CME 307 / MS&E 311: Optimization

Operators

Professor Udell

Management Science and Engineering
Stanford

May 11, 2023

Announcements

-

Outline

Resolvent operator

for relation F, define the **resolvent** of F

$$R_F = (I + F)^{-1}$$

consider resolvent of F

- $(I + F) = \{(x, x + y) : (x, y) \in F\}$
- $R_F = (I + F)^{-1} = \{(x + y, x) : (x, y) \in F\}$
- $R_F = \{(u, v) : (u - v) \in F(v)\}$

Prox is the resolvent of ∂f

- $\mathbf{prox}_f = R_{\partial f} = (I + \partial f)^{-1}$

Prox is the resolvent of ∂f

- $\mathbf{prox}_f = R_{\partial f} = (I + \partial f)^{-1}$
  proof: let $z \in \mathbf{prox}_f(x)$,

$$\begin{aligned} z &= \operatorname*{argmin}_z f(z) + \frac{1}{2}\|z - x\|^2 \\ 0 &\in \partial f(z) + z - x \\ (x - z) &\in \partial f(z) \end{aligned}$$

- $\mathbf{prox}_f = \nabla h^*$ where $h(x) = f(x) + \frac{1}{2}\|x\|^2$

Prox is the resolvent of ∂f

- $\mathbf{prox}_f = R_{\partial f} = (I + \partial f)^{-1}$
  proof: let $z \in \mathbf{prox}_f(x)$,

$$\begin{aligned} z &= \operatorname*{argmin}_z f(z) + \frac{1}{2}\|z - x\|^2 \\ 0 &\in \partial f(z) + z - x \\ (x - z) &\in \partial f(z) \end{aligned}$$

- $\mathbf{prox}_f = \nabla h^*$ where $h(x) = f(x) + \frac{1}{2}\|x\|^2$
  proof: h is CCP and $\partial h = \partial f + I$, so

$$\nabla h^* = (\partial h)^{-1} = (I + \partial f)^{-1}$$

- $\mathbf{prox}_f$ is a function

Prox is the resolvent of ∂f

- $\mathbf{prox}_f = R_{\partial f} = (I + \partial f)^{-1}$
  proof: let $z \in \mathbf{prox}_f(x)$,

$$\begin{aligned} z &= \operatorname*{argmin}_z f(z) + \frac{1}{2}\|z - x\|^2 \\ 0 &\in \partial f(z) + z - x \\ (x - z) &\in \partial f(z) \end{aligned}$$

- $\mathbf{prox}_f = \nabla h^*$ where $h(x) = f(x) + \frac{1}{2}\|x\|^2$
  proof: h is CCP and $\partial h = \partial f + I$, so

$$\nabla h^* = (\partial h)^{-1} = (I + \partial f)^{-1}$$

- $\mathbf{prox}_f$ is a function
  proof: h is strongly convex, so h^* is smooth

Outline

Proximal point method

fixed point iteration using prox is called **proximal point method**

$$x^{(k+1)} = \mathbf{prox}_{tf}(x^{(k)})$$

properties:

- $\mathbf{prox}_{tf}$ is $\frac{1}{2}$ averaged for any $\lambda > 0$, so
- converges for any $\lambda > 0$
- to a zero of ∂f (= FPs of $\mathbf{prox}_{\lambda f}$)
- if f is α-strongly convex, $\mathbf{prox}_{\lambda f}$ is a contraction, so converges linearly
- not usually a practical method (often, as hard as solving original problem)

Method of multipliers

consider

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

let

$$g(\mu) = -(\inf_x f(x) + \mu^T(Ax - b)) = f^*(-A^T\mu) + \mu^T b$$

be the (negative) dual function, and consider the proximal point method for $t > 0$

$$y^{(k+1)} = R_{t\partial g}(y^{(k)})$$

- $\partial g(v) = -A\partial(f^*(-A^T v)) + b$
- $x \in \partial(f^*(-A^T v))$ iff $-A^T v \in \partial f(x)$
- so if $v = R_{t\partial g}(y) = (I + t\partial g)^{-1}(y)$, then

$$\begin{array}{rcl} y & \in & v + t\partial g(v) \\ y & = & v - \alpha(Ax - b) \quad \text{for some } x \text{ with} \quad -A^T v \in \partial f(x) \end{array}$$

Method of multipliers

notice x minimizes the **Augmented Lagrangian** $L_\alpha(x,y)$

$$\begin{aligned} 0 &\in \partial f(x) + A^T(y + \alpha(Ax-b)) \\ x &\in \operatorname*{argmin}_x f(x) + y^T(Ax-b) + \alpha/2\|Ax-b\|^2 = L_\alpha(x,y) \end{aligned}$$

so proximal point method for g is

$$\begin{aligned} x^{(k+1)} &\in \operatorname*{argmin}_x L_\alpha(x, y^{(k)}) \\ y^{(k+1)} &= y^{(k)} + \alpha(Ax^{(k+1)} - b) \end{aligned}$$

also called the **method of multipliers**

properties:

- always converges
- if f is smooth, then g is strongly convex, $R_{t\partial g}$ is a contraction, and the method of multipliers converges linearly
- useful if f is smooth and A is very sparse (alternative: optimize over $x \in x_0 + (A)z$; but (A) is generally dense)

Composition rules

suppose A has Lipschitz constant L_A, B has Lipschitz constant L_B
then $A \circ B$ has Lipschitz constant $\leq L_A L_B$

Composition rules

suppose A has Lipschitz constant L_A, B has Lipschitz constant L_B
then $A \circ B$ has Lipschitz constant $\leq L_A L_B$

proof:

$$\|A \circ By - A \circ Bx\| \leq L_A \|By - Bx\| \leq L_A L_B \|y - x\|$$

- nonexpansive $\circ$ nonexpansive $=$ nonexpansive
- nonexpansive $\circ$ contractive $=$ contractive

Outline

Reductions

suppose f is smooth, g is non-smooth but proxable. solve unconstrained problem

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

or, rewrite as

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & Ax = y \end{array}$$

Reductions

suppose f is smooth, g is non-smooth but proxable. solve unconstrained problem

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

or, rewrite as

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & Ax = y \end{array}$$

how general is this formulation?

Two linear operators

suppose f is smooth, g is non-smooth but proxable. solve

$$\text{minimize} \quad f(Bx) + g(Ax)$$

reformulate

Two linear operators

suppose f is smooth, g is non-smooth but proxable. solve

$$\text{minimize} \quad f(Bx) + g(Ax)$$

reformulate:

$f(Mx)$ is smooth whenever f is, so it's already in the right form

Two linear operators

suppose f is smooth, g is non-smooth but proxable. solve

$$\text{minimize} \quad f(Bx) + g(Ax)$$

reformulate:
$f(Mx)$ is smooth whenever f is, so it's already in the right form

special case: $f(x) = \sum_{i=1}^{m} f_i(x)$

Many fs

suppose f_i is smooth for $i = 1, \ldots, m$, g is non-smooth but proxable. solve

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n f_i(x_i) + g(y) \\ \text{subject to} & \sum_{i=1}^n A_i x_i = y \end{array}$$

reformulate:

Many fs

suppose f_i is smooth for $i = 1, \ldots, m$, g is non-smooth but proxable.
solve

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n f_i(x_i) + g(y) \\ \text{subject to} & \sum_{i=1}^n A_i x_i = y \end{array}$$

reformulate: $x = (x_1, \ldots, x_m)$, $f(x) = \sum_{i=1}^n f_i(x_i)$,
$Ax = \sum_{i=1}^n A_i x_i = y$.

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & Ax = y \end{array}$$

Many gs

suppose f is smooth, g_i is non-smooth but proxable for $i = 1, \ldots, m$.
solve

$$\begin{array}{ll} \text{minimize} & f(x) + \sum_{i=1}^m g_i(y_i) \\ \text{subject to} & A_i x = y_i \end{array}$$

reformulate:

Many gs

suppose f is smooth, g_i is non-smooth but proxable for $i = 1, \ldots, m$.
solve

$$\begin{array}{ll} \text{minimize} & f(x) + \sum_{i=1}^m g_i(y_i) \\ \text{subject to} & A_i x = y_i \end{array}$$

reformulate: $Ax = (A_1 x, \ldots, A_m x) = y$, $g(y) = \sum_{i=1}^m g_i(y_i)$.
g is separable so still proxable.

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & Ax = y \end{array}$$

Conic problem

suppose we have a conic problem over cone K

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \end{array}$$

reformulate:

Conic problem

suppose we have a conic problem over cone K

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \end{array}$$

reformulate:

$$\begin{array}{ll} \text{minimize} & c^T x + I_{\mathcal{K}}(y - b) \\ \text{subject to} & Ax = y \end{array}$$

Conic problem

suppose we have a conic problem over cone K

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \end{array}$$

reformulate:

$$\begin{array}{ll} \text{minimize} & c^T x + I_{\mathcal{K}}(y - b) \\ \text{subject to} & Ax = y \end{array}$$

$\mathbf{prox}_{I_{\mathcal{K}}} = \Pi_{\mathcal{K}}$ is projection onto cone $\mathcal{K}$

Strongly convex

suppose f is strongly convex, g is non-smooth but proxable. solve

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & Ax = y \end{array}$$

reformulate:

Strongly convex

suppose f is strongly convex, g is non-smooth but proxable. solve

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & Ax = y \end{array}$$

reformulate: duality!

$$\begin{aligned} L(x, y, \mu) &= f(x) + g(y) + \mu^T(Ax - y) \\ \inf_{x,y} L(x, y, \mu) &= -f^*(-A^T\mu) - g^*(\mu) \end{aligned}$$

dual formulation:

$$\text{maximize} \quad f^*(-A^T\mu) + g^*(\mu)$$

notice:

- $f^* \circ (-A^T)$ smooth
- if $g = \sum_{i=1}^m g_i(y_i)$ is separable, so is $g^*(\mu) = \sup_y \sum_{i=1}^m (\mu_i y_i - g_i(y_i))$

Outline

Forward backward splitting

suppose F is $\frac{1}{\beta}$-cocoercive and G is maximal monotone
(eg, $F = \nabla f$ and $G = \partial g$)

$$\begin{array}{ll} \text{find} & x \\ \text{subject to} & 0 \in Fx + Gx \end{array}$$

analyze optimality conditions:

$$\begin{aligned} 0 &\in Fx + Gx \\ -tFx &\in tGx \\ (I - tF)x &\in (I + tG)x \\ x &= (I + tGx)^{-1}(I - tF)x \\ x &= R_{tG}(I - tF)x \end{aligned}$$

Forward backward splitting

$$x^+ = R_{tG}(I - tF)x$$

convergence:

- R_{tG} is $\frac{1}{2}$-averaged
- for $t \in (0, \frac{2}{\beta})$, $I - tB$ is averaged
- so FBS converges
- if either F or G is strongly monotone, then FBS converges linearly

Proximal gradient

suppose f is smooth, g is non-smooth but proxable.
then ∇f is $\frac{1}{\beta}$-cocoercive and ∂g is maximal monotone.

FBS for these operators is called **proximal gradient method**

$$x^{+} = \mathbf{prox}_{tg}(x - t\nabla f(x))$$

solves unconstrained problem

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

convergence:

- for $t \in (0, \frac{2}{\beta})$, converges
- if either f or g is strongly convex, then proximal gradient converges linearly

special case: projected gradient

Proximal gradient: interpretation

consider update that linearizes f and regularizes around $x^{(k)}$

$$
\begin{aligned}
x^{(k+1)} &\in \operatorname*{argmin}_x f(x^{(k)}) + \nabla f(x^{(k)})^T(x - x^{(k)}) + \frac{1}{2t}\|x - x^{(k)}\|^2 \\
&\quad + g(x) \\
0 &\in \nabla f(x^{(k)}) + x^{(k+1)} - x^{(k)} + \partial g(x^{(k+1)}) \\
x^{(k)} - \nabla f(x^{(k)}) &\in x^{(k+1)} + \partial g(x^{(k+1)}) \\
x^{(k+1)} &= \mathbf{prox}_{tg}(x^{(k)} - t\nabla f(x^{(k)}))
\end{aligned}
$$

we see proximal gradient update solves

$$\text{minimize} \quad g + \text{quadratic approximation to } f$$

Proximal gradient: interpretation

consider update that linearizes f and regularizes around $x^{(k)}$

$$
\begin{aligned}
x^{(k+1)} &\in \operatorname*{argmin}_x f(x^{(k)}) + \nabla f(x^{(k)})^T(x - x^{(k)}) + \frac{1}{2t}\|x - x^{(k)}\|^2 \\
&\quad + g(x) \\
0 &\in \nabla f(x^{(k)}) + x^{(k+1)} - x^{(k)} + \partial g(x^{(k+1)}) \\
x^{(k)} - \nabla f(x^{(k)}) &\in x^{(k+1)} + \partial g(x^{(k+1)}) \\
x^{(k+1)} &= \mathbf{prox}_{tg}(x^{(k)} - t\nabla f(x^{(k)}))
\end{aligned}
$$

we see proximal gradient update solves

$$\text{minimize} \quad g + \text{quadratic approximation to } f$$

variable metric:

- regularize with $\|x - x^{(k)}\|_L^2$ instead of $\frac{1}{2t}\|x - x^{(k)}\|^2$
- reduces to standard proximal gradient when $L = \frac{1}{t}I$
- converges so long as f is 1-smooth wrt the metric L

Proximal gradient method and composition

suppose f is smooth and g is proxable

- easy to apply proximal gradient method to

$$\text{minimize} \quad f(Ax) + g(x),$$

  since $\nabla(f(Ax)) = A^T(\nabla f)(Ax)$

- hard to apply proximal gradient method to

$$\text{minimize} \quad f(x) + g(Ax),$$

  since

  - $\mathbf{prox}_{g \circ A}$ may not be easy to evaluate even if $\mathbf{prox}_g$ is easy
  - $\mathbf{prox}_{g \circ A}$ may not be separable even if g is separable

Proximal gradient method and composition

suppose f is smooth and g is proxable

- easy to apply proximal gradient method to

$$\text{minimize} \quad f(Ax) + g(x),$$

  since $\nabla(f(Ax)) = A^T(\nabla f)(Ax)$

- hard to apply proximal gradient method to

$$\text{minimize} \quad f(x) + g(Ax),$$

  since
  - $\mathbf{prox}_{g \circ A}$ may not be easy to evaluate even if $\mathbf{prox}_g$ is easy
  - $\mathbf{prox}_{g \circ A}$ may not be separable even if g is separable

what should we do instead?

Can't we just compute A^{-1}?

let $y = Ax$, can't we just use proximal gradient to solve

$$\text{minimize} \quad f(A^{-1}y) + g(y)?$$

why not?

Can't we just compute A^{-1}?

let $y = Ax$, can't we just use proximal gradient to solve

$$\text{minimize} \quad f(A^{-1}y) + g(y)?$$

why not?

- A may not be invertible
- even if A is invertible, inverting it is numerically unstable
- if A is sparse with s nonzeros, applying A and A^T take $O(s)$ flops, while inverting A takes $O(n^3)$ flops

can we use conjugate gradient?

how about using conjugate gradient instead of forming A^{-1}?

to compute

$$y^+ = \mathbf{prox}_{tg}(y - tA^{-T}(\nabla f)(A^{-1}y)),$$

do

- solve $Ax = y$ for x
- solve $A^T g = \nabla f(x)$ for g
- update $y^+ = \mathbf{prox}_{tg}(y - tg)$

can we use conjugate gradient?

how about using conjugate gradient instead of forming A^{-1}?

to compute

$$y^+ = \mathbf{prox}_{tg}(y - tA^{-T}(\nabla f)(A^{-1}y)),$$

do

- solve $Ax = y$ for x
- solve $A^T g = \nabla f(x)$ for g
- update $y^+ = \mathbf{prox}_{tg}(y - tg)$

problem: what if $y^+ \notin \mathrm{range}(A)$?

Dual proximal gradient method

suppose f is strongly convex and g is proxable. instead of

$$\text{minimize} \quad f(x) + g(Ax),$$

consider its dual problem

$$\text{minimize} \quad f^*(-A^T\mu) + g^*(\mu)$$

proximal gradient on the dual is

$$\mu^{(k+1)} = \mathbf{prox}_{tg^*}(I - A\nabla f^*)(-A^T\mu^{(k)})$$

much easier: only need to multiply by A and A^T

Dual proximal gradient method: convergence

sublinear convergence rate if both operators are nonexpansive:

- f is α-strongly convex $\implies$ f^* is $\frac{1}{\alpha}$-smooth $\implies$ $\nabla(f^* \circ -A^T)$ is $\frac{\alpha}{\|A^T\|^2}$ cocoercive $\implies$ $\nabla(f^* \circ -A^T)$ is $\frac{\|A^T\|^2}{\alpha}$ Lipschitz
- g is CCP $\implies$ g^* is CCP $\implies$ $\mathbf{prox}_{g^*}$ is nonexpansive

so get sublinear convergence if $t \in (0, \frac{2\alpha}{\|A^T\|^2})$

linear convergence if in addition either operator is contractive:

- gradient update is contractive f^* strongly convex, which happens if f β-smooth and A is surjective
- prox update is contractive if g^* is strongly convex which happens if g is smooth

Dual proximal gradient method: challenges

two challenges

- how to recover primal solution from dual solution?
- how to compute $\mathbf{prox}_{tg^*}$?

(we've already seen $y \in \nabla f^*(x)$ iff $x \in \partial f(y)$)

Dual proximal gradient method: recover primal

how to recover primal solution from dual solution?

Dual proximal gradient method: recover primal

how to recover primal solution from dual solution?

if $\mu^\star$ is dual optimal for minimize $f(x) + g(Ax)$,
then KKT conditions $\implies$ $x^\star$ primal optimal iff

$$\begin{aligned} x^\star &\in \operatorname*{argmin}_x f(x) + g(y) + (\mu^\star)^T(Ax - y) \\ 0 &\in \partial f(x^\star) + A^T\mu^\star \\ x^\star &\in (\partial f)^{-1}(-A^T\mu^\star) \\ x^\star &\in \nabla f^*(-A^T\mu^\star) \end{aligned}$$

recovers primal solution

Moreau's identity

Moreau's identity:

$$\mathbf{prox}_g + \mathbf{prox}_{g^*} = I$$

Moreau's identity

Moreau's identity:

$$\mathbf{prox}_g + \mathbf{prox}_{g^*} = I$$

proof: let $z = \mathbf{prox}_g(x)$. then

$$\begin{aligned}
\mathbf{prox}_g(x) = (I + \partial f)^{-1}x &= z \\
x &\in (I + \partial f)(z) \\
x - z &\in \partial f(z) \\
\partial f^*(x - z) &\ni z \\
(I + \partial f^*)(x - z) &\ni x - z + z = x \\
x - z &= (I + \partial f^*)^{-1}x = \mathbf{prox}_g^*(x)
\end{aligned}$$

so $\mathbf{prox}_g(x) + \mathbf{prox}_{g^*}(x) = z + x - z = x$

- scale g by t to compute

$$z = \mathbf{prox}_{tg}(z) + \mathbf{prox}_{(tg)^*}(z) = \mathbf{prox}_{tg}(z) + t\mathbf{prox}_{t^{-1}g^*}(t^{-1}z)$$

Dual proximal gradient method: compute $\mathbf{prox}_{tg^*}$

dual proximal gradient method

$$\begin{aligned} x &= \nabla f^*(-A^T\mu) \\ \mu^+ &= \mathbf{prox}_{tg^*}(\mu + tAx) \end{aligned}$$

how to compute $\mathbf{prox}_{tg^*}(\mu + tAx)$?

Dual proximal gradient method: compute $\mathbf{prox}_{tg^*}$

dual proximal gradient method

$$\begin{aligned} x &= \nabla f^*(-A^T\mu) \\ \mu^+ &= \mathbf{prox}_{tg^*}(\mu + tAx) \end{aligned}$$

how to compute $\mathbf{prox}_{tg^*}(\mu + tAx)$?

use Moreau's identity with $tz = \mu + tAx$:

$$\mathbf{prox}_{tg^*}(tz) = tz - \mathbf{prox}_{1/tg}(z)$$

dual proximal gradient method becomes

$$\begin{aligned} x &= \nabla f^*(-A^T\mu) \\ \mu^+ &= \mu + tAx - \mathbf{prox}_{1/tg}(\mu/t + Ax) \end{aligned}$$

Dual proximal gradient method: interpretation

dual proximal gradient method

$$\begin{aligned} x &= \nabla f^*(-A^T\mu) \\ \mu^+ &= \mu + tAx - \mathbf{prox}_{1/tg}(\mu/t + Ax) \end{aligned}$$

- state $\nabla f^*(-A^T\mu)$ explicitly:

  $$\nabla f^*(-A^T\mu) = \operatorname*{argmax}_x (-A^T\mu)^T x - f(x) = \operatorname*{argmin}_x f(x) + \mu^T Ax$$

- state $\mathbf{prox}_{1/tg}(\mu/t + Ax)$ explicitly:

$$\mathbf{prox}_{1/tg}(\mu/t + Ax) = \operatorname*{argmin}_y g(y) + \frac{t}{2}\|y - Ax - \mu/t\|^2$$

dual proximal gradient method becomes

$$\begin{aligned} x &= \operatorname*{argmin}_x f(x) + \mu^T Ax \\ y &= \operatorname*{argmin}_y g(y) + \frac{t}{2}\|y - Ax - \mu/t\|^2 \\ \mu^+ &= \mu + t(Ax - y) \end{aligned}$$

Many more splitting methods

- Peaceman Rachford Splitting
- Douglas Rachford Splitting
- Davis Yin Three Operator Splitting
- Chambolle Pock
- ADMM

details in Ryu and Boyd monograph

Chambolle Pock

consider the problem

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

Chambolle Pock iteration is

$$\begin{aligned} x^{(k+1)} &= R_{t\partial f}(x^{(k)} - tA^T \mu^{(k)}) \\ \mu^{(k+1)} &= R_{t\partial g^*}(\mu^{(k)} + tA(2x^{(k+1)} - x^{(k)})) \end{aligned}$$

- converges when $t < \frac{1}{\|A\|}$
- easy whenever f and g are proxable
- only requires multiplication by A and A^T

Distributed optimization

consider the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n f_i(x_i) + \sum_{j=1}^n g_j(y_j) \\ \text{subject to} & Ax = y \end{array}$$

Chambolle Pock iteration is a distributed optimization method! define

$$\mathcal{N}(i) = \{j : A_{ij} \neq 0\} \quad \mathcal{N}(j) = \{i : A_{ij} \neq 0\}$$

CP iteration is

- for each i, compute

$$x_i^{(k+1)} = \mathbf{prox}_{f_i}(x_i^{(k)} - t \sum_{j \in \mathcal{N}(i)} A_{ij} u_j^{(k)})$$

- for each j, compute

$$\mu_j^{(k+1)} = \mathbf{prox}_{g_j^*}(\mu_j^{(k)} + t \sum_{i \in \mathcal{N}(j)} A_{ij}(2x_i^{(k+1)} - x_i^{(k)}))$$

ADMM

consider the problem

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

Augmented Lagrangian for this problem (with dual variable y) is

$$L_t(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + t/2\|Ax + Bz - c\|^2$$

Alternating Directions Method of Multipliers (ADMM) iteration is

$$\begin{aligned} x^{(k+1)} &= \operatorname*{argmin}_x L_t(x, z^{(k)}, y^{(k)}) \\ z^{(k+1)} &= \operatorname*{argmin}_z L_t(x^{(k+1)}, z, y^{(k)}) \\ y^{(k+1)} &= y^{(k)} + \frac{1}{t}(Ax^{(k+1)} + Bz^{(k+1)} - c) \end{aligned}$$

(special case of Douglas Rachford splitting)

ADMM

properties:

- converges for any $t > 0$ (but can be very slow)
- letting $y = tu$, equivalent to the iteration

$$
\begin{aligned}
x^{(k+1)} &= \operatorname*{argmin}_x f(x) + t/2\|Ax + Bz^{(k)} - c + u^{(k)}\|^2 \\
z^{(k+1)} &= \operatorname*{argmin}_z g(z) + t/2\|Ax^{(k+1)} + Bz - c + u^{(k)}\|^2 \\
u^{(k+1)} &= u^{(k)} + Ax^{(k+1)} + Bz^{(k+1)} - c
\end{aligned}
$$

- frequently used for distributed optimization:
  problems decouple if A or B is diagonal
  (note this is **more restrictive requirement** for distributed opt compared to Chambolle Pock)

Operator splitting for distributed optimization

economy with m agents and n goods.

- agent i has consumption vector $x_i \in \mathbf{R}^n$
- agent i produces $(x_i)_j$ of good j if $(x_i)_j > 0$
- agent i consumes $-(x_i)_j$ of good j if $(x_i)_j < 0$
- agent i has utility function $f_i(x_i)$
- supply = demand if $\sum_i x_i = 0$.

the economy solves the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n f_i(x_i) \\ \text{subject to} & \sum_i x_i = 0 \end{array}$$

References

- Parikh and Boyd, Proximal Algorithms
- Ryu and Boyd, Primer on Monotone Operator Methods
- Davis and Yin, Convergence Rate Analysis of Several Splitting Schemes
- Pontus Gisselson, Course on Large-Scale Convex Optimization http://www.control.lth.se/ls-convex-2015/