Operator Theory for Distributed Optimisation

Nicola Bastianello

with: Prof.s Ruggero Carli, Luca Schenato

May 9, 2019

Contents

Distributed optimisation

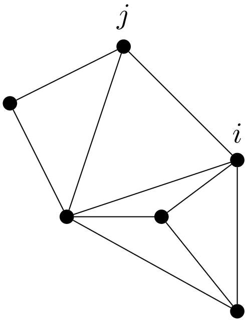
Ingredients:

- Distributed system
  - nodes can: 1) process information locally, 2) communicate with neighbours
  - *e.g.* co-operative robots, wireless sensor networks, smart grids, ...
- Problem
  - minimise the sum of *local costs* f_i

$$\min_{x_i} \sum_{i=1}^{N} f_i(x_i)$$

$$\text{s.t.} \quad x_i = x_j \text{ if } i, j \text{ neighbours}$$

where x_i is the *state* of node i



Distributed optimisation (cont'd)

Distributed consensus optimisation

1. perform local computations (*e.g.* minimise f_i)
2. exchange the results with neighbours (*e.g.* receive x_j)
3. incorporate information received (updating x_i)

- *Proposed algorithms*: gradient, Newton-Raphson, ADMM, ...

Issues:

- Faulty communications
  - packet loss, noise, ...
- Asynchronous operations
  - no clock synchronisation
- etc.
  - noise, switching communication topology, delays, ...

Contents

Operator theory

Operator (or **mapping**):

$$\mathcal{T} : \mathbb{H} \to \mathbb{H},$$

with $\mathbb{H}$ Hilbert space (here: $\mathbb{H} = \mathbb{R}^n$)

Types of operators:

- non-expansive $\|\mathcal{T}\boldsymbol{z} - \mathcal{T}\boldsymbol{w}\| \leq \|\boldsymbol{z} - \boldsymbol{w}\| \quad \forall \boldsymbol{z}, \boldsymbol{w} \in \mathbb{H}$
- contractive $\|\mathcal{T}\boldsymbol{z} - \mathcal{T}\boldsymbol{w}\| \leq \rho \|\boldsymbol{z} - \boldsymbol{w}\|$, $\rho \in (0, 1) \quad \forall \boldsymbol{z}, \boldsymbol{w} \in \mathbb{H}$
- averaged $\mathcal{T}\boldsymbol{z} = (1 - \alpha)\boldsymbol{z} + \alpha\mathcal{R}\boldsymbol{z}$, $\mathcal{R}$ non-expansive, $\alpha \in (0, 1)$

Operator theory (cont'd)

Objective: finding the *fixed point(s)* of $\mathcal{T}$ with an iterative algorithm

- *i.e.* $\operatorname{fix}(\mathcal{T}) := \{\bar{\boldsymbol{z}} \in \mathbb{H} \mid \bar{\boldsymbol{z}} = \mathcal{T}\bar{\boldsymbol{z}}\}$

Fixed point algorithms:

- *Banach-Picard*: for $\mathcal{T}$ contractive

$$\boldsymbol{z}(k+1) = \mathcal{T}\boldsymbol{z}(k) \qquad k \in \mathbb{N}$$

  $\Rightarrow$ linear convergence to the unique fixed point

- *Krasnosel'skiĭ-Mann* : for $\mathcal{T}$ non-expansive, $\alpha \in (0,1)$

$$\boldsymbol{z}(k+1) = (1-\alpha)\boldsymbol{z}(k) + \alpha\mathcal{T}\boldsymbol{z}(k) \qquad k \in \mathbb{N}$$

  $\Rightarrow$ $O(1/\sqrt{k})$ convergence to one fixed point

Affine operators

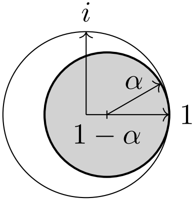
$\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ is *affine* if

$$\mathcal{T}\boldsymbol{z} = \boldsymbol{T}\boldsymbol{z} + \boldsymbol{u}$$

$\boldsymbol{T} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{u} \in \mathbb{R}^n$

Properties:

- $\text{fix}(\mathcal{T}) \neq \emptyset$ iff $\boldsymbol{u} \in \text{im}(\boldsymbol{I} - \boldsymbol{T})$
- distribution eigenvalues of $\boldsymbol{T}$



- convergence rate: largest eigenvalues strictly inside unit circle

Convex optimisation

Operator theory for **convex optimisation**:

$$\min_{\boldsymbol{z}} f(\boldsymbol{z})$$

$f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ a closed, proper and convex function

- Idea: design an operator with: $\operatorname{fix}(\mathcal{T})$ = minima of f

For example **proximal operator** ($\rho > 0$):

$$\operatorname{prox}_{\rho f}(\boldsymbol{z}) = \arg\min_{\boldsymbol{w}} \left\{ f(\boldsymbol{w}) + \frac{1}{2\rho} \|\boldsymbol{w} - \boldsymbol{z}\|^2 \right\}$$

or **reflective operator**: $\operatorname{refl}_{\rho f}(\boldsymbol{z}) = 2 \operatorname{prox}_{\rho f}(\boldsymbol{z}) - \boldsymbol{z}$

Splitting operators

Given the problem

$$\min_{\boldsymbol{z}} \{f(\boldsymbol{z}) + g(\boldsymbol{z})\}$$

$f, g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ closed, proper and convex functions

- directly computing prox_{f+g} may be too difficult!

but we can exploit its structure with **splitting operators**:

- smaller computations
- applied to dual: ADMM, gradient, Newton-Raphson
- distributed optimisation: *prove convergence with non-idealities*

For example **Peaceman-Rachford**:

$$\mathcal{T}_{\mathrm{PR}} = \mathrm{refl}_{\rho g} \circ \mathrm{refl}_{\rho f}$$

Contents

ADMM

Given the problem

$$\min_{\boldsymbol{x},\boldsymbol{y}} \{f(\boldsymbol{x}) + g(\boldsymbol{y})\}$$
$$\text{s.t. } \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} = \boldsymbol{c}$$

$f, g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ closed, proper and convex functions

we define the *augmented Lagrangian*

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{w}) = f(\boldsymbol{x}) + g(\boldsymbol{y}) - \langle \boldsymbol{w}, \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} - \boldsymbol{c}\rangle + \frac{\rho}{2} \|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} - \boldsymbol{c}\|^2$$

where *Lagrange multipliers* $\boldsymbol{w}$ and *penalty* $\rho > 0$

ADMM (cont'd)

Given the problem

$$\min_{\boldsymbol{x},\boldsymbol{y}} \{f(\boldsymbol{x}) + g(\boldsymbol{y})\}$$
$$\text{s.t. } \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} = \boldsymbol{c}$$

$f, g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ closed, proper and convex functions

the *(relaxed) ADMM* is ($\alpha \in (0, 1)$):

$$\boldsymbol{y}(k+1) = \arg\min_{\boldsymbol{y}} \Big\{ \mathcal{L}(\boldsymbol{x}(k), \boldsymbol{y}; \boldsymbol{w}(k)) + $$
$$+ \rho(2\alpha - 1)\langle \boldsymbol{B}\boldsymbol{y}, \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{y}(k) - \boldsymbol{c}\rangle \Big\}$$
$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \Big(\boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{y}(k+1) - \boldsymbol{c}\Big) + $$
$$- \rho(2\alpha - 1)\Big(\boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{y}(k) - \boldsymbol{c}\Big)$$
$$\boldsymbol{x}(k+1) = \arg\min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}(k); \boldsymbol{w}(k))$$

ADMM as splitting operator

Alternatively, given the (Fenchel) dual problem

$$\min_{\boldsymbol{w}} \{d_f(\boldsymbol{w}) + d_g(\boldsymbol{w})\}$$

applying the Krasnosel'skiĭ-Mann to the *Peaceman-Rachford operator*

$$\mathcal{T}_{\mathrm{PR}} = \mathrm{refl}_{\rho g} \circ \mathrm{refl}_{\rho f}$$

i.e.

$$\boldsymbol{z}(k+1) = (1-\alpha)\boldsymbol{z}(k) + \alpha\, \mathrm{refl}_{\rho g}\left(\mathrm{refl}_{\rho f}(\boldsymbol{z}(k))\right)$$

- is equivalent to the *(relaxed) ADMM*

ADMM for distributed optimisation

We can rewrite the *consensus constraints*

$$x_i = x_j \text{ if } i, j \text{ neighbours}$$

as

$$x_i = y_{ij}, \quad x_j = y_{ji}, \quad y_{ij} = y_{ji} \quad \text{if } i, j \text{ neighbours}$$

or equivalently

$$\boldsymbol{A}\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{0}, \quad \boldsymbol{y} = \boldsymbol{P}\boldsymbol{y}$$

where $\boldsymbol{P}$ permutation matrix that swaps y_{ij} with y_{ji}

ADMM for distributed optimisation (cont'd)

Thus

$$\min_{x_i} \sum_{i=1}^{N} f_i(x_i)$$
$$\text{s.t.} \quad x_i = x_j \text{ if } i, j \text{ neighbours}$$

becomes

$$\min_{\boldsymbol{x},\boldsymbol{y}} \left\{ \sum_{i=1}^{N} f_i(x_i) + \iota_{(\boldsymbol{I}-\boldsymbol{P})}(\boldsymbol{y}) \right\}$$
$$\text{s.t. } \boldsymbol{A}\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{0}$$
$$\iota_{(\boldsymbol{I}-\boldsymbol{P})}(\boldsymbol{y}) = \begin{cases} 0 & \text{if } \boldsymbol{y} = \boldsymbol{P}\boldsymbol{y} \\ +\infty & \text{otherwise} \end{cases}$$

- and we can apply the ADMM

ADMM for distributed optimisation (cont'd)

The distributed ADMM is given by

$$x_i(k) = \arg\min_{x_i} \left\{ f_i(x_i) - \langle x_i, \sum_{j \in \mathcal{N}} z_{ji}(k) \rangle + \frac{\rho}{2} |\mathcal{N}_i| \, \|x_i\|^2 \right\}$$
$$z_{ij}(k+1) = (1-\alpha) z_{ij}(k) - \alpha z_{ji}(k) + 2\alpha\rho x_i(k)$$

where

$$\mathcal{N}_i = \{ j \text{ s.t. } j \text{ neighbour of } i \}$$

Convergence

Assumptions

There is an equal *packet loss probability* p on each link.
Each node performs an *update with probability* u.
Packet loss and update events are *independent*.

Proposition (convergence)

The states of the nodes converge *almost surely* to the optimal solution of the distributed optimisation problem, *i.e.*

$$\lim_{k \to \infty} x_i(k) = x^* \qquad \forall i.$$

- **Proof**: exploit convergence of Peaceman-Rachford when only a *subset of co-ordinates is updated*.

Linear convergence

Assumption

The local costs f_i are *strongly convex*.

Thorem (Linear convergence)

There exists a neighbourhood of x^* s.t. any initial condition inside it converges linearly, in mean square, *i.e.*

$$\mathbb{E}\left[\|x_i(k) - x^*\|^2\right] \leq C\lambda_M^k$$

$C \in \mathbb{R}$ and $\lambda_M = \max_{\lambda \in \Lambda(\Sigma)} |\lambda| < 1$

- Σ depends on packet loss & update probabilities, graph, local costs

Sketch of proof

- the robust & asynchronous ADMM can be rewritten as

$$\begin{aligned}\boldsymbol{x}(k) &= \boldsymbol{Q}\boldsymbol{A}^\top \boldsymbol{z}(k) + \boldsymbol{r} \\ \boldsymbol{z}(k+1) &= \boldsymbol{T}\boldsymbol{z}(k) + \boldsymbol{u} + o(\boldsymbol{x}(k) - \boldsymbol{x}^*)\end{aligned}$$

  where $\boldsymbol{Q}$, $\boldsymbol{r}$ depend on graph and local costs
- since $\boldsymbol{x}^*$ is unique we can prove $\ker(\boldsymbol{I} - \boldsymbol{T}) \subset \ker(\boldsymbol{A}^\top)$
- moreover, the dual update is affine, hence we can write

$$\boldsymbol{T} = \boldsymbol{T}_1 + \boldsymbol{T}_{<1}$$

  where $\boldsymbol{T}_1$ depends on eigenvectors relative to eigenvalues in 1

Sketch of proof (cont'd)

- then the primal error is

$$\boldsymbol{x}(k) - \boldsymbol{x}^* = \boldsymbol{Q}\boldsymbol{A}^\top \boldsymbol{T}_{<1}(\boldsymbol{z}(k) - \bar{\boldsymbol{z}}) + o(\|\boldsymbol{x}(k) - \boldsymbol{x}^*\|)$$

$\bar{\boldsymbol{z}} \in \mathrm{fix}(\boldsymbol{T} \cdot + \boldsymbol{u})$

- accounting for the random updates and taking the norm

$$\mathbb{E}\left[\|\boldsymbol{x}(k) - \boldsymbol{x}^*\|^2\right] \leq C\,\|\Sigma\|^k + o(\|\boldsymbol{x}(k) - \boldsymbol{x}^*\|)$$

with Σ having all eigenvalues inside the unitary circle

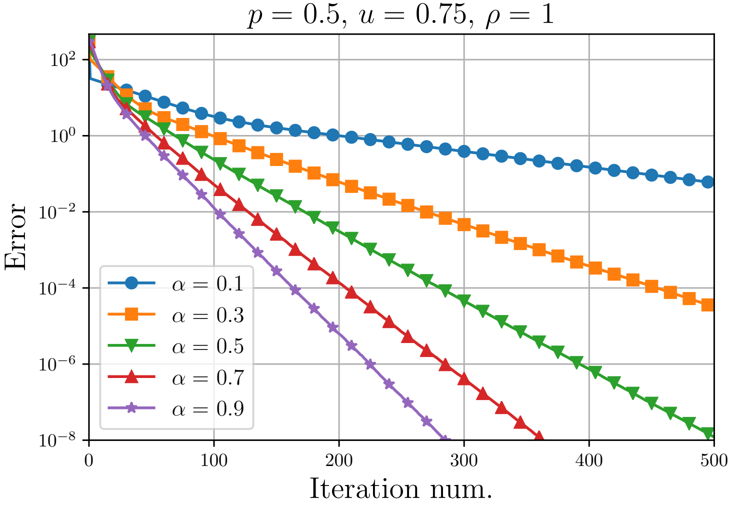
Contents

Setup

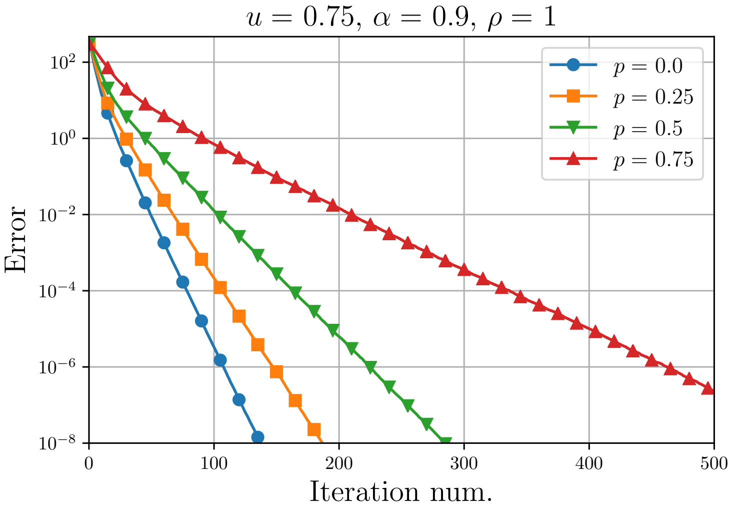
Simulations on

- random graph $N = 25$ nodes
- packet loss probability p and update probability u
- 100 Monte Carlo iterations

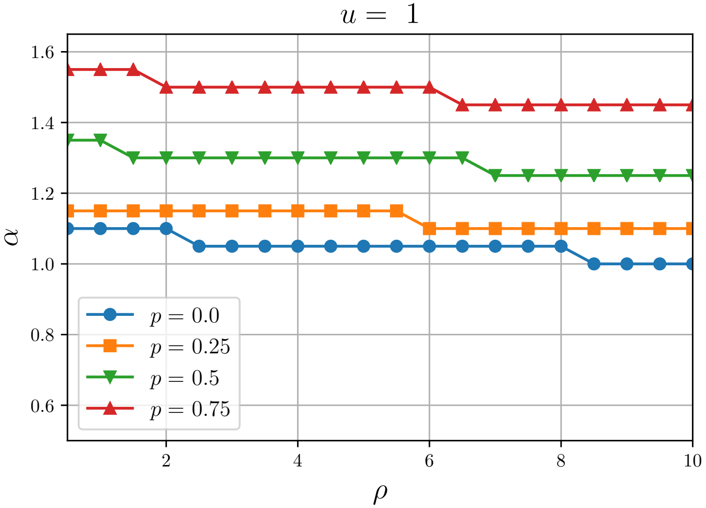
Varying α



Varying packet loss



Stability regions



Contents

Conclusion

Recap

- distributed optimisation with operator theory
- robust and asynchronous ADMM
- linear convergence

Future work

- ADMM with other non-idealities (*e.g.* noise)
- extend results to gradient method
- general convergence of Krasnosel'skiĭ-Mann with non-idealities

Thank you for your attention!