Where does ADMM come from: Lagrangian point of view

Andersen Ang
Dept. Combinatorics & Optimization, University of Waterloo, Canada
ms$\boldsymbol{x}$ang@uwaterloo.ca, where $\boldsymbol{x} = \lfloor \pi \rfloor$ Homepage: angms.science

First draft: November 1, 2022 Last update: November 9, 2022

The starting problem: a linearly constrained minimization

$$(\mathcal{P}) \ : \ \min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}.$$

- $f : \mathbb{R}^n \to \mathbb{R}$ is
  - convex (will specify strictly convexity later)
  - closed (epigraph of f is a closed set, this is a common assumption)
  - possibly nondifferentiable (∇f is not continuous)
- $\boldsymbol{x} \in \mathbb{R}^n$ is the optimization variable.
- $\boldsymbol{A} \in \mathbb{R}^{m \times n}$
  - in general need not to be full rank
  - for studying theoretical property $\boldsymbol{A}$ has to be full rank
- $\boldsymbol{b} \in \mathbb{R}^m$.

Lagrangian

$$(\mathcal{P}) \ : \ \min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

- Joseph-Louis Lagrange: add[1] the constraint into the objective function!

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \rangle.$$

- Lagrangian $\mathcal{L}$ is a function of two variables: $\boldsymbol{x}$ and $\boldsymbol{\lambda} \in \mathbb{R}^m$: Lagrangian multiplier
- Now we solve

$$(\mathcal{S}) \ : \ \min_{\boldsymbol{x}} \max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}). \qquad \text{(Lagrangian saddle point problem)}$$

Assuming strong duality[2], solving $\mathcal{S}$ gives the same result as solving the original problem $\Longrightarrow$ that's why Lagrangian approach is useful.

[1] Actually you can subtract: $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) - \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \rangle$, the sign of $\boldsymbol{\lambda}$ doesn't matter.

[2] Slater's condition: $\mathcal{P}$ is a convex problem and $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ has a solution, which translate to $\boldsymbol{A}$ is full rank.

Why Lagrangian?

$$(\mathcal{P}) \ : \ \min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

- $\mathcal{P}$ is **constrained**.
  - Some constraint is hard to solve.
  - To find $\boldsymbol{x}^*$ by an iterative algorithm we may need to keep all iterates $\boldsymbol{x}_k$ to be **feasible**.
  - You need to worry about both the feasibility and the optimality of $\boldsymbol{x}$.

- $\mathcal{S}$ has no constraint.

$$(\mathcal{S}) \ : \ \min_{\boldsymbol{x}} \max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}). \qquad \text{(Lagrangian saddle point problem)}$$

  - All $\boldsymbol{x}, \boldsymbol{\lambda}$ are feasible, no need to worry about them being infeasible, only need to worry about them being "not good enough" (not close to optimality).
  - No feasibility restriction seems nice, but you are paying a trade off by introducing an extra variable $\boldsymbol{\lambda}$.

Simplifying the Lagrangian: dual

$$(\mathcal{S}) \ : \ \min_{\boldsymbol{x}} \max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}). \qquad \text{(Lagrangian saddle point problem)}$$

- We have two variables $\boldsymbol{x}, \boldsymbol{\lambda}$.
- It is natural to think of eliminating one variable.
  - The idea: we just introduced $\boldsymbol{\lambda}$ in $\mathcal{L}$, so eliminating $\boldsymbol{\lambda}$ means we go back to the original problem (making a U-turn), so we better eliminate $\boldsymbol{x}$ instead.
  - Generally speaking: in some cases eliminating $\boldsymbol{\lambda}$ is better.
- How to eliminate $\boldsymbol{x}$ form $\mathcal{L}$: treat $\boldsymbol{\lambda}$ as a constant and solve $\min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x})$.
- The remaining of $\mathcal{L}$ after eliminating $\boldsymbol{x}$ is called the **dual**.

Computing the dual

$$(\mathcal{P}) \;:\; \min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

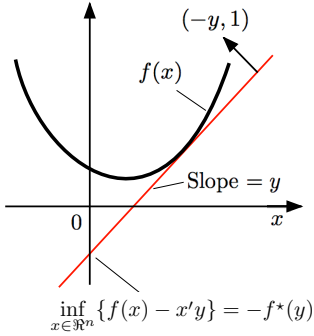
$$\min_{\boldsymbol{x}} \max_{\boldsymbol{\lambda}} \Big\{ \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \rangle \Big\}$$

$$\begin{aligned}
d(\boldsymbol{\lambda}) \coloneqq \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}) &= \min_{\boldsymbol{x}} f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \rangle \\
&= \min_{\boldsymbol{x}} f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} \rangle + \underbrace{\langle \boldsymbol{\lambda}, -\boldsymbol{b} \rangle}_{\text{constant}} \\
&= \min_{\boldsymbol{x}} f(\boldsymbol{x}) + \langle \boldsymbol{A}^\top \boldsymbol{\lambda}, \boldsymbol{x} \rangle - \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle && \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} \rangle = \langle \boldsymbol{A}^\top \boldsymbol{\lambda}, \boldsymbol{x} \rangle \\
&= -\Big(\max_{\boldsymbol{x}} -f(\boldsymbol{x}) - \langle \boldsymbol{A}^\top \boldsymbol{\lambda}, \boldsymbol{x} \rangle \Big) - \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle && \min \phi(\boldsymbol{x}) = -\max -\phi(\boldsymbol{x}) \\
&= -\Big(\max_{\boldsymbol{x}} \langle -\boldsymbol{A}^\top \boldsymbol{\lambda}, \boldsymbol{x} \rangle - f(\boldsymbol{x}) \Big) - \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle \\
&= -f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) - \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle && f^*(\boldsymbol{\xi}) \coloneqq \max_{\boldsymbol{\zeta}} \langle \boldsymbol{\xi}, \boldsymbol{\zeta} \rangle - f(\boldsymbol{\zeta})
\end{aligned}$$

- The dual function, only on $\boldsymbol{\lambda}$ is $d(\boldsymbol{\lambda}) = -f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) - \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle$.
- If you are unfamiliar with convex analysis, the WTF-moment here is the conjugate $f^*(\boldsymbol{\xi}) \coloneqq \max \Big(\langle \boldsymbol{\xi}, \boldsymbol{\zeta} \rangle - f(\boldsymbol{\zeta}) \Big)$.

Revision: what is conjugate (Bertsekas's explanations)

- Define a closed convex function by its epigraph.
- Describe the epigraph by supporting hyperplanes.
- Conjugate function = crossing point of the hyperplanes



Primal Description	Dual Description
Values $f(x)$	Crossing points $f^*(y)$

Solving for $\boldsymbol{\lambda}^*$ on the dual

$$(\mathcal{P}) \;:\; \min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

$$\min_{\boldsymbol{x}} \max_{\boldsymbol{\lambda}} \Big\{ \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \rangle \Big\} \xrightarrow[\boldsymbol{x} \text{ is gone}]{\min_{\boldsymbol{x}}} \max_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda}).$$

- The dual function $d(\boldsymbol{\lambda}) = -f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) - \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle$ is only on $\boldsymbol{\lambda}$.
- If we solved for the optimal $\boldsymbol{\lambda}$

  $$\boldsymbol{\lambda}^* \in \operatorname*{argmax}_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda}),$$

  then we can solve for our original $\boldsymbol{x}$ using $\boldsymbol{\lambda}^*$ as

  $$\boldsymbol{x}^* \in \operatorname*{argmin}_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}^*).$$

- Why this works: strong duality[3].
- Use $\in$ instead of $=$ since the solution to $\operatorname*{argmax}_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda})$, $\operatorname*{argmin}_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}^*)$ is possibly nonunique!

[3] Revision: weak duality is $d^* < p^*$, and strong duality is $d^* = p^*$.
Strong duality for $\mathcal{P}$ holds if Slater's condition is true: f is convex and $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ has a solution.

Algorithm based on Lagrangian for solving $\mathcal{P}$

$$(\mathcal{P}) \ : \ \min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Algorithm 1: Algorithm based on Lagrangian

1	$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \rangle$	write down the Lagrangian
2	$d(\boldsymbol{\lambda}) = -f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) - \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle$	derive the dual using conjugate
3	$\boldsymbol{\lambda}^* \in \operatorname*{argmax}_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda})$	solve the dual problem
4	$\boldsymbol{x}^* \in \operatorname*{argmin}_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}^*)$	solve the primal solution

Drawbacks: it assumes steps 3,4 are **have closed-form solution**.

- **Exactly solving** $\operatorname{argmax} d(\boldsymbol{\lambda})$ in general can be hard.
- **Exactly solving** $\operatorname{argmin} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}^*)$ in general can be hard.
- If we cannot solve step 3, then we cannot even proceed to step 4.

How to improve **Algorithm based on Lagrangian**

Instead of exactly solving $\operatorname{argmax}_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda})$, we approximately solve it

Algorithm 2: An iterative algorithm

1 Write down $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \rangle$ and $d(\boldsymbol{\lambda}) = -f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) - \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle$

2 Initialize $\boldsymbol{x}_0, \boldsymbol{\lambda}_0$

3 **for** $k = 1, 2, \ldots$ **do**

4 Get $\boldsymbol{\lambda}_{k+1}$ from $\boldsymbol{x}_k, \boldsymbol{\lambda}_k$ that approximately solve $\operatorname{argmax}_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda})$ better

5 Solve $\boldsymbol{x}_{k+1} \in \operatorname{argmin}_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}_{k+1})$ using $\boldsymbol{\lambda}_{k+1}$

6 **end**

How to approximately solve $\operatorname{argmax}_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda})$: gradient ascent

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha_k \nabla_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda}) \Big|_{\boldsymbol{\lambda} = \boldsymbol{\lambda}_k}.$$

Is $d(\boldsymbol{\lambda})$ even differentiable? Is $\nabla_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda})$ unique? How to compute $\nabla_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda})$?

Differentiation of the dual

$$d(\boldsymbol{\lambda}) = -f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) - \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle$$

Some facts[4]:

1. Fact: $d(\boldsymbol{\lambda})$ is concave[5] so gradient accent solves $\max d(\boldsymbol{\lambda})$ eventually if we can compute ∇d
2. Fact: if $\boldsymbol{x}_k \in \operatorname{argmin} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}_k)$ is unique (only one minimizer), then
   - $d(\boldsymbol{\lambda})$ is differentiable at $\boldsymbol{\lambda}_k$
   - $\nabla_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda})\Big|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}_k} = \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}$
3. Fact: if $\boldsymbol{x}_k \in \operatorname{argmin} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}_k)$ is not unique, then
   - $d(\boldsymbol{\lambda})$ is not classically differentiable at $\boldsymbol{\lambda}_k$
   - we have to use subgradient for $d(\boldsymbol{\lambda})$ at $\boldsymbol{\lambda}_k$
   - $\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}$ is only a subgradient of $d(\boldsymbol{\lambda})$ at $\boldsymbol{\lambda}_k$
4. Fact: if f is strictly convex then (2) is true and we have

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha_k(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}).$$

[4] For proof see any convex analysis text.

[5] conjugate function is always convex, negative of convex is concave

Dual ascent algorithm for solving Lagrangian saddle point problem

- Problem: $\min_{\boldsymbol{x}} f(\boldsymbol{x})$ s.t. $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, f strictly convex and $\boldsymbol{A}$ full rank.
- Lagrangian saddle point problem: $\min_{\boldsymbol{x}} \max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$

Algorithm 3: Dual ascent Lagrangian algorithm

1 Write down $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \rangle$ and $d(\boldsymbol{\lambda}) = -f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) - \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle$
2 Initialize $\boldsymbol{x}_0, \boldsymbol{\lambda}_0$
3 **for** $k = 1, 2, \ldots$ **do**
4 $\quad \boldsymbol{x}_{k+1} = \operatorname{argmin}_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}_k)$
5 $\quad \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha_k(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{b})$
6 **end**

- Note: here the x-step and the y-step have swapped order.
- Need to determine dual ascent stepsize α_k
- This algorithm is already looking similar to ADMM!

Drawback of the dual ascent algorithm

Algorithm 4: Dual ascent Lagrangian algorithm

$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \rangle$
$d(\boldsymbol{\lambda}) = -f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) - \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle$
Initialize $\boldsymbol{x}_0, \boldsymbol{\lambda}_0$
for $k = 1, 2, \ldots$ **do**
    $\boldsymbol{x}_{k+1} = \operatorname{argmin}_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}_k)$
    $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha_k(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{b})$
end

- We have equality $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha_k(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{b})$ based on the assumption that $d(\boldsymbol{\lambda})$ is differentiable $\Leftarrow \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}_k)$ has unique minimizer $\Leftarrow f$ is strictly convex
- f is strictly convex is a strong assumption: that means only problems with strictly convex f can use dual ascent
- To make the algorithm works for more problems, we relax such condition.

Revision: strong and strict convexity

- Fact: strong convexity $\implies$ strict convexity.
- Fact: a function f is ρ-strongly convex for $\rho > 0$ if $f(\boldsymbol{x}) - \frac{\rho}{2}\|\boldsymbol{x}\|_2^2$ is convex.
- To relax the strict convexity condition in dual ascent, consider the following augmented Lagrangian: let $\rho \geq 0$

$$\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \rangle + \frac{\rho}{2}\|\boldsymbol{x}\|_2^2.$$

  Now $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}_k)$ is strongly convex ($\implies$ strictly convex) so $\min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}_k)$ has unique solution.
- But adding $\frac{\rho}{2}\|\boldsymbol{x}\|_2^2$ will force $\boldsymbol{x}$ to be unnecessarily small (bad for some applications). A better augmentation is $\frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$, i.e.

$$\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \rangle + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2.$$

  Now it will not force $\boldsymbol{x}$ to be unnecessarily small but encourage $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, exactly what we want!
- If $\boldsymbol{A}$ is full rank, then $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}_k)$ is $\rho\gamma$-strongly convex, where γ is the smallest eigenvalue value of $\boldsymbol{A}^\top \boldsymbol{A}$.

Augmented Lagrangian is $\rho\gamma$-strongly convex

$$\mathcal{L}_\rho(\boldsymbol{x}) = f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} \rangle + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$$

- Let γ be the smallest eigenvalue of $\boldsymbol{A}^\top \boldsymbol{A}$.

$$\begin{aligned} \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 &= \frac{\rho}{2}\langle \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x}, \boldsymbol{x} \rangle + \text{other terms} \\ &\geq \frac{\rho}{2}\langle \gamma \boldsymbol{I}\boldsymbol{x}, \boldsymbol{x} \rangle + \text{other terms} \\ &= \frac{\rho\gamma}{2}\|\boldsymbol{x}\|_2^2 + \text{other terms} \end{aligned}$$

- Hence $\mathcal{L}_p(\boldsymbol{x})$ is $\rho\gamma$-strongly convex.
- This also means that $\boldsymbol{A}$ need to be full rank, otherwise $\gamma = 0$.
- For comparison, dual $d(\boldsymbol{\lambda}) = -f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) - \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle$ is concave but not strongly concave.

Augmented Lagrangian Algorithm

- Problem: $\min_{\boldsymbol{x}} f(\boldsymbol{x})$ s.t. $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, f convex[6] and $\boldsymbol{A}$ full rank.
- Lagrangian saddle point problem: $\min_{\boldsymbol{x}} \max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \rangle$.
- Augmented Lagrangian $\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \rangle + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$ is $\rho\gamma$-strongly convex.

Algorithm 5: Augmented Lagrangian algorithm

Initialize $\boldsymbol{x}_0, \boldsymbol{\lambda}_0$
for $k = 1, 2, \ldots$ **do**
  $\boldsymbol{x}_{k+1} = \operatorname{argmin}_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}_k)$
  $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \rho(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{b})$
end

Remarks

- We use ρ as the gradient ascent stepsize.
- We do not see the dual $d(\boldsymbol{\lambda}) = -f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) - \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle$ because of dual ascent.
- Fact: after adding $\frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$, the dual gradient is still $\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{b}$.

[6] Only convexity is assumed, f needs no strictly convex.

Look at the gradient of dual again

- For $\mathcal{L}$ with strictly convex f,
  we have $\nabla d(\boldsymbol{\lambda}) = \boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{b}$, where $d(\boldsymbol{\lambda}) = \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$
- For $\mathcal{L}_\rho$ with convex f and $\boldsymbol{A}$ full rank,
  we have $\nabla d_\rho(\boldsymbol{\lambda}) = \boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{b}$, where $d_\rho(\boldsymbol{\lambda}) = \min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda})$
- $\nabla d(\boldsymbol{\lambda}) = \nabla d_\rho(\boldsymbol{\lambda}) = \boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{b}$ and being $\|\boldsymbol{A}\|_2$-Lipschitz
- For $\nabla d_\rho(\boldsymbol{\lambda})$

$$\|\nabla d_\rho(\boldsymbol{\lambda}) - \nabla d_\rho(\boldsymbol{\lambda}')\|_2 \leq \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}'\|_2 \leq \frac{1}{\rho}\|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|_2.$$

Prove $\|\nabla d_\rho(\boldsymbol{\lambda}) - \nabla d_\rho(\boldsymbol{\lambda}')\|_2 \le \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}'\|_2 \le \dfrac{1}{\rho}\|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|_2$

- Let $\boldsymbol{x} = \underset{\boldsymbol{x}}{\operatorname{argmin}}\, \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda})$ and $\boldsymbol{x}' = \underset{\boldsymbol{x}}{\operatorname{argmin}}\, \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}')$.
- By subgradient 1st-order optimality,

$$\begin{aligned} & \boldsymbol{0} \in \partial f(\boldsymbol{x}) + \boldsymbol{A}^\top \boldsymbol{\lambda} + \rho \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}), \quad && \boldsymbol{0} \in \partial f(\boldsymbol{x}') + \boldsymbol{A}^\top \boldsymbol{\lambda}' + \rho \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{x}' - \boldsymbol{b}) \\ \iff & -\boldsymbol{A}^\top \boldsymbol{\lambda} - \rho \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \in \partial f(\boldsymbol{x}), && \boldsymbol{A}^\top \boldsymbol{\lambda}' - \rho \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{x}' - \boldsymbol{b}) \in \partial f(\boldsymbol{x}') \end{aligned}$$

- By the fact that subgradient ∂f is a monotone operator: $\left\langle \partial f(\boldsymbol{x}) - \partial f(\boldsymbol{x}')\,,\, \boldsymbol{x} - \boldsymbol{x}' \right\rangle \ge 0$, we have

$$\left\langle -\boldsymbol{A}^\top \boldsymbol{\lambda} - \rho \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) + \boldsymbol{A}^\top \boldsymbol{\lambda}' + \rho \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{x}' - \boldsymbol{b})\,,\, \boldsymbol{x} - \boldsymbol{x}' \right\rangle \ge 0.$$

$$\left\langle -(\boldsymbol{\lambda} - \boldsymbol{\lambda}') - \rho(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}')\,,\, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}' \right\rangle \ge 0.$$

$$\left\langle -(\boldsymbol{\lambda} - \boldsymbol{\lambda}')\,,\, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}' \right\rangle \ge \rho \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}'\|_2^2$$

- Cauchy-Schwarz inequality

$$\|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|_2 \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}'\|_2 \ge \left| \left\langle -(\boldsymbol{\lambda} - \boldsymbol{\lambda}')\,,\, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}' \right\rangle \right| \ge \left\langle -(\boldsymbol{\lambda} - \boldsymbol{\lambda}')\,,\, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}' \right\rangle \ge \rho \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}'\|_2^2.$$

First term and last term gives $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}'\|_2 \le \dfrac{1}{\rho}\|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|_2$.

Augmented Lagrangian for composite function

- Consider problem

$$(\mathcal{P}_0) \ : \ \min_{\boldsymbol{x}} f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x})$$

where both f, g are convex functions and $\boldsymbol{A}$ is full rank.

- Rewrite $\mathcal{P}_0$ as

$$(\mathcal{P}_1) \ : \ \min_{\boldsymbol{x}} f(\boldsymbol{x}) + g(\boldsymbol{y}) \ \text{ s.t. } \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$$

- Solve $\mathcal{P}_1$ by Augmented Lagrangian

$$\min_{\boldsymbol{x},\boldsymbol{y}} \max_{\boldsymbol{\lambda}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + g(\boldsymbol{y}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle + \frac{\rho}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2.$$

Augmented Lagrangian for general problem

- Consider problem

$$(\mathcal{P}') \ : \ \min_{\boldsymbol{x}} f(\boldsymbol{x}) + g(\boldsymbol{y}) \ \text{ s.t. } \ \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c} = \boldsymbol{0}$$

  where both f, g are convex functions and $\boldsymbol{A}, \boldsymbol{B}$ are full rank.

- Solve $\mathcal{P}'$ by Augmented Lagrangian

$$\min_{\boldsymbol{x},\boldsymbol{y}} \max_{\boldsymbol{\lambda}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + g(\boldsymbol{y}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c} \rangle + \frac{\rho}{2} \|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c}\|_2^2.$$

Algorithm 6: Augmented Lagrangian algorithm

1 Initialize $\boldsymbol{x}_0, \boldsymbol{y}_0, \boldsymbol{\lambda}_0$
2 **for** $k = 1, 2, \ldots$ **do**
3 $\quad (\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1}) = \operatorname{argmin}_{\boldsymbol{x},\boldsymbol{y}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}_k)$ joint-minimization
4 $\quad \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha_k(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{y}_{k+1} + \boldsymbol{c})$ dual ascent
5 **end**

How ADMM is discovered

Algorithm 7: Augmented Lagrangian algorithm

Initialize $\boldsymbol{x}_0, \boldsymbol{y}_0, \boldsymbol{\lambda}_0$
for $k = 1, 2, \ldots$ **do**
  $(\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1}) = \underset{\boldsymbol{x},\boldsymbol{y}}{\operatorname{argmin}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}_k)$ joint-minimization
  $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha_k(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{y}_{k+1} + \boldsymbol{c})$ dual ascent
end

- Glowinski and Marrocco (1976) suggested to split $\boldsymbol{x}, \boldsymbol{y}$ in a coordinate descent fashion, i.e.

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \underset{\boldsymbol{x},\boldsymbol{y}}{\operatorname{argmin}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{y}_k, \boldsymbol{\lambda}_k) \\ \boldsymbol{y}_{k+1} &= \underset{\boldsymbol{x},\boldsymbol{y}}{\operatorname{argmin}} \mathcal{L}_\rho(\boldsymbol{x}_{k+1}, \boldsymbol{y}, \boldsymbol{\lambda}_k) \end{aligned}$$

  they found that this gives the best performance.
- **This is the moment ADMM was born.** It was discovered as an empirical heuristic with no theory support.
- Jonathan Eckstein: **ADMM does not approximate the ALM**.

ADMM algorithm

- For problem

$$(\mathcal{P}') \ : \ \min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}) + g(\boldsymbol{y}) \ \text{ s.t. } \ \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c} = \boldsymbol{0}$$

  where both f, g are convex functions and $\boldsymbol{A}, \boldsymbol{B}$ are full rank.

Algorithm 8: ADMM

Initialize $\boldsymbol{x}_0, \boldsymbol{y}_0, \boldsymbol{\lambda}_0, \rho$
for $k = 1, 2, \ldots$ **do**
  $\boldsymbol{x}_{k+1} = \operatorname{argmin}_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{y}_k, \boldsymbol{\lambda}_k)$
  $\boldsymbol{y}_{k+1} = \operatorname{argmin}_{\boldsymbol{y}} \mathcal{L}_\rho(\boldsymbol{x}_{k+1}, \boldsymbol{y}, \boldsymbol{\lambda}_k)$
  $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \rho(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{y}_{k+1} + \boldsymbol{c})$
end

Extra slide 1: what's the dual for $\min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}) + g(\boldsymbol{y})$ s.t. $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c} = \boldsymbol{0}$?

- Original problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) + g(\boldsymbol{y}) \text{ s.t. } \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c} = \boldsymbol{0} \quad f, g \text{ convex and } \boldsymbol{A}, \boldsymbol{B} \text{ full rank.}$$

- Augmented Lagrangian saddle point problem

$$\min_{\boldsymbol{x},\boldsymbol{y}} \max_{\boldsymbol{\lambda}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + g(\boldsymbol{y}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c} \rangle + \frac{\rho}{2} \|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c}\|_2^2.$$

- Un-augmented Lagrangian saddle point problem ($\rho = 0$)

$$\min_{\boldsymbol{x},\boldsymbol{y}} \max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + g(\boldsymbol{y}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c} \rangle.$$

- Dual on $\mathcal{L}$ (not $\mathcal{L}_\rho$)

$$\begin{aligned}
d(\boldsymbol{\lambda}) &:= \min_{\boldsymbol{x},\boldsymbol{y}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) \\
&= \min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}) + g(\boldsymbol{y}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} \rangle + \langle \boldsymbol{\lambda}, \boldsymbol{B}\boldsymbol{y} \rangle + \langle \boldsymbol{\lambda}, \boldsymbol{c} \rangle \\
&= -\Big(\max_{\boldsymbol{x},\boldsymbol{y}} -f(\boldsymbol{x}) - g(\boldsymbol{y}) - -\langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} \rangle - \langle \boldsymbol{\lambda}, \boldsymbol{B}\boldsymbol{y} \rangle - \langle \boldsymbol{\lambda}, \boldsymbol{c} \rangle \Big) \\
&= -\Big(\Big[\max_{\boldsymbol{x}} -f(\boldsymbol{x}) - \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} \rangle \Big] + \Big[\max_{\boldsymbol{y}} -g(\boldsymbol{y}) - \langle \boldsymbol{\lambda}, \boldsymbol{B}\boldsymbol{y} \rangle \Big] + \Big[- \langle \boldsymbol{\lambda}, \boldsymbol{c} \rangle \Big] \Big) \\
&= -\Big(\Big[\max_{\boldsymbol{x}} -f(\boldsymbol{x}) + \langle -\boldsymbol{A}^\top \boldsymbol{\lambda}, \boldsymbol{x} \rangle \Big] + \Big[\max_{\boldsymbol{y}} -g(\boldsymbol{y}) + \langle -\boldsymbol{B}^\top \boldsymbol{\lambda}, \boldsymbol{y} \rangle \Big] + \Big[- \langle \boldsymbol{\lambda}, \boldsymbol{c} \rangle \Big] \Big) \\
&= -\Big(\Big[f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) \Big] + \Big[g^*(\boldsymbol{B}^\top \boldsymbol{\lambda}) \Big] + \Big[- \langle \boldsymbol{\lambda}, \boldsymbol{c} \rangle \Big] \Big) \\
&= -f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) - g^*(\boldsymbol{B}^\top \boldsymbol{\lambda}) + \langle \boldsymbol{\lambda}, \boldsymbol{c} \rangle
\end{aligned}$$

Extra slide 2: why $\nabla_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda}) = \boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{y}_{k+1} + \boldsymbol{c}$?

- Recall subgradient for **concave** function:

$$\boldsymbol{q} \in \partial d(\boldsymbol{\lambda}) \quad \Longleftrightarrow \quad d(\boldsymbol{\gamma}) \leq d(\boldsymbol{\lambda}) + \langle \boldsymbol{q}, \boldsymbol{\gamma} - \boldsymbol{\lambda} \rangle \ \forall \boldsymbol{\gamma}.$$

- By definition of dual

$$\begin{aligned}
d(\boldsymbol{\gamma}) &= \min_{\boldsymbol{x},\boldsymbol{y}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\gamma}) \\
&= \min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}) + g(\boldsymbol{y}) + \langle \boldsymbol{\gamma}, \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c} \rangle \\
&= \min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}) + g(\boldsymbol{y}) + \langle \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c}, \boldsymbol{\gamma} \rangle \\
&= \min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}) + g(\boldsymbol{y}) + \langle \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c}, \boldsymbol{\gamma} + \boldsymbol{\lambda} - \boldsymbol{\lambda} \rangle \\
&= \min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}) + g(\boldsymbol{y}) + \langle \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c}, \boldsymbol{\lambda} \rangle + \langle \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c}, \boldsymbol{\gamma} - \boldsymbol{\lambda} \rangle \\
&\overset{*}{\leq} f(\boldsymbol{x}_{k+1}) + g(\boldsymbol{y}_{k+1}) + \langle \boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{y}_{k+1} + \boldsymbol{c}, \boldsymbol{\lambda} \rangle + \langle \boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{y}_{k+1} + \boldsymbol{c}, \boldsymbol{\gamma} - \boldsymbol{\lambda} \rangle \\
&\overset{**}{\leq} d(\boldsymbol{\lambda}) + \langle \boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{y}_{k+1} + \boldsymbol{c}, \boldsymbol{\gamma} - \boldsymbol{\lambda} \rangle
\end{aligned}$$

$*$: by definition of argmin.

$**$: by definition of $\boldsymbol{\lambda}$ in the ADMM algorithm and by the definition of dual.

Early histroy of ADMM

- Douglas-Rachford splitting[7] 1956
  - Another way to derive ADMM, much earlier than ADMM.
  - Requires knowledge of convex analysis and monotone operator theory.
  - It was first used to solve heat equation.
- Glowinski and Marrocco suggested to split $\boldsymbol{x}, \boldsymbol{y}$ in a coordinate descent fashion 1976
  - The moment ADMM was born.
  - Pure heuristic, no theory.
- Lions and Mercier 1979
- Gabay 1983
- Eckstein and Bertsekas 1992

[7] J. Douglas and H.H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," Transactions of the American Mathematical Society 82 (1956) 421-439.

A slide from Jonathan Eckstein (early contributor of ADMM)

Getting the History Right

- Douglas and Rachford had a different representation of the operations in their method, but equivalent
- However, the original Douglas-Rachford publication was only for linear operators
  - Applied to very specific linear operators related to the discretized 2-D heat equation
- Lions and Mercier (1979) generalized the idea from linear maps to general monotone set-valued maps (but kept the name)
- Gabay (1983) showed that the ADMM is just this idea applied to the dual of $\min f(x) + g(Mx)$
- The composition-of-nonexpansive maps interpretation may first be found (as an aside) in Lawrence & Spingarn (1987)
- E & Bertsekas (1992) contains some equivalent analysis and exploits the relationship with the proximal point algorithm to derive approximate and over-relaxed versions

Last page - summary

- The history of ADMM
  - Lagrangian multiplier — to handle constrained program
  - Augmented Lagrangian — to relax f strictly convex
  - ADMM: splitting primal variables — discovered empirically
- Not discussed / discuss next
  - Douglas-Rachford splitting
  - Convergence of ADMM

End of document