject to } x_{(i)} = x_{(j)}, \ \forall i \neq j.$$

Decentralized gradient descent (DGD)

Nedic-Ozdaglar'09:

- average in a neighborhood
- apply an individual gradient descent

Decentralized gradient descent (DGD)

Nedic-Ozdaglar'09:

- average in a neighborhood
- apply an individual gradient descent

$$x_{(i)}^{k+1} = \sum_j w_{ij} x_{(j)}^k - \alpha \nabla f_i(x_{(i)}^k), \quad \text{by agents } i = 1, 2, \ldots, n.$$

Decentralized gradient descent (DGD)

Nedic-Ozdaglar'09:

- average in a neighborhood
- apply an individual gradient descent

$$x_{(i)}^{k+1} = \sum_j w_{ij} x_{(j)}^k - \alpha \nabla f_i(x_{(i)}^k), \quad \text{by agents } i = 1, 2, \ldots, n.$$

$$\text{Compact form:} \quad \boxed{\mathbf{x}^{k+1} = W\mathbf{x}^k - \alpha \nabla \mathbf{f}(\mathbf{x}^k)}$$

Decentralized gradient descent (DGD)

Nedic-Ozdaglar'09:

- average in a neighborhood
- apply an individual gradient descent

$$x_{(i)}^{k+1} = \sum_j w_{ij} x_{(j)}^k - \alpha \nabla f_i(x_{(i)}^k), \quad \text{by agents } i = 1, 2, \ldots, n.$$

Compact form: $\boxed{\mathbf{x}^{k+1} = W\mathbf{x}^k - \alpha \nabla \mathbf{f}(\mathbf{x}^k)}$

This talk assumes synchronous and fixed topology, relaxed in practice.

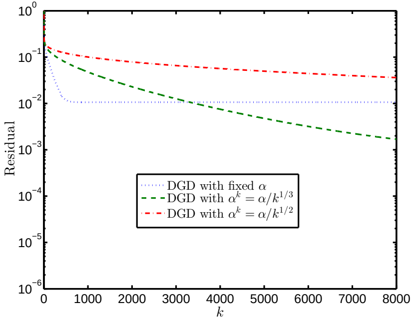
Matrix $W = [w_{ij}]$ is the *mixing matrix*:

- $w_{ij} = 0,\ i \neq j$, if nodes i and j are not neighbors
- **assumption:** *symmetric, doubly stochastic*

$$W = W^T,\ W\mathbf{1} = \mathbf{1},\ \mathbf{1}^T W = \mathbf{1}^T.$$

Example: decentralized least-squares

fixed v.s. diminishing step size



Fixed step size: quick but will stall; too large α causes divergence

Diminishing step size: slower but converges to consensus solution x^*

- $\alpha/k^{1/3}$: Jakovetic-Xavier-Moura'14
- $\alpha/k^{1/2}$: I-An Chen'12

DGD:

$$\mathbf{x}^{k+1} = W\mathbf{x}^{k} - \alpha\nabla\mathbf{f}(\mathbf{x}^{k}).$$

DGD:

$$\mathbf{x}^{k+1} = W\mathbf{x}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k).$$

Interpretation 1: unit-step gradient descent iteration

$$\mathbf{x}^{k+1} = (I - \nabla\xi_\alpha)(\mathbf{x}^k)$$

DGD:

$$\mathbf{x}^{k+1} = W\mathbf{x}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k).$$

Interpretation 1: unit-step gradient descent iteration

$$\mathbf{x}^{k+1} = (I - \nabla\xi_\alpha)(\mathbf{x}^k)$$

applied to the Lyapunov function

$$\xi_\alpha(\mathbf{x}) := \frac{1}{2}\mathrm{tr}(\mathbf{x}^T(I - W)\mathbf{x}) + \alpha\mathbf{1}^T\mathbf{f}(\mathbf{x}).$$

DGD:

$$\mathbf{x}^{k+1} = W\mathbf{x}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k).$$

Interpretation 1: unit-step gradient descent iteration

$$\mathbf{x}^{k+1} = (I - \nabla\xi_\alpha)(\mathbf{x}^k)$$

applied to the Lyapunov function

$$\xi_\alpha(\mathbf{x}) := \frac{1}{2}\mathrm{tr}(\mathbf{x}^T(I - W)\mathbf{x}) + \alpha\mathbf{1}^T\mathbf{f}(\mathbf{x}).$$

Interpretation 2: inexact gradient descent applied to

$$\min_{\bar{x}} \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\bar{x}).$$

DGD:

$$\mathbf{x}^{k+1} = W\mathbf{x}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k).$$

Interpretation 1: unit-step gradient descent iteration

$$\mathbf{x}^{k+1} = (I - \nabla\xi_\alpha)(\mathbf{x}^k)$$

applied to the Lyapunov function

$$\xi_\alpha(\mathbf{x}) := \frac{1}{2}\mathrm{tr}(\mathbf{x}^T(I - W)\mathbf{x}) + \alpha\mathbf{1}^T\mathbf{f}(\mathbf{x}).$$

Interpretation 2: inexact gradient descent applied to

$$\min_{\bar{x}} \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\bar{x}).$$

Reason: multiply $\frac{1}{n}\mathbf{1}^T \times (\text{DGD formula})$:

$$\bar{x}^{k+1} = \bar{x}^k - \alpha\left[\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_{(i)}^k)\right].$$

New results (with K.Yuan and Q.Ling)

- **Assume** ∇f_i is L_i-Lipschitz, and $\alpha \leq (1+\lambda_n(W))/\max_i L_i$
- Proved boundedness of everything and convergence (not to right solution)
  (the bound is tight; counterexamples exist if it is voided)
  (dropped boundedness assumptions on ∇f_i from previous work)
- Bounded deviation from mean $\sim O(\frac{\alpha}{1-\beta})$, where β is 2nd largest absolute eigenvalue of W
- Objective error $\sim O(\frac{1}{\alpha k})$ until reaching $O(\frac{\alpha}{1-\beta})$
- If all f_i are strongly convex, objective and point errors converge *linearly* until reaching $O(\frac{\alpha}{1-\beta})$

New results (with K.Yuan and Q.Ling)

- **Assume** ∇f_i is L_i-Lipschitz, and $\alpha \leq (1 + \lambda_n(W))/\max_i L_i$
- Proved boundedness of everything and convergence (not to right solution)
  (the bound is tight; counterexamples exist if it is voided)
  (dropped boundedness assumptions on ∇f_i from previous work)
- Bounded deviation from mean $\sim O(\frac{\alpha}{1-\beta})$, where β is 2nd largest absolute eigenvalue of W
- Objective error $\sim O(\frac{1}{\alpha k})$ until reaching $O(\frac{\alpha}{1-\beta})$
- If all f_i are strongly convex, objective and point errors converge *linearly* until reaching $O(\frac{\alpha}{1-\beta})$

Take-home: DGD performs just like (centralized) gradient descent, except

- spectra of W affects speed and final accuracy
- small α: slow and accurate
- large α: fast and inaccurate
- decreasing α: even slower but exact

Speed-exactness dilemma

DGD iteration:

$$\mathbf{x}^{k+1} = W\mathbf{x}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k)$$

Limit:

$$\hat{\mathbf{x}} := \lim_{k\to\infty} \mathbf{x}^k,$$

$\lim_k$(DGD iteration):

$$(W - I)\,\hat{\mathbf{x}} + \alpha\nabla\mathbf{f}(\hat{\mathbf{x}}) = 0.$$

Since $\hat{\mathbf{x}}$ is consensual $\iff (W - I)\,\hat{\mathbf{x}} = 0 \iff \nabla\mathbf{f}(\hat{\mathbf{x}}) = 0$, we have

Speed-exactness dilemma

DGD iteration:

$$\mathbf{x}^{k+1} = W\mathbf{x}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k)$$

Limit:

$$\hat{\mathbf{x}} := \lim_{k\to\infty} \mathbf{x}^k,$$

$\lim_k$(DGD iteration):

$$(W - I)\,\hat{\mathbf{x}} + \alpha\nabla\mathbf{f}(\hat{\mathbf{x}}) = 0.$$

Since $\hat{\mathbf{x}}$ is consensual $\iff (W - I)\,\hat{\mathbf{x}} = 0 \iff \nabla\mathbf{f}(\hat{\mathbf{x}}) = 0$, we have

Proposition

DGD is exact with a fixed α only if a single x minimizes all f_i's.

However, the original problem only minimizes the sum.

Develop new algorithm: EXTRA

Assume:

- convergence $\mathbf{x}^k \to \bar{\mathbf{x}}$;
- same assumptions on W and, $W\mathbf{y} = \mathbf{y} \iff \mathbf{y} = \mathbf{1}$

Goal: obtain

- $\bar{\mathbf{x}}$ is consensual $\iff W\bar{\mathbf{x}} = \bar{\mathbf{x}}$;
- $\bar{\mathbf{x}}$ is optimal $\iff \mathbf{1}^{\mathrm{T}}\nabla\mathbf{f}(\bar{\mathbf{x}}) = 0$.

Develop new algorithm: EXTRA

Assume:

- convergence $\mathbf{x}^k \to \bar{\mathbf{x}}$;
- same assumptions on W and, $W\mathbf{y} = \mathbf{y} \iff \mathbf{y} = \mathbf{1}$

Goal: obtain

- $\bar{\mathbf{x}}$ is consensual $\iff W\bar{\mathbf{x}} = \bar{\mathbf{x}}$;
- $\bar{\mathbf{x}}$ is optimal $\iff \mathbf{1}^{\mathrm{T}}\nabla \mathbf{f}(\bar{\mathbf{x}}) = 0$.

Reason: original problem $\min_{x\in\mathbb{R}^p} \sum_{i=1}^n f_{(i)}(x)$ is equivalent to

$$\underset{\mathbf{x}\in\mathbb{R}^{n\times p}}{\text{minimize}}\ \mathbf{1}^{\mathrm{T}}\mathbf{f}(\mathbf{x}),\ \text{subject to } W\mathbf{x} = \mathbf{x}.$$

Introduce

$$\overline{W} := (W + I)/2.$$

Take the difference between two DGD iterations

$$\mathbf{x}^{k+1} = \overline{W}\mathbf{x}^{k} - \alpha\nabla\mathbf{f}(\mathbf{x}^{k}), \tag{1}$$

$$\mathbf{x}^{k+2} = W\mathbf{x}^{k+1} - \alpha\nabla\mathbf{f}(\mathbf{x}^{k+1}), \tag{2}$$

Introduce

$$\overline{W} := (W + I)/2.$$

Take the difference between two DGD iterations

$$\mathbf{x}^{k+1} = \overline{W}\mathbf{x}^{k} - \alpha\nabla\mathbf{f}(\mathbf{x}^{k}), \tag{1}$$

$$\mathbf{x}^{k+2} = W\mathbf{x}^{k+1} - \alpha\nabla\mathbf{f}(\mathbf{x}^{k+1}), \tag{2}$$

we get the new iteration: "EXTRA"

$$\boxed{\mathbf{x}^{k+2} - \mathbf{x}^{k+1} = W\mathbf{x}^{k+1} - \overline{W}\mathbf{x}^{k} - \alpha\nabla\mathbf{f}(\mathbf{x}^{k+1}) + \alpha\nabla\mathbf{f}(\mathbf{x}^{k}).} \tag{3}$$

Introduce

$$\overline{W} := (W + I)/2.$$

Take the difference between two DGD iterations

$$\mathbf{x}^{k+1} = \overline{W}\mathbf{x}^{k} - \alpha\nabla\mathbf{f}(\mathbf{x}^{k}), \tag{1}$$

$$\mathbf{x}^{k+2} = W\mathbf{x}^{k+1} - \alpha\nabla\mathbf{f}(\mathbf{x}^{k+1}), \tag{2}$$

we get the new iteration: "EXTRA"

$$\boxed{\mathbf{x}^{k+2} - \mathbf{x}^{k+1} = W\mathbf{x}^{k+1} - \overline{W}\mathbf{x}^{k} - \alpha\nabla\mathbf{f}(\mathbf{x}^{k+1}) + \alpha\nabla\mathbf{f}(\mathbf{x}^{k}).} \tag{3}$$

Letting $k \to \infty$ and canceling terms give us:

$$0 = (W - \overline{W})\bar{\mathbf{x}} = \frac{1}{2}\left(W\bar{\mathbf{x}} - \bar{\mathbf{x}}\right).$$

$\implies W\bar{\mathbf{x}} = \bar{\mathbf{x}} \implies \bar{\mathbf{x}}$ **is consensual**.

Adding 1st iteration (still DGD)

$$\mathbf{x}^1 = W\mathbf{x}^0 - \alpha \nabla \mathbf{f}(\mathbf{x}^0)$$

to iterations $2, \ldots, k$ in $\boxed{\text{box}}$ **gives**

$$\mathbf{x}^{k+2} = W\mathbf{x}^{k+1} - \alpha \nabla \mathbf{f}(\mathbf{x}^{k+1}) + \sum_{i=0}^{k} (W - \overline{W})\mathbf{x}^i.$$

Adding 1st iteration (still DGD)

$$\mathbf{x}^1 = W\mathbf{x}^0 - \alpha\nabla\mathbf{f}(\mathbf{x}^0)$$

to iterations $2, \ldots, k$ in $\boxed{\text{box}}$ **gives**

$$\mathbf{x}^{k+2} = W\mathbf{x}^{k+1} - \alpha\nabla\mathbf{f}(\mathbf{x}^{k+1}) + \sum_{i=0}^{k}(W - \overline{W})\mathbf{x}^i.$$

Letting $k \to \infty$ and using $W\bar{\mathbf{x}} = \bar{\mathbf{x}}$ yield

$$\alpha\nabla\mathbf{f}(\bar{\mathbf{x}}) = \sum_{i=1}^{\infty}(W - \overline{W})\mathbf{x}^i.$$

Adding 1st iteration (still DGD)

$$\mathbf{x}^1 = W\mathbf{x}^0 - \alpha\nabla\mathbf{f}(\mathbf{x}^0)$$

to iterations $2, \ldots, k$ in $\boxed{\text{box}}$ **gives**

$$\mathbf{x}^{k+2} = W\mathbf{x}^{k+1} - \alpha\nabla\mathbf{f}(\mathbf{x}^{k+1}) + \sum_{i=0}^{k}(W - \overline{W})\mathbf{x}^i.$$

Letting $k \to \infty$ and using $W\bar{\mathbf{x}} = \bar{\mathbf{x}}$ yield

$$\alpha\nabla\mathbf{f}(\bar{\mathbf{x}}) = \sum_{i=1}^{\infty}(W - \overline{W})\mathbf{x}^i.$$

Using *left-stochasticity* $\mathbf{1}^T(W - \overline{W}) = 0$, we have

$$\mathbf{1}^T\nabla\mathbf{f}(\bar{\mathbf{x}}) = 0,$$

$\Longrightarrow$ $\bar{\mathbf{x}}$ is also optimal.

Proposition

Assuming convergence and $\mathbf{x}^k \to \bar{\mathbf{x}}$, then $\bar{\mathbf{x}}$ is an optimal consensus solution.

Explanation

New iteration:

$$\mathbf{x}^{k+1} = W\mathbf{x}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k) + \underbrace{\sum_{i=0}^{k-1}(W - \overline{W})\mathbf{x}^i}_{\text{correction}}.$$

- Assuming $\mathbf{x}^k$ is asymptotically consensual, so $\mathbf{x}^{k+1} - W\mathbf{x}^k$ is vanishing.
- need $\mathbf{1}^T\nabla\mathbf{f}(\mathbf{x}^k) \to 0$ (optimality). So, $\nabla\mathbf{f}(\mathbf{x}^k)$ needs to be neutralized over $\mathrm{span}\{\mathbf{1}\}^\perp$.
- $\sum_{i=0}^{k-1}(W - \overline{W})\mathbf{x}^i$ is the simplest term we found for this purpose.

Convergence results

Theorem (sublinear $1/k$ convergence)

Assume (i) convex objectives with Lipschitz gradients, (ii) consensus solution x^* exists, (iii) symmetric doubly stochastic W and $\overline{W}$ obeying

$$\overline{W} \succ 0 \quad \textit{and} \quad \frac{I+W}{2} \succeq \overline{W} \succeq W.$$

If step size $\alpha < 2\lambda_{\min}(\overline{W})/\max L_i$, then EXTRA has $O(1/k)$ ergodic convergence.

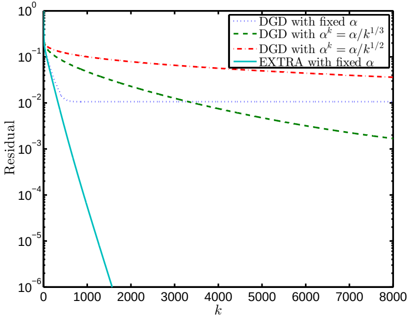
Theorem (linear convergence)

In addition, if

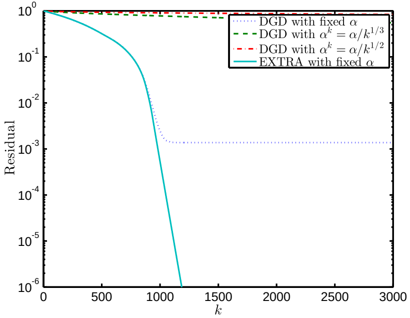
$$\sum_{i=1}^{n} f_i(x)$$

is (restrict) strongly convex, then $\|\mathbf{x}^k - \mathbf{x}^*\|_W$ converges to 0 with a global R-linear rate.

Example: decentralized least squares



Example: decentralized sum of Huber functions



Other numerical results

In our paper (Shi-Ling-Wu-Yin, arXiv:1404.6264)

- Results with **hand-optimized** parameters for all solvers
- Logistic regression example
- Some discussions on different mixing matrices W, such as general symmetric doubly stochastic (Tsitisklis'84), Laplacian-based $W = I - L/\tau$ (Xiao-Boyd'04, Sayed'12), Mestropolis (Xiao-Boyd-Lall'06), symmetric fastest distributed linear averaging (FDLA, Xiao-Boyd'-04).

Limitations and future work

Asymmetric mixing matrix W:

- $\mathbf{1}^T W \neq \mathbf{1}^T$: I may forget to send to neighbors, easier case
- $W\mathbf{1} \neq \mathbf{1}$: neighbors may not receive my messages, more difficult case (Macua, Leon, and co-authors can ensure $\mathbf{1}^T W = \mathbf{1}^T$ and $W\mathbf{1} = \mathbf{1}$)

Limitations and future work

Asymmetric mixing matrix W:

- $\mathbf{1}^T W \neq \mathbf{1}^T$: I may forget to send to neighbors, easier case
- $W\mathbf{1} \neq \mathbf{1}$: neighbors may not receive my messages, more difficult case (Macua, Leon, and co-authors can ensure $\mathbf{1}^T W = \mathbf{1}^T$ and $W\mathbf{1} = \mathbf{1}$)

Convergence improvement:

- optimal $O(1/k^2)$ convergence
- better constants by optimizing W

Limitations and future work

Asymmetric mixing matrix W:

- $\mathbf{1}^T W \neq \mathbf{1}^T$: I may forget to send to neighbors, easier case
- $W\mathbf{1} \neq \mathbf{1}$: neighbors may not receive my messages, more difficult case (Macua, Leon, and co-authors can ensure $\mathbf{1}^T W = \mathbf{1}^T$ and $W\mathbf{1} = \mathbf{1}$)

Convergence improvement:

- optimal $O(1/k^2)$ convergence
- better constants by optimizing W

Dynamic

- network topology varies over time
- $\mathbf{f}$ varies over time

Next: develop and analyze an ADMM approach

Build optimization algorithms that run on networks from basic operators:

- forward (gradient desc.) operator: $\mathbf{fwd}_f := (I - \nabla f)$
- **backward (proximal) operator:** $\mathbf{prox}_f$ (define later)
- **reflection operator:** $\mathbf{refl}_f := \mathbf{prox}_f + (\mathbf{prox}_f - I)$
- averaging operator: W where $W\mathbf{1} = \mathbf{1}$.

Next: develop and analyze an ADMM approach

Build optimization algorithms that run on networks from basic operators:

- forward (gradient desc.) operator: $\mathbf{fwd}_f := (I - \nabla f)$
- **backward (proximal) operator:** $\mathbf{prox}_f$ (define later)
- **reflection operator:** $\mathbf{refl}_f := \mathbf{prox}_f + (\mathbf{prox}_f - I)$
- averaging operator: W where $W\mathbf{1} = \mathbf{1}$.

Main references for distributed and decentralized ADMM:

- Bertsekas-Tsitsiklas'89 (distributed ADMM)
- Palomar-Chiang'06 (dual decomposition, network utility)
- Schizas-Ribeiro-Giannakis'08 (decentralized ADMM)

Proximal (backward) operator

- **Definition**: for a proper closed convex f (possibly nonsmooth), $\gamma > 0$,

$$\mathbf{prox}_{\gamma f}(y) := \arg\min_x \gamma f(x) + \frac{1}{2}\|x - y\|^2.$$

- **Equivalently,** $x = \mathbf{prox}_{\gamma f}(y)$ if and only if

$$\gamma \widetilde{\nabla} f(x) + (x - y) = 0, \quad \widetilde{\nabla} f(x) \in \partial f(x).$$

Proximal (backward) operator

- **Definition**: for a proper closed convex f (possibly nonsmooth), $\gamma > 0$,

$$\mathbf{prox}_{\gamma f}(y) := \arg\min_x \gamma f(x) + \frac{1}{2}\|x - y\|^2.$$

- **Equivalently,** $x = \mathbf{prox}_{\gamma f}(y)$ if and only if

$$\gamma \widetilde{\nabla} f(x) + (x - y) = 0, \quad \widetilde{\nabla} f(x) \in \partial f(x).$$

- **Generalization to projection**: Let C be a closed, nonempty set.
  Let $f := \chi_C$, which returns 0 if $x \in C$; ∞ if $x \notin C$.

$$\mathbf{prox}_{\chi_C} \equiv P_C$$

Proximal (backward) operator

- **Definition**: for a proper closed convex f (possibly nonsmooth), $\gamma > 0$,

$$\mathbf{prox}_{\gamma f}(y) := \arg\min_{x} \gamma f(x) + \frac{1}{2}\|x - y\|^2.$$

- **Equivalently,** $x = \mathbf{prox}_{\gamma f}(y)$ if and only if

$$\gamma \widetilde{\nabla} f(x) + (x - y) = 0, \quad \widetilde{\nabla} f(x) \in \partial f(x).$$

- **Generalization to projection**: Let C be a closed, nonempty set. Let $f := \chi_C$, which returns 0 if $x \in C$; ∞ if $x \notin C$.

$$\mathbf{prox}_{\chi_C} \equiv P_C$$

- **Reflection**:

$$\mathbf{refl}_{\gamma f} := \mathbf{prox}_{\gamma f} + (\mathbf{prox}_{\gamma f} - I) = 2\mathbf{prox}_{\gamma h} - I.$$

Forward vs backward

- **Forward**: explicit, easier to compute, γ must be small enough

$$z^{k+1} = z^k - \gamma \widetilde{\nabla} f(z^k).$$

- **Backward**: implicit, difficult to compute except for few, $\gamma > 0$ is ok

$$z^{k+1} = z^k - \gamma \widetilde{\nabla} f(z^{k+1}).$$

What is splitting?

- Use basic operators (forward, proximal, reflection) of f and g to solve

$$\underset{x \in \mathcal{H}}{\text{minimize}} \; f(x) + g(x)$$

  and

$$\underset{x \in \mathcal{H}_1, y \in \mathcal{H}_2}{\text{minimize}} \; f(x) + g(y) \quad \text{subject to } Ax + By = b,$$

Assumptions:

- $\mathcal{H}, \mathcal{H}_1, \mathcal{H}_2$ are Hilbert spaces, may be finite dimensional
- All functions are proper, closed, convex; may or may not be differentiable
- Saddle point must exist when duality is used

Examples

$$\underset{x}{\text{minimize}}\ \ f(x) + g(x)$$

- **point in the intersection:** $f = \chi_{C_1}$ and $g = \chi_{C_2}$.

  $$\text{Find } x \in C_1 \cap C_2 \iff \text{minimize}\, f(x) + g(x)$$

- **constrained optimization:** $f = \chi_C$, general g.

  $$\text{minimize}\, g(x),\ \text{subject to } x \in C \iff \text{minimize}\, f(x) + g(x)$$

- **regularized regression:** f is *data fitting*, g enforces *prior knowledge*

Examples

$$\underset{x}{\text{minimize}} \;\; f(x) + g(x)$$

- **point in the intersection:** $f = \chi_{C_1}$ and $g = \chi_{C_2}$.

$$\text{Find } x \in C_1 \cap C_2 \iff \text{minimize}\, f(x) + g(x)$$

- **constrained optimization:** $f = \chi_C$, general g.

$$\text{minimize}\, g(x),\;\text{subject to } x \in C \;\; \iff \text{minimize}\, f(x) + g(x)$$

- **regularized regression:** f is *data fitting*, g enforces *prior knowledge*
- **consensus optimization:**

$$\text{minimize} \sum_{i=1}^{m} h_i(x) \iff \text{minimize}\, f(\mathbf{x}) + g(\mathbf{x})$$

where $\mathbf{x} = (x_1, \ldots, x_m)$, $f(\mathbf{x}) = \sum_{i=1}^{m} h_i(x_i)$, $g(x) = \chi_{\{\mathbf{x} | x_1 = \cdots = x_m\}}(\mathbf{x})$

Forward-backward splitting (FBS)

- assumption: g is **differentiable**

$$z^{k+1} = \mathbf{prox}_{\gamma f} \circ \mathbf{fwd}_{\gamma g}(z^k) = \mathbf{prox}_{\gamma f}\left(z^k - \gamma \nabla g(z^k)\right)$$

- extends the gradient–projection iteration (when $f = \chi_C$)
- traces back to 1970s: Bruck[1], Lions and Mercier[2]
- converge if step size $\gamma \in (0, 2/L)$, where L is the Lip. constant of ∇g

---

[1] R. Bruck "An iterative solution of a variational inequality for certain monotone operator in a Hilbert space" 1975

[2] P. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," 1979.

Douglas-Rachford splitting (DRS)

- ~~g is differentiable~~
- **DRS algorithm:**

$$z^{k+1} = \left(\frac{1}{2}I + \frac{1}{2}\mathbf{refl}_{\gamma f}\mathbf{refl}_{\gamma g}\right) z^k$$

- $z^k \rightharpoonup$ to a fixed point, given existence; unbounded, otherwise[3]
- fixed points $\not\equiv$ minimizers of $f + g$.
- however, $\mathbf{prox}_{\gamma g}(z^k) \rightharpoonup$ a minimizer (first proof in 2011).[4]
- **early history:**
  - proposed by Douglas and Rachford (1956) to solve matrix equations.
  - analyzed for monotone operator by Lions and Mercier (1979) [5].

[3] J.Eckstein, D.Bertsekas "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators." Math. Prog. 1992.

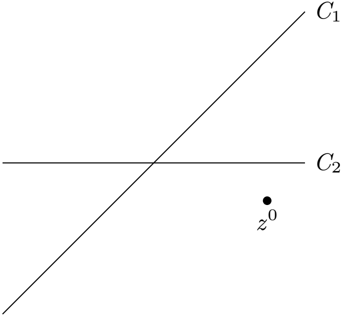
[4] Svaiter, On weak convergence of the Douglas-Rachford method

[5] Splitting algorithms for the sum of two nonlinear operators

DRS special case: “reflect, reflect, average”

C_1 and C_2 are closed convex sets. Find $x \in C_1 \cap C_2$, assumed to exist.

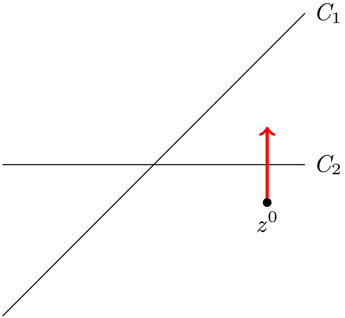
$$z^{k+1} = \frac{1}{2}z^k + \frac{1}{2}(2P_{C_1} - I)(2P_{C_2} - I)(z^k).$$



DRS special case: "reflect, reflect, average"

C_1 and C_2 are closed convex sets. Find $x \in C_1 \cap C_2$, assumed to exist.

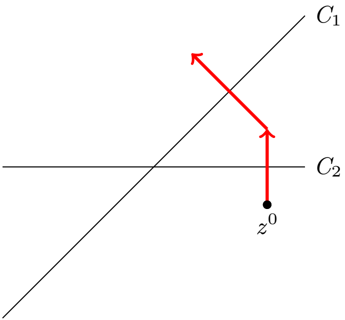
$$z^{k+1} = \frac{1}{2} z^k + \frac{1}{2}(2P_{C_1} - I)(2P_{C_2} - I)(z^k).$$



DRS special case: "reflect, reflect, average"

C_1 and C_2 are closed convex sets. Find $x \in C_1 \cap C_2$, assumed to exist.

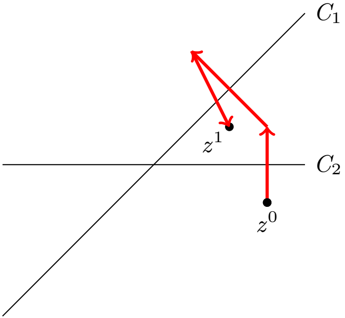
$$z^{k+1} = \frac{1}{2} z^k + \frac{1}{2}(2P_{C_1} - I)(2P_{C_2} - I)(z^k).$$



DRS special case: “reflect, reflect, average”

C_1 and C_2 are closed convex sets. Find $x \in C_1 \cap C_2$, assumed to exist.

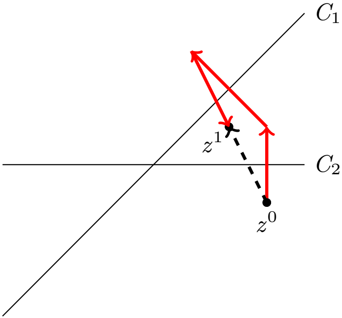
$$z^{k+1} = \frac{1}{2}z^k + \frac{1}{2}(2P_{C_1} - I)(2P_{C_2} - I)(z^k).$$



DRS special case: "reflect, reflect, average"

C_1 and C_2 are closed convex sets. Find $x \in C_1 \cap C_2$, assumed to exist.

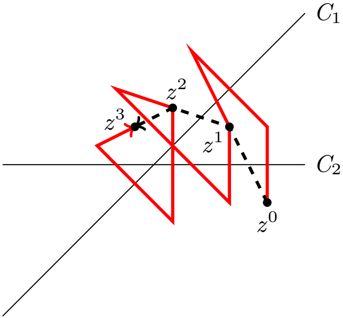
$$z^{k+1} = \frac{1}{2} z^k + \frac{1}{2}(2P_{C_1} - I)(2P_{C_2} - I)(z^k).$$



DRS special case: "reflect, reflect, average"

C_1 and C_2 are closed convex sets. Find $x \in C_1 \cap C_2$, assumed to exist.

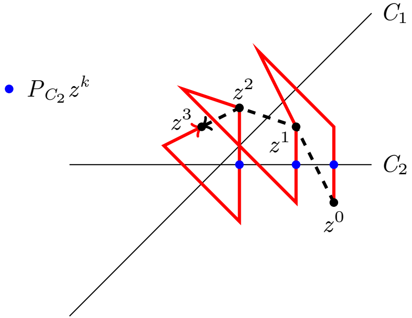
$$z^{k+1} = \frac{1}{2} z^k + \frac{1}{2}(2P_{C_1} - I)(2P_{C_2} - I)(z^k).$$



DRS special case: "reflect, reflect, average"

C_1 and C_2 are closed convex sets. Find $x \in C_1 \cap C_2$, assumed to exist.

$$z^{k+1} = \frac{1}{2} z^k + \frac{1}{2}(2P_{C_1} - I)(2P_{C_2} - I)(z^k).$$



Peaceman-Rachford splitting (PRS)

- DRS without averaging:

$$z^{k+1} = \mathbf{refl}_{\gamma f} \mathbf{refl}_{\gamma g}(z^k)$$

- may not converge (may orbit with a fixed distance to the solution set)
- when it does converge, often faster than DRS

First-order algorithms: subgradient form

- **(Sub)gradient descent:**

$$z^{k+1} = z^k - \gamma \widetilde{\nabla} f(z^k) - \gamma \widetilde{\nabla} g(z^k).$$

- **Proximal point algorithm (PPA):**

$$z^{k+1} = z^k - \gamma \widetilde{\nabla} f(z^{k+1}) - \gamma \widetilde{\nabla} g(z^{k+1}).$$

- **Forward backward splitting (FBS):**

$$z^{k+1} = z^k - \gamma \widetilde{\nabla} f(z^{k+1}) - \gamma \widetilde{\nabla} g(z^k).$$

- **Douglas Rachford splitting (DRS):**

$$z^{k+1} = z^k - \gamma \widetilde{\nabla} f(x_f^k) - \gamma \widetilde{\nabla} g(x_g^k).$$

- **Douglas Rachford splitting (PRS):**

$$z^{k+1} = z^k - 2\gamma \widetilde{\nabla} f(x_f^k) - 2\gamma \widetilde{\nabla} g(x_g^k).$$

Example: consensus optimization

$$\underset{x}{\text{minimize}} \sum_{i=1}^{m} h_i(x)$$

- **variable splitting:** introduce
  - $\mathbf{x} = (x_1, \ldots, x_m)$,
  - $f(\mathbf{x}) = \sum_{i=1}^{m} h_i(x_i)$,
  - $g(x) = \chi_{\{\mathbf{x} | x_1 = \cdots = x_m\}}(\mathbf{x})$
- **reduce to two splitting problem:**

$$\underset{\mathbf{x}}{\text{minimize}} \ f(\mathbf{x}) + g(\mathbf{x})$$

- **DRS iteration:** for $k = 0, 1, 2, \ldots$, iteration

$$\text{consensus average} \quad \bar{z}^k = \frac{1}{m} \sum_{i=1}^{m} z_i^k$$

$$\text{for all } i \text{ in parallel} \quad \begin{cases} x_i^k = \mathbf{prox}_{\gamma f_i}(2\bar{z}^k - z_i^k); \\ z_i^{k+1} = \frac{1}{2} z_i^k + \frac{1}{2}(2x_i^k - (2\bar{z}^k - z_i^k)) \end{cases}$$

Linearly constrained splitting problem

- **Formulation:**

$$\begin{aligned} \underset{x\in\mathcal{H}_1, y\in\mathcal{H}_2}{\text{minimize}} \quad & f(x) + g(y) \\ \text{subject to} \quad & Ax + By = b \end{aligned}$$

  where $A : \mathcal{H}_1 \to \mathcal{G}$ and $B : \mathcal{H}_2 \to \mathcal{G}$ are linear

- **Function: split awkward combinations of f and g**
- **Main problems can be turned into this form by operator/variable splitting**

ADMM = DRS applied to the dual

- **Lagrangian:**

$$\mathcal{L}(x, y; w) := f(x) + g(y) - w^T(Ax + By - b)$$

- **Lagrange dual:**

$$\max_w(\min_{x,y} \mathcal{L}(x, y; w)) \iff \underset{w}{\text{minimize}} \; f^*(A^T w) + g^*(B^T w) - b^T w$$

where * denotes the convex conjugate (i.e., Legendar transform)

ADMM = DRS applied to the dual

- **Lagrangian:**

$$\mathcal{L}(x, y; w) := f(x) + g(y) - w^T(Ax + By - b)$$

- **Lagrange dual:**

$$\max_w(\min_{x,y} \mathcal{L}(x, y; w)) \iff \underset{w}{\text{minimize}} \; f^*(A^T w) + g^*(B^T w) - b^T w$$

where * denotes the convex conjugate (i.e., Legendar transform)

- Introduce

$$d_f(w) := f^*(A^T w) \quad \text{and} \quad d_g(w) := g^*(B^T w) - b^T w$$

- Apply **DRS algorithm** to

$$\underset{w \in \mathcal{G}}{\text{minimize}} \; d_f(w) + d_g(w)$$

- Obtain the simplified **dual DRS iteration:**

$$
\begin{aligned}
y^{k+1} &= \arg\min_{y} \mathcal{L}(x^k, y; w^k) \\
w^{k+1} &= w^k - \gamma(Ax^k + By^k - b) \\
x^{k+1} &= \arg\min_{x} \mathcal{L}(x, y^{k+1}; w^{k+1})
\end{aligned}
$$

  (sequence z^k is *hidden*)

- It is exactly equivalent to ADMM (alternating direction method of multipliers)

Example: consensus optimization

- **Consensus problem** can be turned to

$$\begin{aligned} \underset{\mathbf{x},\mathbf{y}}{\text{minimize}} \quad & \sum_{i\in\mathcal{V}} f_i(x_{(i)}) \\ \text{subject to} \quad & x_{(i)} = y_{ij},\ x_{(j)} = y_{ij},\ \forall (i,j) \in \mathcal{E}, \end{aligned}$$

where $\mathcal{V}$ and $\mathcal{E}$ is the set of network nodes and edges, respectively.

Example: consensus optimization

- **Consensus problem** can be turned to

$$\begin{aligned} \underset{\mathbf{x},\mathbf{y}}{\text{minimize}} \quad & \sum_{i\in\mathcal{V}} f_i(x_{(i)}) \\ \text{subject to} \quad & x_{(i)} = y_{ij},\ x_{(j)} = y_{ij},\ \forall (i,j) \in \mathcal{E}, \end{aligned}$$

  where $\mathcal{V}$ and $\mathcal{E}$ is the set of network nodes and edges, respectively.

- Apply ADMM and obtain simplified iteration:

$$\begin{cases} x_i^{k+1} = \arg\min_{x_i} f_i(x_i) + \frac{\gamma|\mathcal{N}_i|}{2}\|x_i - x_i^k - \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} x_j^k + \frac{1}{\gamma|\mathcal{N}_i|}\alpha_i\|^2 + \frac{\gamma|\mathcal{N}_i|}{2}\|x_i\|^2 \\ \alpha_i^{k+1} = \alpha_i^k + \gamma\left(|\mathcal{N}_i|x_i^{k+1} - \sum_{j\in\mathcal{N}_i} x_j^{k+1}\right). \end{cases}$$

  ($\mathcal{N}_i$ is the set of neighbors of node i.)

Convergence results for general ADMM (joint with D. Davis)

Ergodic rate: let $\overline{x}^k$ and $\overline{y}^k$ be the *running mean variables*

$$|f(\overline{x}^k) + g(\overline{y}^k) - f(x^*) - g(y^*)| = O\left(\frac{1}{k}\right),$$

$$\|A\overline{x}^k + B\overline{y}^k - b\|^2 = O\left(\frac{1}{k^2}\right).$$

Nonergodic rate:

$$|f(x^k) + g(y^k) - f(x^*) - g(y^*)| = o\left(\frac{1}{\sqrt{k}}\right),$$

$$\|Ax^k + By^k - b\|^2 = o\left(\frac{1}{k}\right).$$

Convergence results for general ADMM (joint with D. Davis)

Ergodic rate: let $\overline{x}^k$ and $\overline{y}^k$ be the *running mean variables*

$$|f(\overline{x}^k) + g(\overline{y}^k) - f(x^*) - g(y^*)| = O\left(\frac{1}{k}\right),$$

$$\|A\overline{x}^k + B\overline{y}^k - b\|^2 = O\left(\frac{1}{k^2}\right).$$

Nonergodic rate:

$$|f(x^k) + g(y^k) - f(x^*) - g(y^*)| = o\left(\frac{1}{\sqrt{k}}\right),$$

$$\|Ax^k + By^k - b\|^2 = o\left(\frac{1}{k}\right).$$

Comments:

- Neither objective error or constraint violation is monotonic.
- Better ergodic rate does not mean we should use the mean. It means current iterates may not be as stable in some cases.
- Rates are given under convexity and saddle-point existence only. Lipschitz gradients and/or strong convexity will improve them.

Application to decentralized ADMM for consensus problem

Ergodic rates:

$$\left|\sum_{i=1}^{m} f_i(\overline{x}_i^k) - f(x^*)\right| = O\left(\frac{1}{k+1}\right) \quad \text{and} \quad \sum_{\substack{i\in V\\ j\in N_i}} \|\overline{x}_i^k - \overline{z}_{ij}^k\|^2 = O\left(\frac{1}{(k+1)^2}\right).$$

Nonergodic rates:

$$\left|\sum_{i=1}^{m} f_i(x_i^k) - f(x^*)\right| = o\left(\frac{1}{\sqrt{k+1}}\right) \quad \text{and} \quad \sum_{\substack{i\in \mathcal{V}\\ j\in \mathcal{N}_i}} \|x_i^k - y_{ij}^k\|^2 = o\left(\frac{1}{k+1}\right).$$

Linear rates for all if f_i are strongly convex (with W.Shi and Q.Ling).

How do we show it?

Roughly, first do operator theoretic analysis: treat each iteration as

$$z^{k+1} = Tz^k$$

- establish *firmly nonexpansiveness*

$$\|Tx - Ty\|^2 \leq \|x - y\|^2 - \|(I - T)x - (I - T)y\|^2$$

- establish the rate for *fixed-point residual*

$$\|Tz^k - z^k\|^2$$

How do we show it?

Roughly, first do operator theoretic analysis: treat each iteration as

$$z^{k+1} = Tz^k$$

- establish *firmly nonexpansiveness*

$$\|Tx - Ty\|^2 \leq \|x - y\|^2 - \|(I - T)x - (I - T)y\|^2$$

- establish the rate for *fixed-point residual*

$$\|Tz^k - z^k\|^2$$

Then, do optimization analysis

- establish relation between $\|Tz^k - z^k\|^2$ and $f(x^k) + g(y^k)$
- for ADMM, apply Fenchel-Young inequality to translate from primal to dual

Conclusions

- Four **basic operators** are building blocks of many first-order algorithms
- **Splitting and duality**. They increase the scope those basic operators by orders of magnitude.
- Still lots of room to develop simple yet powerful algorithms
- Convex optimization: it is possible to achieve convergence rates on a network "similar to" the centralized case.