Chapter 5

Operator Splitting Methods

Convex Optimization for Computer Vision

SS 2016

Michael Moeller
Thomas Möllenhoff
Emanuel Laude
Computer Vision Group
Department of Computer Science
TU München

Recap and Motivation

- **Last 3 lectures**: PDHG method for minimizing structured convex problems

$$\min_{u \in \mathbb{R}^n} G(u) + F(Ku)$$

- Unintuitive overrelaxation, rather involved convergence analysis
- Next lectures: simple and unified convergence analysis of many different algorithms within a single approach
- Key ideas: monotone operators, fixed point iterations
- Give a new understanding of convex optimization algorithms

Relations

Notation

- A relation R on $\mathbb{R}^n$ is a subset of $\mathbb{R}^n \times \mathbb{R}^n$
- We will refer to it as a set-valued **operator** and overload the usual matrix notation

$$R(x) = Rx := \{y \in \mathbb{R}^n \mid (x, y) \in R\}.$$

- If Rx is a singleton or empty for all x, then R is a function (or single-valued operator) with domain

$$\text{dom}(R) := \{x \in \mathbb{R}^n \mid Rx \neq \emptyset\}$$

- Abuse of notation: identify singleton $\{x\}$ with x, i.e., write $Rx = y$ instead of $Rx \ni y$ if R is function
- Concept: identifying functions with their *graph*

Some Examples

- Empty relation: $\emptyset$
- Identity: $I := \{(u, u) \mid u \in \mathbb{R}^n\}$
- Zero: $0 := \{(u, 0) \mid u \in \mathbb{R}^n\}$
- Gradient relation:

$$\nabla E := \{(u, \nabla E(u)) \mid u \in \mathbb{R}^n\}$$

- Subdifferential relation:

$$\partial E := \{(u, g) \mid u \in \mathrm{dom}(E), E(v) \geq E(u) + \langle g, v - u \rangle, \forall v \in \mathbb{R}^n\}$$

- Another possible view: think of relations as a set valued functions, e.g., $\partial E : \mathbb{R}^n \to \mathcal{P}(\mathbb{R}^n)$

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Our Goal

Solve generalized equation (inclusion) problem

$$0 \in R(u)$$

i.e., find $u \in \mathbb{R}^n$ such that $(u, 0) \in R$.

Examples:

- Set $R = \partial E$, then the goal is to find $0 \in \partial E(u)$
- This are just the optimality conditions of our prototypical optimization problem:

$$\arg\min_{u \in \mathbb{R}^n} \; E(u)$$

- Finding saddle-points $(\tilde{u}, \tilde{p})$ of

$$PD(u,p) = G(u) - F^*(p) + \langle Ku, p \rangle$$

corresponds to the inclusion problem

$$0 \in \begin{bmatrix} \partial G & K^T \\ -K & \partial F^* \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix}$$

Operations on Relations

- Inverse $R^{-1} = \{(y, x) \mid (x, y) \in R\}$
  - Exists for *any* relation
  - Reduces to inverse function when R is injective function
- Addition $R + S = \{(x, y + z) \mid (x, y) \in R, (x, z) \in S\}$
- Scaling $\lambda R = \{(x, \lambda y) \mid (x, y) \in R\}$
- Resolvent $J_{\lambda R} := (I + \lambda R)^{-1}$

Examples:

- $I + \lambda R = \{(x, x + \lambda y) \mid (x, y) \in R\}$
- $J_R = \{(x + \lambda y, x) \mid (x, y) \in R\}$
- E closed, proper, convex: $(\partial E)^{-1} = \partial E^*$

$\rightarrow$ **Draw a picture for** $E(u) = |u|$

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Monotone Operators

Monotone Operators

Definition

The set-valued operator $T \subset \mathbb{R}^n \times \mathbb{R}^n$ is called **monotone** if

$$\langle u - v, Tu - Tv \rangle \geq 0, \; \forall u, v \in \mathbb{R}^n. \qquad \text{Notation}^1$$

An operator T is called **maximally monotone** if it is not contained in any other monotone operator.

- Maximal monotonicity is an important technical detail, but we will be sloppy about it for the rest of the course

Examples of monotone operators:

- Monotonically non-decreasing functions $T : \mathbb{R} \to \mathbb{R}$
- Any positive semi-definite matrix A: $\langle Ax - Ay, x - y \rangle \geq 0$
- Subdifferential of a convex function ∂f
- Proximity operators of convex functions $\text{prox}_{\tau f} : \mathbb{R}^n \to \mathbb{R}^n$

[1] This is again abuse of notation for $\langle u - v, p - q \rangle \geq 0, \; \forall p \in Tu, \forall q \in Tv$

Monotone Operators

Calculus rules (exercise):

- T monotone, $\lambda \geq 0 \Rightarrow \lambda T$ monotone
- T monotone $\Rightarrow T^{-1}$ monotone
- R, S monotone, $\lambda \geq 0 \Rightarrow R + \lambda S$ is monotone

Some important definitions/properties:

- Lipschitz operators (and in particular nonexpansive operators) are single-valued (functions)
- x is called *fixed point* of operator T if $x = Tx$
- If F is nonexpansive (Lipschitz constant $L \leq 1$) and $\operatorname{dom} T = \mathbb{R}^n$ then the set of fixed points $(I - F)^{-1}(0)$ is closed and convex **(exercise)**

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Resolvent and Cayley Operators

- Let $T \subset \mathbb{R}^n \times \mathbb{R}^n$ be set-valued operator
- The *resolvent operator* of T is given as $J_{\lambda T} := (I + \lambda T)^{-1}$
- Special case: $T = \partial f$, $J_{\lambda \partial f}$ is proximal operator of f
- From previous slide: resolvent is monotone if T is monotone
- The *Cayley operator* (or reflection operator) of T is defined as $C_{\lambda T} := 2J_{\lambda T} - I$

Facts:

- $0 \in Tx$ if and only if $x = J_{\lambda T}x = C_{\lambda T}x$
- If T is monotone, then $J_{\lambda T}$ and $C_{\lambda T}$ are nonexpansive

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Fixed Point Iterations

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

The Main Algorithm

- Recall that $u \in \mathbb{R}^n$ is fixed point of $F : \mathbb{R}^n \to \mathbb{R}^n$, if $u = Fu$
- The main algorithm of this chapter is the *fixed point* or *Picard iteration* for some given $u^0 \in \mathbb{R}^n$:

$$u^{k+1} = Fu^k, \qquad k = 0, 1, 2, \dots$$

- We will see that many important convex optimization algorithms can be written in this form
- Allows simple and unified analysis

Iteration of Contraction Mappings

Contraction Mapping Theorem

Suppose that $F : \mathbb{R}^n \to \mathbb{R}^n$ is a contraction with Lipschitz constant $L < 1$. Then the fixed point iteration

$$u^{k+1} = Fu^k,$$

also called contraction mapping algorithm, converges to the unique fixed point of F.

→ Proof: see literature[2]

- Example: the gradient method can be written as
$$u^{k+1} = (I - \tau \nabla E)u^k$$
- Suppose E is m-strongly convex and L-smooth, then $I - \tau \nabla E$ is Lipschitz with $L_{GM} = \max\{|1 - \tau m|, |1 - \tau L|\}$
- $I - \tau \nabla E$ is contractive for $\tau \in (0, 2/L)$

[2] This theorem is also known as the Banach fixed point theorem.

Iteration of Averaged Nonexpansive Mappings

- Recall that a mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is called *nonexpansive* if it is Lipschitz with constant $L \leq 1$.
- Fixed point iteration of nonexpansive mapping doesn't necessarily converge (example: rotation, reflection)
- The mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is called *averaged* if $F = (1 - \theta)I + \theta T$, for some nonexpansive operator T and $\theta \in (0, 1)$

Theorem: Krasnosel'skii-Mann

Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be averaged, and denote the (non-empty) set of fixed points of F as U. Then the sequence (u^k) produced by the iteration

$$u^{k+1} = Fu^k$$

converges to a fixed point $u^* \in U$, i.e., $u^k \to u^*$.

$\to$ Proof: board!

Example: gradient method

- Assume E is L-smooth but not strongly convex
- Possible to show that the operator $(I - \tau\nabla E)$ is Lipschitz continuous with parameter $L_{GM} = \max\{1, |1 - \tau L|\}$
- For $0 < \tau \leq 2/L$, this operator is nonexpansive
- It is also averaged for $0 < \tau < 2/L$ since

$$(I - \tau\nabla E) = (1 - \theta)I + \theta(I - (2/L)\nabla E),$$

  with $\theta = \tau L/2 < 1$.
- Hence, we get convergence of the gradient descent method from the previous theorem

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Proximal Point Algorithm

The Proximal Point Algorithm

- Recall our original goal of finding $u \in \mathbb{R}^n$ with

$$0 \in Tu,$$

  for $T \subset \mathbb{R}^n \times \mathbb{R}^n$ monotone.
- We have seen that fixed points of resolvent operator $J_{\lambda T}$ are the zeros of T

Definition: Proximal Point Algorithm (PPA) [3]

Given some maximally monotone operator $T \subset \mathbb{R}^n \times \mathbb{R}^n$, and some sequence $(\lambda_k) > 0$. Then the iteration

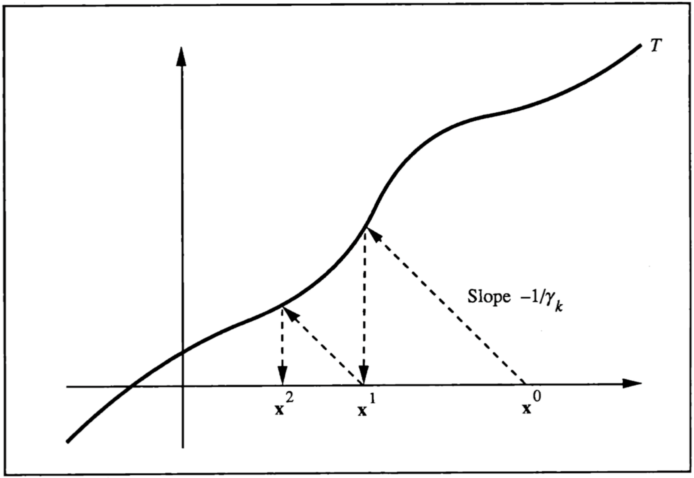
$$u^{k+1} = (I + \lambda_k T)^{-1} u^k,$$

is called the *proximal point algorithm*.

[3] R. T. Rockafellar, Monotone Operators and the Proximal Point Algorithm, SIAM J. Control and Optimization, 1976

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Intuition of the Proximal Point Algorithm [4]



[4] Eckstein, Splitting methods for monotone operators with applications to parallel optimzation, 1989, pp. 42

Convergence of Proximal Point Algorithm

- The resolvent $J_{\lambda T} = (I + \lambda T)^{-1}$ is an averaged operator
- To see this, consider the *reflection* or *Cayley* operator

$$C_{\lambda T} := 2J_{\lambda T} - I \Leftrightarrow J_{\lambda T} = \frac{1}{2} I + \frac{1}{2} C_{\lambda T}$$

- Hence $J_{\lambda T}$ is averaged with $\theta = \frac{1}{2}$, as we have seen in the last lecture that $C_{\lambda T}$ is nonexpansive
- Proximal Point algorithm converges as it is fixed point iteration of averaged operator

PDHG Revisited

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

PDHG as Proximal Point Method

- Remember that for convex-concave saddle point problems

$$PD(u,p) = G(u) - F^*(p) + \langle Ku, p\rangle$$

we have the following:

$$(\tilde{u},\tilde{p}) = \arg\operatorname{minmax}_{u,p} PD(u,p) \Leftrightarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \underbrace{\begin{bmatrix} \partial G(\tilde{u}) + K^T\tilde{p} \\ -K\tilde{u} + \partial F^*(\tilde{p}) \end{bmatrix}}_{=:T(\tilde{u},\tilde{p})}$$

- For convex F^* and G, T is monotone
- Idea: use the proximal point to find zero of T
- Stack primal and dual variables into vector $z = (u,p)^T$:

$$z^{k+1} = (I + \lambda T)^{-1} z^k \;\Leftrightarrow\; z^k - z^{k+1} \in \lambda T z^{k+1}$$

- Plugging things in yields

$$u^k - u^{k+1} \in \lambda \partial G(u^{k+1}) + \lambda K^T p^{k+1}$$
$$p^k - p^{k+1} \in \lambda \partial F^*(p^{k+1}) - \lambda K u^{k+1}$$

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

PDHG as Proximal Point Method

- Reformulating the following

$$0 \in \lambda^{-1} \begin{bmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{bmatrix} + \underbrace{\begin{bmatrix} \partial G(u^{k+1}) + K^T p^{k+1} \\ \partial F^*(p^{k+1}) - K u^{k+1} \end{bmatrix}}_{=: T(\tilde{u}, \tilde{p})}$$

leads to:

$$\begin{aligned} u^{k+1} &= (I + \lambda \partial G)^{-1} (u^k - \lambda K^T p^{k+1}) \\ &= \operatorname{prox}_{\lambda G}(u^k - \lambda K^T p^{k+1}) \\ p^{k+1} &= (I + \lambda \partial F^*)^{-1} (p^k + \lambda K u^{k+1}) \\ &= \operatorname{prox}_{\lambda F^*}(p^k + \lambda K u^{k+1}) \end{aligned}$$

- Almost looks like the PDHG method, step size λ
- **Problem:** cannot implement this algorithm, since updates in u^{k+1} and p^{k+1} depend on each other

PDHG as Proximal Point Method

- Consider the following:

$$0 \in M \begin{bmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{bmatrix} + \underbrace{\begin{bmatrix} \partial G(u^{k+1}) + K^T p^{k+1} \\ \partial F^*(p^{k+1}) - K u^{k+1} \end{bmatrix}}_{=: T(\tilde{u}, \tilde{p})}$$

- Step size $M \in \mathbb{R}^{(n+m)\times(n+m)}$ is now a matrix
- Take the following choice

$$M = \begin{bmatrix} \frac{1}{\tau} I & -K^T \\ -\theta K & \frac{1}{\sigma} I \end{bmatrix}$$

- Allows to recover PDHG as proximal point algorithm (PPA)

$$u^{k+1} = \text{prox}_{\tau G}(u^k - \tau K^T p^k),$$
$$p^{k+1} = \text{prox}_{\sigma F^*}(p^k + \sigma K(u^{k+1} + \theta(u^{k+1} - u^k)))$$

- This is called generalized or customized PPA:

$$0 \in M(z^{k+1} - z^k) + Tz^{k+1} \;\Leftrightarrow\; z^{k+1} = (M + T)^{-1} M z^k$$

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Convergence of Customized Proximal Point Method

- For symmetric, positive definite M, we can write $M = L^T L$, L invertible (Cholesky decomposition)
- Apply classical PPA to operator $T' = L^{-T} \circ T \circ L^{-1}$

$$y^{k+1} = (I + L^{-T} \circ T \circ L^{-1})^{-1} y^k$$

- T (maximally) monotone $\Rightarrow L^{-T} \circ T \circ L^{-1}$ (maximally) monotone [5]
- Define $Lx = y$, then $0 \in (L^{-T} \circ T \circ L^{-1})y \Leftrightarrow 0 \in Tx$
- Writing out the algorithm in terms of x yields

$$0 \in M(x^{k+1} - x^k) + Tx^{k+1}$$

- Hence customized PPA inherits convergence from classical proximal point

[5] Bauschke, Combettes, Convex Analysis and Monotone Operator Theory in Hilbert Spaces, Theorem 24.5

Convergence of PDHG

- When is the step size matrix symmetric positive definite?

$$M = \begin{bmatrix} \frac{1}{\tau} I & -K^T \\ -\theta K & \frac{1}{\sigma} I \end{bmatrix}$$

- Step size requirement for PDHG is $\tau\sigma \left\| K \right\|^2 < 1$, $\tau\sigma > 0$

Lemma (Pock-Chambolle-2011 [6])

Let $\theta = 1$, T and Σ symmetric positive definite maps satisfying

$$\left\| \Sigma^{\frac{1}{2}} K \mathrm{T}^{\frac{1}{2}} \right\|^2 < 1,$$

then the block matrix

$$M = \begin{bmatrix} \mathrm{T}^{-1} & -K^T \\ -\theta K & \Sigma^{-1} \end{bmatrix}$$

is symmetric and positive definite.

[6] T. Pock, A. Chambolle, Diagonal Preconditioning for first-order primal-dual algorithms in convex optimization, ICCV 2011

Summary

- Customized proximal point algorithms yield a whole family of methods, many choices of M are concievable

$$0 \in M(z^{k+1} - z^k) + Tz^{k+1}$$

- PDHG corresponds to one particular choice of M
- Overrelaxation with $\theta = 1$ required to make M symmetric
- Convergence follows from convergence of classical proximal point algorithm
- Classical proximal point converges as it is fixed point iteration of averaged operator
- **Next lecture:** Douglas-Rachford splitting and ADMM

Organizational Remarks

Exams:

- **Important:** Registration deadline 30.06. in TUMonline!
- Exam (oral): 18.07. and 19.07.
- Repeat exam (oral): 05.10. and 06.10.
- Sign up for timeslots in exercise class on Friday 17.06.

Remaining lectures:

- Next Monday 20.06. hints for getting started with the optimization challenge!
- 22.06. Some practical considerations of PDHG/ADMM
- 27.06. - 01.07. no lecture / exercises, repeat and review what you have learned!
- 04.07. - 11.07. Miscellaneous topics on modifications and accelerations, open research questions/challenges
- Last lecture on 13.07. repeat of content, questions

Douglas-Rachford Splitting

Motivation

- Last lecture: proximal point algorithm for finding the zero of a monotone operator T

$$0 \in Tu \;\Leftrightarrow\; u = (I + \lambda T)^{-1} u$$

- Often the resolvent $J_{\lambda T} := (I + \lambda T)^{-1}$ is hard to compute
- One remedy: matrix-valued step-size / customized PPA

$$u^{k+1} = (M + T)^{-1} M u^k$$

- Another possibility are **splitting methods**
- They exploit further structure of the problem:

$$T = A + B$$

- Resolvents $J_{\lambda A} = (I + \lambda A)^{-1}$ and $J_{\lambda B} = (I + \lambda B)^{-1}$ can be more easily evaluated than $J_{\lambda T}$

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Splitting methods

- $T = A + B$, A and B maximal monotone
- Cayley operators $C_A = 2J_A - I$ and $C_B = 2J_A - I$ are nonexpansive
- Composition $C_A C_B$ also nonexpansive
- **Main result:** ($\rightarrow$ board!)

$$0 \in Au + Bu \;\Leftrightarrow\; C_A C_B v = v,\; u = J_B v$$

- Hence, solutions can be found from fixed point of the operator $C_A C_B$

$\rightarrow$ **Draw a picture for** $T = \partial\iota_{C_1} + \partial\iota_{C_2}$**!**

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Splitting Methods

- *Peaceman-Rachford* splitting is undamped iteration

$$v^{k+1} = C_A C_B v^k$$

- Doesn't converge in the general case, needs either C_A or C_B to be a contraction
- *Douglas-Rachford* splitting [7] is the damped iteration

$$v^{k+1} = \left(\frac{1}{2}I + \frac{1}{2}C_A C_B\right) v^k,$$

- Recover solution by $u^* = J_B v^*$
- Always converges if there exists a solution $0 \in Au^* + Bu^*$, since it's fixed point iteration of averaged operator

[7] J. Douglas, H. H. Rachford, On the numerical solution of heat conduction problems in two and three space variables. Transactions of the AMS, 1956.

Douglas-Rachford Splitting (DRS)

- The Douglas-Rachford iteration $v^{k+1} = \left(\frac{1}{2}I + \frac{1}{2}C_A C_B\right) v^k$ can be written as

$$
\begin{aligned}
u_b^{k+1} &= J_B(v^k), \\
\tilde{v}^{k+1} &= 2u_b^{k+1} - v^k, \\
u_a^{k+1} &= J_A(\tilde{v}^{k+1}), \\
v^{k+1} &= v^k + u_a^{k+1} - u_b^{k+1}.
\end{aligned}
$$

- u_a^k and u_b^k can be thought of estimates to a solution
- v^k running sum of residuals, drives u_a^k and u_b^k together

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Application to Convex Optimization

- Let's apply DRS to minimize

$$\min_{u \in \mathbb{R}^n} \; G(u) + F(u)$$

- $G : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, $F : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ closed, proper, cvx.
- Optimality conditions (assuming $\text{ri}(\text{dom}G) \cap \text{ri}(\text{dom}F) \neq \emptyset$):

$$0 \in \tau \partial G(u) + \tau \partial F(u)$$

- Find zero of $T = A + B$, $A = \tau \partial F$, $B = \tau \partial G$
- The algorithm becomes (after slight simplifications):

$$\begin{aligned} u^{k+1} &= \text{prox}_{\tau G}(v^k), \\ v^{k+1} &= \text{prox}_{\tau F}(2u^{k+1} - v^k) + v^k - u^{k+1}. \end{aligned}$$

Reformulation of DRS

- We can rewrite the step in v^{k+1} using Moreau's Identity

$$
\begin{aligned}
u^{k+1} &= \text{prox}_{\tau G}(v^k), \\
v^{k+1} &= \text{prox}_{\tau F}(2u^{k+1} - v^k) + v^k - u^{k+1} \\
&= u^{k+1} + \tau \text{prox}_{(1/\tau)F^*}((2u^{k+1} - v^k)/\tau)
\end{aligned}
$$

- Introduce variable $p^k = \frac{u^k - v^k}{\tau} \Leftrightarrow v^k = u^k - \tau p^k$, $\sigma = 1/\tau$

$$
\begin{aligned}
u^{k+1} &= \text{prox}_{\tau G}(u^k - \tau p^k), \\
p^{k+1} &= \text{prox}_{\sigma F^*}(p^k + \sigma(2u^{k+1} - u^k))
\end{aligned}
$$

- Looks familiar? :-)
- Applying DRS on the primal problem $\min_u G(u) + F(u)$ is equivalent to PDHG!

Optimization Problems with Compositions

- Ideally we'd like to solve problems of the form

$$\min_u \; G(u) + F(w), \quad \text{s.t.} \quad w = Ku$$

- In many applications we would actually like to minimize

$$\min_u \; G(u) + \sum_{i=1}^{N} F_i(K_i u)$$

- Rewrite using trick:

$$w = \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix}, K = \begin{bmatrix} K_1 \\ \dots \\ K_N \end{bmatrix}, \quad \rightarrow F(w) = \sum_{i=1}^{N} F_i(w_i)$$

- Virtually any convex optimization problem fits into this form
- Even problems looking very complicated at first glance can be split up into many simple substeps

Option 1: Graph Projection Splitting

- We want to minimize for $K : \mathbb{R}^n \to \mathbb{R}^m$

$$\min_{u \in \mathbb{R}^n, w \in \mathbb{R}^m} \; G(u) + F(w) \quad \text{s.t.} \quad Ku = w$$

- Rewrite problem using $(u, w) \in \mathbb{R}^{n+m}$ as

$$\min_{u,w} \; \tilde{G}(u, w) + \tilde{F}(u, w)$$

- Set $\tilde{G}(u, w) = G(u) + F(w)$
- Set $\tilde{F}(u, w) = \begin{cases} 0, & \text{if } Ku = w \\ \infty, & \text{else.} \end{cases}$
- Proximal operator for $\tilde{G}$ is simple if proximal operators for F and G are simple
- Proximal operator for $\tilde{F}$ is projection onto the graph of $Ku = w$ (solving a least squares problem)

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Option 1: Graph Projection Splitting

- Iterations can be written as [8]

$$(u^{k+1/2}, w^{k+1/2}) = \left(\text{prox}_G(u^k - \tilde{u}^k), \text{prox}_F(w^k - \tilde{w}^k)\right),$$

$$(u^{k+1}, w^{k+1}) = \Pi(u^{k+1/2} + \tilde{u}^k, w^{k+1/2} + \tilde{w}^k),$$

$$(\tilde{u}^{k+1}, \tilde{w}^{k+1}) = (\tilde{u}^k + u^{k+1/2} - u^{k+1}, \tilde{w}^k + w^{k+1/2} - w^{k+1}).$$

- Projection is given as:

$$\Pi(c,d) = A^{-1} \begin{bmatrix} c + A^T d \\ 0 \end{bmatrix}, A = \begin{bmatrix} I & K^T \\ K & -I \end{bmatrix}$$

- Can use (preconditioned) conjugate gradient to approximately compute projection
- Important: warm-start linear system solver with solution from previous iteration
- Other possibility: factorization caching

[8] N. Parikh, S. Boyd, Block Splitting for Distributed Optimization, 2014

Option 2: DRS for Problems with Compositions

- Consider the dual problem to $\min_u \; G(u) + F(Ku)$

$$\min_p \; G^*(-K^*p) + F^*(p) = (G^* \circ -K^*)(p) + F^*(p)$$

- Applying DRS yields the following:

$$\begin{aligned} u^{k+1} &= \text{prox}_{\sigma(G^* \circ -K^*)}(v^k), \\ v^{k+1} &= \text{prox}_{\sigma F^*}(2u^{k+1} - v^k) + v^k - u^{k+1} \end{aligned}$$

- Reorder slightly with new variable w^{k+1}

$$\begin{aligned} u^{k+1} &= \text{prox}_{\sigma(G^* \circ -K^*)}(v^k), \\ p^{k+1} &= \text{prox}_{\sigma F^*}(2u^{k+1} - v^k), \\ v^{k+1} &= p^{k+1} + v^k - u^{k+1} \end{aligned}$$

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Option 2: DRS for Problems with Compositions

- The prox involving the composition is given by:

$$\text{prox}_{\sigma(G^* \circ -K^*)}(v) = v + \sigma K \operatorname*{argmin}_u G(u) + \frac{\sigma}{2} \left\| Ku + \frac{v}{\sigma} \right\|^2$$

- Often expensive or difficult to evaluate due to the Ku-term
- Iteration can be written as

$$\begin{aligned}
u^{k+1} &= \operatorname*{argmin}_u G(u) + \frac{\sigma}{2} \left\| Ku + \frac{v^k}{\sigma} \right\|^2, \\
\tilde{u}^{k+1} &= v^k + \sigma K u^{k+1}, \\
p^{k+1} &= \text{prox}_{\sigma F^*}(2\tilde{u}^{k+1} - v^k), \\
v^{k+1} &= p^{k+1} + v^k - \tilde{u}^{k+1}
\end{aligned}$$

- Alternatively this can be simplified to

$$\begin{aligned}
u^{k+1} &= \operatorname*{argmin}_u G(u) + \frac{\sigma}{2} \left\| Ku + \frac{v^k}{\sigma} \right\|^2, \\
p^{k+1} &= \text{prox}_{\sigma F^*}(v^k + 2\sigma K u^{k+1}), \\
v^{k+1} &= p^{k+1} - \sigma K u^{k+1}
\end{aligned}$$

Option 2: DRS for Problems with Compositions

- Even more simple:

$$u^{k+1} = \operatorname*{argmin}_u G(u) + \frac{\sigma}{2} \left\| Ku + \frac{p^k - \sigma K u^k}{\sigma} \right\|^2,$$
$$p^{k+1} = \operatorname{prox}_{\sigma F^*}(p^k + \sigma K(2u^{k+1} - u^k)),$$

- Optimality conditions for the iterates:

$$0 \in \partial G(u^{k+1}) + \sigma K^T(Ku^{k+1} + \frac{1}{\sigma}(p^k - \sigma K u^k))$$
$$0 \in \partial F^*(p^{k+1}) + \frac{1}{\sigma}(p^{k+1} - p^k - \sigma K 2u^{k+1} + \sigma K u^k)$$

- Adding and substracting $K^T p^{k+1}$ to first line yields

$$0 \in \partial G(u^{k+1}) + K^T p^{k+1} + \sigma K^T K(u^{k+1} - u^k) - K^T(p^{k+1} - p^k)$$
$$0 \in \partial F^*(p^{k+1}) - Ku^{k+1} - K(u^{k+1} - u^k) + \frac{1}{\sigma}(p^{k+1} - p^k)$$

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Relation to PDHG

- Previous iterations can be written as PPA, $z = (u, p)^T$:

$$0 \in \underbrace{\begin{bmatrix} \partial G & K^T \\ -K & \partial F^* \end{bmatrix} \begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix}}_{Tz^{k+1}} + \underbrace{\begin{bmatrix} \frac{1}{\tau} I & -K^T \\ -K & \frac{1}{\sigma} I \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{bmatrix}}_{z^{k+1} - z^k}$$

- Matrix M only positive semidefinite, our convergence result for Proximal Point algorithm does not apply directly
- PDHG with $\theta = 1$ can be seen as inexact/approximative DRS,

$$\sigma K^T K \approx \frac{1}{\tau} I$$

- Often makes iterations much cheaper
- For semi-orthogonal ($K^T K = \nu I$) this approximation is exact

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Alternating Direction Method of Multipliers (ADMM)

- Recall this formulation

$$
\begin{aligned}
u^{k+1} &= \operatorname*{argmin}_u G(u) + \frac{\sigma}{2} \left\| Ku + \frac{v^k}{\sigma} \right\|^2, \\
p^{k+1} &= \operatorname{prox}_{\sigma F^*}(v^k + 2\sigma K u^{k+1}), \\
v^{k+1} &= p^{k+1} - \sigma K u^{k+1}
\end{aligned}
$$

- Apply Moreau's identity to step in p^{k+1}

$$
\begin{aligned}
u^{k+1} &= \operatorname*{argmin}_u G(u) + \frac{\sigma}{2} \left\| Ku + \frac{v^k}{\sigma} \right\|^2, \\
p^{k+1} &= v^k + 2\sigma K u^{k+1} - \sigma \operatorname{prox}_{\sigma F}\left(\frac{v^k}{\sigma} + 2Ku^{k+1}\right), \\
v^{k+1} &= p^{k+1} - \sigma K u^{k+1}
\end{aligned}
$$

Alternating Direction Method of Multipliers (ADMM)

- Make new variable for $\text{prox}_{\sigma F}$-step, write prox as argmin:

$$
\begin{aligned}
u^{k+1} &= \operatorname*{argmin}_u G(u) + \frac{\sigma}{2} \left\| Ku + \frac{v^k}{\sigma} \right\|^2, \\
w^{k+1} &= \operatorname*{argmin}_w F(w) + \frac{\sigma}{2} \left\| w - \frac{v^k}{\sigma} - 2Ku^{k+1} \right\|^2, \\
p^{k+1} &= v^k + 2\sigma K u^{k+1} - \sigma w^{k+1}, \\
v^{k+1} &= p^{k+1} - \sigma K u^{k+1}
\end{aligned}
$$

- Replacing the variable v^k in the u^{k+1} update yields

$$
u^{k+1} = \operatorname*{argmin}_u G(u) + \frac{\sigma}{2} \left\| Ku + \frac{p^k - \sigma K u^k}{\sigma} \right\|^2,
$$

Alternating Direction Method of Multipliers (ADMM)

- Replace variable p^k in all update steps

$$u^{k+1} = \underset{u}{\operatorname{argmin}}\, G(u) + \frac{\sigma}{2}\left\|Ku + \frac{v^{k-1} + \sigma K u^k - \sigma w^k}{\sigma}\right\|^2,$$

$$w^{k+1} = \underset{w}{\operatorname{argmin}}\, F(w) + \frac{\sigma}{2}\left\|w - \frac{v^k}{\sigma} - 2Ku^{k+1}\right\|^2,$$

$$v^{k+1} = v^k + \sigma(Ku^{k+1} - w^{k+1})$$

- Rewrite as:

$$u^{k+1} = \underset{u}{\operatorname{argmin}}\, G(u) + \frac{\sigma}{2}\left\|Ku - w^k + \frac{v^{k-1} + \sigma K u^k}{\sigma}\right\|^2,$$

$$w^{k+1} = \underset{w}{\operatorname{argmin}}\, F(w) + \frac{\sigma}{2}\left\|w - Ku^{k+1} - \frac{v^k + \sigma K u^{k+1}}{\sigma}\right\|^2,$$

$$v^{k+1} = v^k + \sigma(Ku^{k+1} - w^{k+1})$$

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Alternating Direction Method of Multipliers (ADMM)

- Using the following fact we can further rewrite the updates:

$$\operatorname*{argmin}_a \ \frac{\sigma}{2}\left\|a - \frac{b}{\sigma}\right\|^2 = \operatorname*{argmin}_a \ -\langle a, b\rangle + \frac{\sigma}{2}\|a\|^2$$

- Pulling terms of the squared norm:

$$u^{k+1} = \operatorname*{argmin}_u G(u) + \langle Ku, v^{k-1} + \sigma K u^k\rangle + \frac{\sigma}{2}\left\|Ku - w^k\right\|^2,$$

$$w^{k+1} = \operatorname*{argmin}_w F(w) - \langle w, v^k + \sigma K u^{k+1}\rangle + \frac{\sigma}{2}\left\|w - Ku^{k+1}\right\|^2,$$

$$v^{k+1} = v^k + \sigma(Ku^{k+1} - w^{k+1})$$

- Reintroduce $p^{k+1} = v^k + \sigma K u^{k+1}$, can be rewritten as:

$$u^{k+1} = \operatorname*{argmin}_u G(u) + \langle Ku, p^k\rangle + \frac{\sigma}{2}\left\|Ku - w^k\right\|^2,$$

$$w^{k+1} = \operatorname*{argmin}_w F(w) - \langle w, p^{k+1}\rangle + \frac{\sigma}{2}\left\|w - Ku^{k+1}\right\|^2,$$

$$p^{k+1} = p^k + \sigma(Ku^{k+1} - w^k)$$

Alternating Direction Method of Multipliers (ADMM)

- Let $\bar{w}^{k+1} = w^k$:

$$u^{k+1} = \operatorname*{argmin}_u G(u) + \langle Ku, p^k \rangle + \frac{\sigma}{2} \left\| Ku - \bar{w}^{k+1} \right\|^2,$$

$$\bar{w}^{k+2} = \operatorname*{argmin}_w F(w) - \langle w, p^{k+1} \rangle + \frac{\sigma}{2} \left\| w - Ku^{k+1} \right\|^2,$$

$$p^{k+1} = p^k + \sigma(Ku^{k+1} - \bar{w}^{k+1})$$

- Change order of first two iterates:

$$\bar{w}^{k+1} = \operatorname*{argmin}_w F(w) - \langle w, p^k \rangle + \frac{\sigma}{2} \left\| w - Ku^k \right\|^2,$$

$$u^{k+1} = \operatorname*{argmin}_u G(u) + \langle Ku, p^k \rangle + \frac{\sigma}{2} \left\| Ku - \bar{w}^{k+1} \right\|^2,$$

$$p^{k+1} = p^k + \sigma(Ku^{k+1} - \bar{w}^{k+1})$$

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Alternating Direction Method of Multipliers (ADMM)

- Final update equations:

$$w^{k+1} = \underset{w}{\operatorname{argmin}}\, F(w) - \langle w, p^k \rangle + \frac{\sigma}{2} \left\| w - Ku^k \right\|^2,$$

$$u^{k+1} = \underset{u}{\operatorname{argmin}}\, G(u) + \langle Ku, p^k \rangle + \frac{\sigma}{2} \left\| Ku - w^{k+1} \right\|^2,$$

$$p^{k+1} = p^k + \sigma(Ku^{k+1} - w^{k+1})$$

- Alternating minimization of the **augmented Lagrangian**:

$$L^{\tau}_{\text{aug}}(u, w, p) = G(u) + F(w) + \langle p, Ku - w \rangle + \frac{\tau}{2} \left\| Ku - w \right\|^2$$

- The method in this form is called Alternating Direction Method of Multipliers (ADMM)
- It has gained enormous popularity recently [9], over 3458 citations in 5 years

[9] Boyd et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, 2011

Conclusion

- Splitting methods split problem into simpler subproblems
- Many other splitting approaches exist that can explicitly handle differentiable functions (Forward-Backward, Forward-Backward-Forward, Davis-Yin, ...)
- Many relations exist between the primal-dual algorithms, often special cases of one another
- Depending on the problem structure, better to use either Graph Projection/DRS/ADMM or PDHG (more next week!)
- **Rule of thumb:** Graph Projection/DRS/ADMM few expensive iterations, PDHG many cheap iterations

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Recalling customized proximal point methods

Method review

Primal problem

$$\min_u G(u) + F(Ku)$$

Primal-Dual form

$$\min_u \max_p G(u) + \rangle Ku, p \langle - F^*(p)$$

Primal dual hybrid gradient

$$\begin{aligned} p^{k+1} &= \text{prox}_{\sigma F^*}(p^k + \sigma K \bar{u}^k), \\ u^{k+1} &= \text{prox}_{\tau G}(u^k - \tau K^* p^{k+1}), \qquad \text{(PDHG)} \\ \bar{u}^{k+1} &= 2u^{k+1} - u^k. \end{aligned}$$

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Method review

Primal problem

$$\min_u G(u) + F(Ku)$$

Augmented Lagrangian

$$L_{\text{aug}}^{\tau}(u, w, p) = G(u) + F(w) + \langle p, Ku - w \rangle + \frac{\tau}{2} \|Ku - w\|^2$$

Alternating directions method of multipliers (ADMM) on primal

$$\begin{aligned} u^{k+1} &= \operatorname*{argmin}_u L_{\text{aug}}^{\tau}(u, w^k, p^k), \\ w^{k+1} &= \operatorname*{argmin}_w L_{\text{aug}}^{\tau}(u^{k+1}, w, p^k), \\ p^{k+1} &= p^k + \tau \left(Ku^{k+1} - w^{k+1} \right) \end{aligned} \qquad \text{(ADMM)}$$

= Douglas-Rachford Splitting (DRS) on the dual.

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Various reformulations!

Note that various reformulations lead to many different algorithms under the same name!

For example:

- $\tilde{F}(u) := G^*(-K^*u)$, $\tilde{G}(u) = F^*(u)$,

$$\min_u \tilde{G}(u) + \tilde{F}(Ku)$$

  is the dual problem.

- $\tilde{K} = (K, -I)$, $\tilde{u} = (u, w)$, $\tilde{F} = \delta_0$, $\tilde{G}(\tilde{u}) = F(w) + G(u)$:

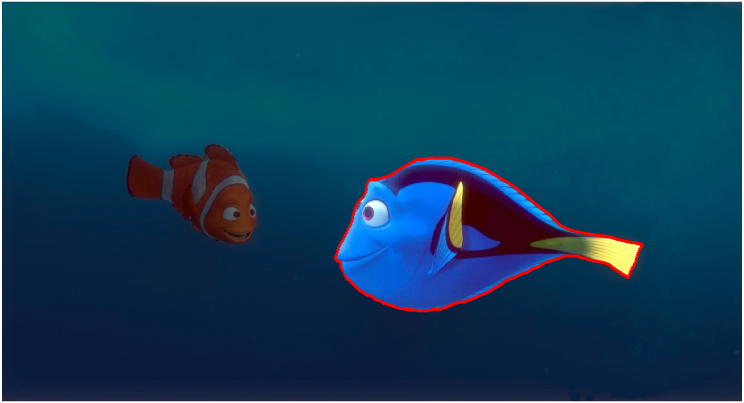
$$\min_u \tilde{G}(\tilde{u}) + \tilde{F}(\tilde{K}\tilde{u})$$

  leads to ”graph projection” methods.

Application of customized PP algorithms to computer vision problems

Single view 3D reconstruction

Let Ω be the image domain, $S \subset \Omega$ an object.



From: Finding Nemo, `https://ohmy.disney.com/movies/2015/12/20/dory-finding-nemo-hero/`

Goal: Estimate a 3D model

First version: Single view 2.5D reconstruction

Oswald, Töppe, Cremers CVPR 2012: Find a height map that has minimal surface for fixed volume and respects the contour.

Mathematically for height map $u : S \to \mathbb{R}$

- $\int_S u(x)\, dx = V$, where V is a user given volume
- Constrain $u_{|\partial S} = 0$
- Minimize $\int_S \sqrt{1 + |\nabla u(x)|^2}\, dx$ (surface area)

Discrete form

$$\min_u \quad \sum_i \sqrt{1 + |(Du)_i|^2} + \delta_{\Sigma_V}(u),$$

for a suitable gradient operator D (respecting $u_{|\partial S} = 0$),

$$\Sigma_V = \{u \in \mathbb{R}^{|S|} \mid \sum_i u_i = V\}.$$

Single view 2.5D reconstruction

How can we minimize

$$E(u) = \sum_i \sqrt{1 + |(Du)_i|^2} + \delta_{\Sigma_V}(u) \ ?$$

One option: Gradient projection.

- Descent on the term that does not have an easy prox:

$$u^{k+1/2} = u^k - \tau D^* v^k, \qquad v_{i,:} = \frac{(Du^k)_{i,:}}{\sqrt{1 + |(Du^k)_{i,:}|^2}}$$

  for suitable τ, with $D : \mathbb{R}^n \to \mathbb{R}^{n \times 2}$.

- Project onto constraint set:

$$\text{proj}_{\Sigma_V}(v) = \underset{u}{\text{argmin}} \frac{1}{2}\|u - v\|_2^2 + \delta_{\Sigma_V}(u)$$

  Board: How does the projection look like?

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Single view 2.5D reconstruction

$$\underset{u}{\operatorname{argmin}} \frac{1}{2}\|u - v\|_2^2 + \delta_{\Sigma_V}(u) = \underset{u}{\operatorname{argmin}} \frac{1}{2}\|u - v\|_2^2 + \delta_{.-V}(\langle \mathbf{1}, u \rangle)$$

Optimality condition

$$0 = \hat{u} - v + \mathbf{1}p, \qquad p \in \partial\delta_{.-V}(\langle \mathbf{1}, \hat{u} \rangle)$$

$$\sum_i \hat{u}_i = V$$

Take inner product of the above equation with $\mathbf{1}$:

$$0 = V - \sum_i v_i + np,$$

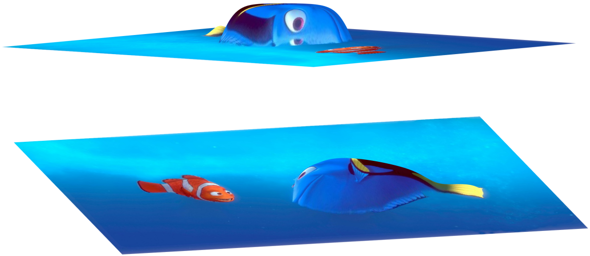
$$\Rightarrow p = \frac{1}{n}\left(V - \sum_i v_i\right),$$

which yields

$$\hat{u} = v - \mathbf{1}\frac{1}{n}\left(V - \sum_i v_i\right) = v - \operatorname{mean}(v)\mathbf{1} + \mathbf{1}\frac{V}{n}$$

Single view 2.5D reconstruction

It works! :-)



Oringinal image from: Finding Nemo,

https://ohmy.disney.com/movies/2015/12/20/dory-finding-nemo-hero/

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Single view 2.5D reconstruction

What about our primal-dual/splitting methods?

$$\min_{u} \quad \sum_{i} \sqrt{1 + |(Du)_i|^2} + \delta_{\Sigma_V}(u),$$

Natural reformulation:

$$\min_{u,d} \quad \sum_{i} \sqrt{1 + |d_i|^2} + \delta_{\Sigma_V}(u), \quad Du = d.$$

But is $F(d) = \sum_i \sqrt{1 + |d_i|^2}$ simple?

- Somewhat yes, as it reduces to a 1D problem.
- Somewhat no, as there is no (easy) closed form solution.

Reformulation that makes the prox operator really easy?

Single view 2.5D reconstruction

Let's start with

$$\min_{u,d} \quad \sum_i \sqrt{1+|d_i|^2} + \delta_{\Sigma_V}(u), \quad Du = d.$$

Note that

$$\sqrt{1+|d_i|^2} = \left|(d_i, 1)^T\right|$$

Idea: Introduce variable e with constraint $e_i = 1$ for all i!

$$\min_{u,d,e} \quad \underbrace{\sum_i \underbrace{\sqrt{e_i^2+|d_i|^2}}_{=|(d_i,e_i)^T|}}_{=\|(d,e)\|_{2,1}} + \delta_{\Sigma_V}(u), \quad Du = d, e = \mathbf{1}$$

Single view 2.5D reconstruction

$$\min_{u,d,e} \quad \|(d,e)\|_{2,1} + \delta_{\Sigma_V}(u), \quad Du = d, e = \mathbf{1}$$

Now the proximity operators of the two functions are simple!

$$\min_{u,d,e} \max_{p,q} \|(d,e)\|_{2,1} + \delta_{\Sigma_V}(u) + \left\langle \begin{pmatrix} p \\ q \end{pmatrix}, \begin{pmatrix} -D & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} u \\ d \\ e \end{pmatrix} \right\rangle - \langle q, \mathbf{1} \rangle$$

Option 1: Use (PDHG) now!

$$\begin{aligned} (\text{dual var})^{k+1} &= \text{prox}_{\sigma F^*}((\text{dual var})^{k} + \sigma K\, \overline{(\text{primal var})}^{k}), \\ (\text{primal var})^{k+1} &= \text{prox}_{\tau G}((\text{primal var})^{k} - \tau K^* (\text{dual var})^{k+1}), \\ \overline{(\text{primal var})}^{k+1} &= (\text{primal var})^{k+1} - (\text{primal var})^{k}. \end{aligned}$$

$\rightarrow$ *Board!*

Single view 2.5D reconstruction

We have

$$K\,\overline{(\text{primal var})}^k = \begin{pmatrix} \bar{d}^k - D\bar{u}^k \\ \bar{e}^k \end{pmatrix}$$

and by identifying the relevant parts of the energy we obtain

$$\begin{aligned} & F^*(p,q) = \langle q, \mathbf{1} \rangle \\ \Rightarrow\ & p^{k+1} = p^k + \sigma(\bar{d}^k - D\bar{u}^k) \\ & q^{k+1} = \operatorname*{argmin}_q \frac{1}{2}\|q - (q^k + \sigma \bar{e}^k)\|^2 + \sigma\langle q, \mathbf{1}\rangle \\ \Rightarrow\ & q^{k+1} = q^k + \sigma(\bar{e}^k - \mathbf{1}) \end{aligned}$$

Single view 2.5D reconstruction

We have

$$K^* \text{ (dual var)}^{k+1} = \begin{pmatrix} -D^* & 0 \\ I & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} p^{k+1} \\ q^{k+1} \end{pmatrix} = \begin{pmatrix} -D^* p^{k+1} \\ p^{k+1} \\ q^{k+1} \end{pmatrix}$$

and by identifying the relevant parts of the energy we obtain

$$G(u, d, e) = \|(d, e)\|_{2,1} + \delta_{\Sigma_V}(u)$$

$$\Rightarrow \quad u^{k+1} = \text{prox}_{\delta_{\Sigma_V}}(u^k + \tau D^* p^{k+1})$$

$$\Rightarrow \quad u^{k+1} = u^k + \tau D^* p^{k+1} + \left(\frac{V}{n} - \text{mean}\left(u^k + \tau D^* p^{k+1} \right) \right) \mathbf{1}$$

$$(d, e)^{k+1} = \text{prox}_{\tau \|\cdot\|_{2,1}}\left((d, e)^k - \tau (p, q)^{k+1} \right)$$

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Single view 2.5D reconstruction

Complete algorithm:

$$
\begin{aligned}
p^{k+1} &= p^k + \sigma(\bar{d}^k - D\bar{u}^k) \\
q^{k+1} &= q^k + \sigma(\bar{e}^k - \mathbf{1}) \\
u^{k+1} &= u^k + \tau D^* p^{k+1} + \left(\frac{V}{n} - \text{mean}\left(u^k + \tau D^* p^{k+1}\right)\right)\mathbf{1} \\
(d,e)^{k+1} &= \text{prox}_{\tau\|\cdot\|_{2,1}}\left((d,e)^k - \tau(p,q)^{k+1}\right) \\
\bar{u}^{k+1} &= 2u^{k+1} - u^k \\
\bar{d}^{k+1} &= 2d^{k+1} - d^k \\
\bar{e}^{k+1} &= 2e^{k+1} - e^k
\end{aligned}
$$

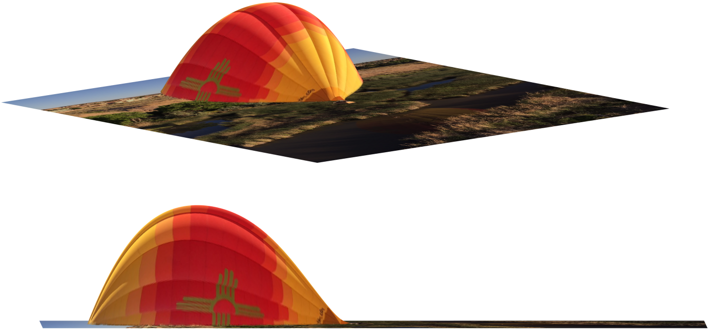
Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Single view 2.5D reconstruction

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Single view 2.5D reconstruction

It still works! :-)



Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Single view 2.5D reconstruction

Let's get back to the original formulation:

$$\min_{u,d,e} \quad \|(d,e)\|_{2,1} + \delta_{\Sigma_V}(u), \quad Du = d, e = \mathbf{1}$$

Now the proximity operators of the two functions are simple!

$$\min_{u,d,e} \max_{p,q} \|(d,e)\|_{2,1} + \delta_{\Sigma_V}(u) + \left\langle \begin{pmatrix} p \\ q \end{pmatrix}, \begin{pmatrix} -D & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} u \\ d \\ e \end{pmatrix} \right\rangle - \langle q, \mathbf{1} \rangle$$

Can we reduce the number of variables?

Single view 2.5D reconstruction

$$\min_{u,d,e}\max_{p,q} \|(d,e)\|_{2,1}+\delta_{\Sigma_V}(u)+\left\langle \begin{pmatrix} p \\ q \end{pmatrix}, \begin{pmatrix} -D & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} u \\ d \\ e \end{pmatrix} \right\rangle - \langle q, \mathbf{1}\rangle$$

Rewritten:

$$\min_{u,d,e}\max_{p,q} \|(d,e)\|_{2,1}+\delta_{\Sigma_V}(u)+\left\langle \begin{pmatrix} p \\ q \end{pmatrix}, \begin{pmatrix} d \\ e \end{pmatrix} \right\rangle - \langle p, Du\rangle - \langle q, \mathbf{1}\rangle$$

Switching min and max (without an explicit proof)

$$\begin{aligned} &\min_{u}\max_{p,q} \left(\min_{d,e} \|(d,e)\|_{2,1} + \left\langle \begin{pmatrix} p \\ q \end{pmatrix}, \begin{pmatrix} d \\ e \end{pmatrix} \right\rangle \right) \\ &\qquad + \delta_{\Sigma_V}(u) - \langle p, Du\rangle - \langle q, \mathbf{1}\rangle \\ =\ &\min_{u}\max_{p,q} -(\|\cdot\|_{2,1})^*((-p,-q)) + \delta_{\Sigma_V}(u) - \langle p, Du\rangle - \langle q, \mathbf{1}\rangle \end{aligned}$$

Single view 2.5D reconstruction

After substituting $p \to -p$, $q \to -q$:

$$\min_{u} \max_{p,q} \; \delta_{\Sigma_V}(u) + \langle p, Du \rangle + \langle q, \mathbf{1} \rangle - (\| \cdot \|_{2,1})^*((p,q))$$

or in explicit primal-dual form

$$\min_{u} \max_{p,q} \; \delta_{\Sigma_V}(u) + \left\langle \begin{pmatrix} p \\ q \end{pmatrix}, \underbrace{\begin{pmatrix} D \\ 0 \end{pmatrix}}_{=:K} u \right\rangle + \langle q, \mathbf{1} \rangle - \delta_{\| \cdot \|_{2,\infty} \leq 1}(p,q)$$

We saved two variables! Let's apply (PDHG)!

$$\begin{aligned}
(\text{dual var})^{k+1} &= \text{prox}_{\sigma F^*}((\text{dual var})^{k} + \sigma K \, \overline{(\text{primal var})}^{k}), \\
(\text{primal var})^{k+1} &= \text{prox}_{\tau G}((\text{primal var})^{k} - \tau K^* \, (\text{dual var})^{k+1}), \\
\overline{(\text{primal var})}^{k+1} &= (\text{primal var})^{k+1} - (\text{primal var})^{k}.
\end{aligned}$$

Single view 2.5D reconstruction

We have

$$K\,\overline{(\text{primal var})}^k = \begin{pmatrix} \bar{u}^k \\ 0 \end{pmatrix}$$

and by identifying the relevant parts of the energy we obtain

$$\begin{aligned} F^*(p,q) =& \delta_{\|\cdot\|_{2,\infty}\leq 1}(p,q) - \langle q, \mathbf{1}\rangle \\ \Rightarrow \ (p,q)^{k+1} =& \operatorname*{argmin}_{(p,q)} \frac{1}{2}\|(p,q) - ((p^k,q^k) + \sigma(\bar{u}^k, 0))\|^2 \\ & + \sigma\delta_{\|\cdot\|_{2,\infty}\leq 1}(p,q) - \sigma\langle q, \mathbf{1}\rangle \\ (p,q)^{k+1} =& \operatorname*{argmin}_{(p,q)} \frac{1}{2}\|(p,q) - ((p^k,q^k) + \sigma(\bar{u}^k, 1))\|^2 \\ & + \delta_{\|\cdot\|_{2,\infty}\leq 1}(p,q) \end{aligned}$$

And similar to before:

$$u^{k+1} = \ u^k - \tau D^* p^{k+1} + \left(\frac{V}{n} - \text{mean}\left(u^k - \tau D^* p^{k+1}\right)\right)\mathbf{1}$$

Single view 2.5D reconstruction

A side note: We could have had the (re-)formulation

$$\min_u \max_{p,q} \; \delta_{\Sigma_V}(u) + \left\langle \begin{pmatrix} p \\ q \end{pmatrix}, \begin{pmatrix} Du \\ \mathbf{1} \end{pmatrix} \right\rangle - \delta_{\|\cdot\|_{2,\infty}\leq 1}(p,q)$$

much faster by seeing it's direct equivalence to

$$\min_u \; \delta_{\Sigma_V}(u) + \|(Du, \mathbf{1})\|_{2,1}\,.$$

This is an interesting general concept that shows the strong relation between (augmented) Lagrangian and (PDHG):

$$\begin{aligned}
\min_u G(u) + F(Ku) &= \min_{u,d,Ku=d} G(u) + F(d) \\
&= \min_{u,d} \max_p G(u) + F(d) + \langle p, Ku - d\rangle \\
&= \min_u \max_p G(u) + \langle p, Ku\rangle + \min_d (F(d) - \langle p, d\rangle) \\
&= \min_u \max_p G(u) + \langle p, Ku\rangle - F^*(p)
\end{aligned}$$

Single view 2.5D reconstruction

How does ADMM work on

$$\min_{u} \; \delta_{\Sigma_V}(u) + \|(Du, \mathbf{1})\|_{2,1} \, .$$

Introduce a new variable

$$\min_{u,d} \; \delta_{\Sigma_V}(u) + \|d\|_{2,1} \quad \text{s.t. } d = (Du, \mathbf{1}).$$

And the augmented Lagrangian

$$L(u, d, p) = \delta_{\Sigma_V}(u) + \|d\|_{2,1} + \langle p, d - (Du, \mathbf{1})\rangle + \frac{\lambda}{2}\|d - (Du, \mathbf{1})\|^2$$

Compute

$$\begin{aligned} u^{k+1} &= \operatorname*{argmin}_{u} L(u, d^k, p^k), \\ d^{k+1} &= \operatorname*{argmin}_{d} L(u^{k+1}, d, p^k), \\ p^{k+1} &= p^k + \lambda(Du^{k+1} - d^{k+1}). \end{aligned}$$

Michael Moeller
Thomas Möllenhoff
Emanuel Laude

Single view 2.5D reconstruction

$$L(u,d,p) = \delta_{\Sigma_V}(u) + \|d\|_{2,1} + \langle p, (Du, \mathbf{1}) - d\rangle + \frac{\lambda}{2}\|d - (Du,\mathbf{1})\|^2$$

ADMM:

$$\begin{aligned} u^{k+1} &= \operatorname*{argmin}_u L(u, d^k, p^k), \\ d^{k+1} &= \operatorname*{argmin}_d L(u^{k+1}, d, p^k), \\ p^{k+1} &= p^k + \lambda(Du^{k+1} - d^{k+1}). \end{aligned}$$

The update in d is easy (a prox we are very familiar with). But what about u?

$$\begin{aligned} u^{k+1} &= \operatorname*{argmin}_u \delta_{\Sigma_V}(u) + \langle p_1^k, Du\rangle + \frac{\lambda}{2}\|d_1^k - Du\|^2 \\ &= \operatorname*{argmin}_u \delta_{\Sigma_V}(u) + \frac{\lambda}{2}\left\|Du - d_1^k + \frac{1}{\lambda}p_1^k\right\|^2 \end{aligned}$$

Single view 2.5D reconstruction

$$u^{k+1} = \operatorname*{argmin}_u \delta_{\Sigma_V}(u) + \frac{\lambda}{2}\left\|Du - d_1^k + \frac{1}{\lambda}p_1^k\right\|^2$$

- Ideal case: $\mathbf{1} \in \ker(D)$ (unfortunately not true here)
- Let $(\mathbf{1}, A)$ be an orthonormal basis of $\mathbb{R}^n$. Then any u can be represented as

$$u = \alpha_0 \mathbf{1} + \sum_{i=1}^{n-1} \alpha_i a_i,$$

via $\alpha_i = \langle a_i, u\rangle$. Thus we may solve

$$\alpha^{k+1} = \operatorname*{argmin}_{\alpha_i,\ i\in\{1,..,n-1\}} \frac{\lambda}{2}\left\|D\left(\alpha_0\mathbf{1} + \sum_{i=1}^{n-1}\alpha_i a_i\right) - d_1^k + \frac{1}{\lambda}p_1^k\right\|^2$$

$$= \operatorname*{argmin}_{\alpha_i,\ i\in\{1,..,n-1\}} \frac{\lambda}{2}\left\|DA\alpha + \alpha_0 D\mathbf{1} - d_1^k + \frac{1}{\lambda}p_1^k\right\|^2$$

subject to $\alpha_0 = V/n$.

Single view 2.5D reconstruction

Single view 2.5D reconstruction

It still works! :-)