Dual decomposition: Special case of Dual Ascent

- $f(\mathbf{x})$ is decomposable into v blocks of variables (such as in Machine Learning, with decomposition over examples)

Dual decomposition: Special case of Dual Ascent

- $f(\mathbf{x})$ is decomposable into v blocks of variables (such as in Machine Learning, with decomposition over examples)

$$\min_{\mathbf{x}} f(\mathbf{x}) = \min_{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_v} \sum_{i=1}^{v} f_i(\mathbf{x}_i)$$

$$\text{s.t. } A\mathbf{x} = b$$

- Let $A = [A_{*1}, A_{*2} \ldots A_{*i} .. A_{*v}]$ be a matrix of v blocks of columns of A corresponding to the blocks $\mathbf{x}_i$.

$$\underbrace{\begin{bmatrix} A_{11} & A_{1i} & A_{1v} \\ A_{21} & A_{2i} & A_{2v} \\ A_{p1} & A_{pi} & A_{pv} \end{bmatrix}}_{p \text{ Linear constraints}} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_i \\ \mathbf{x}_v \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{v} A_{1i}\mathbf{x}_i \\ \sum_{i=1}^{v} A_{2i}\mathbf{x}_i \\ \sum_{i=1}^{v} A_{pi}\mathbf{x}_i \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_p \end{bmatrix}$$

Dual decomposition (contd.)

[argimin over variables of functions of those individual variables, with the functions not mutually interacting is the vector of individual argmins]

- Thus: $f(\mathbf{x}) = \sum_{i=1}^{V} f_i(\mathbf{x}_i)$ and $\sum_{i=1}^{V} A_{*i}\mathbf{x}_i = \mathbf{b}$
- Using this, simplify the first iterative step of dual ascent as
  $\mathbf{x}^{k+1} = \underset{\mathbf{x}}{\text{argmin}}\, f(\mathbf{x}) + {\lambda^k}^\top (A\mathbf{x} - b)$

Dual decomposition (contd.)

- Thus: $f(\mathbf{x}) = \sum_{i=1}^{v} f_i(\mathbf{x}_i)$ and $\sum_{i=1}^{v} A_{*i}\mathbf{x}_i = \mathbf{b}$
- Using this, simplify the first iterative step of dual ascent as
  $\mathbf{x}^{k+1} = \underset{\mathbf{x}}{\text{argmin}}\, f(\mathbf{x}) + {\lambda^{k}}^{\top}(A\mathbf{x} - b)$

$$= \arg \min_{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_v} \sum_{i=1}^{v} f_i(\mathbf{x}_i) + {\lambda^{k}}^{T}\left(\left(\sum_{i=1}^{v} A_i\mathbf{x}_i\right) - \mathbf{b}\right)$$

- Thus, the following **SCATTER** step can be executed parallely for each block indexed by i after broadcasting λ^k from the previous iteration

Dual decomposition (contd.)

- Thus: $f(\mathbf{x}) = \sum_{i=1}^{v} f_i(\mathbf{x}_i)$ and $\sum_{i=1}^{v} A_{*i}\mathbf{x}_i = \mathbf{b}$
- Using this, simplify the first iterative step of dual ascent as
$$\mathbf{x}^{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}}\, f(\mathbf{x}) + \lambda^{k^\top}(A\mathbf{x} - b)$$
$$= \arg \min_{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_v} \sum_{i=1}^{v} f_i(\mathbf{x}_i) + \lambda^{k^T}\left(\left(\sum_{i=1}^{v} A_i\mathbf{x}_i\right) - \mathbf{b}\right)$$
- Thus, the following **SCATTER** step can be executed parallely for each block indexed by i after broadcasting λ^k from the previous iteration
$$\mathbf{x}_i^{k+1} = \underset{\mathbf{x}_i}{\operatorname{argmin}}\, f_i(\mathbf{x}_i) + \lambda^{k^\top}(A_{*i}\mathbf{x}_i)$$
- Subsequently, GATHER the lammbda in the ascent step

Dual decomposition (contd.) = Computational trick

- Thus: $f(\mathbf{x}) = \sum_{i=1}^{v} f_i(\mathbf{x}_i)$ and $\sum_{i=1}^{v} A_{*i}\mathbf{x}_i = \mathbf{b}$
- Using this, simplify the first iterative step of dual ascent as

$$\mathbf{x}^{k+1} = \operatorname{argmin}_{\mathbf{x}} f(\mathbf{x}) + \lambda^{k^\top}(A\mathbf{x} - b)$$

$$= \arg \min_{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_v} \sum_{i=1}^{v} f_i(\mathbf{x}_i) + \lambda^{k^T} \left(\left(\sum_{i=1}^{v} A_i \mathbf{x}_i \right) - \mathbf{b} \right)$$

- Thus, the following **SCATTER** step can be executed parallely for each block indexed by i after broadcasting λ^k from the previous iteration

only this step involves the fi's and might involve (subgradient) computation

parallelizing this step can be more helpful

$$\mathbf{x}_i^{k+1} = \operatorname{argmin}_{\mathbf{x}_i} f_i(\mathbf{x}_i) + \lambda^{k^\top}(A_{*i}\mathbf{x}_i)$$

- Subsequently, **GATHER** $\mathbf{x}_i^{k+1}$ from all nodes and update λ^{k+1} for again broadcasting

$$\lambda^{k+1} = \lambda^k + t^k(A\mathbf{x}^{k+1} - b)$$

Dual decomposition (contd.)

- If we have an inequality constraint instead of an equality, *e.g.* $A\mathbf{x} \leq b$

Hint: Apply projection step along with dual ascent
If Lambda <0, then make it equal to 0

Dual decomposition (contd.)

- If we have an inequality constraint instead of an equality, *e.g.* $A\mathbf{x} \leq b$
  - Just project the computed λ^{k+1} to $\mathbf{R}_+^m$

$$\lambda^{k+1} \leftarrow \left(\lambda^{k+1}\right)_+$$

$$\textit{i.e.} \quad \lambda^{k+1} \leftarrow max\left(0, \lambda^{k+1}\right)$$

Making dual methods more robust: Augmented Lagrangian

- Dual ascent methods are too sensitive to $t^k \leq m$ (m was a lower bound on curvature)
- The idea is to bring in some **strong convexity** by transforming

$$\min_{\mathbf{x} \in \mathbf{R}^n} f(\mathbf{x})$$
$$\text{s.t. } A\mathbf{x} = \mathbf{b}$$

into

Making dual methods more robust: Augmented Lagrangian

- Dual ascent methods are too sensitive to $t^k \leq m$
- The idea is to bring in some **strong convexity** by transforming

$$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x})$$
$$\text{s.t. } A\mathbf{x} = \mathbf{b}$$

into

$$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \frac{\rho}{2}\|A\mathbf{x} - \mathbf{b}\|^2$$
$$\text{s.t. } A\mathbf{x} = \mathbf{b}$$

If A has full column rank, primal objective is strongly convex with constant $\rho\sigma^2_{min}(A)$

- In the initial iteration, $\lambda^{(0)}$ can be arbitrary and $\mathbf{x}^{(1)}$ need not satisfy $A\mathbf{x} = \mathbf{b}$
  Danger: $\mathbf{x}^{k+1}$ may very slowly start satisfying $A\mathbf{x} = \mathbf{b}$
- The transformed objective does not change the final solution, but improves the convergence of dual ascent methods

Augmented Lagrangian: Making dual methods more robust

- One of our main concerns with dual ascent is the sensitivity to $t^k \leq m$
  - If we take the augmented Lagrangian approach, we can use a default value of t^k **using the strong convexity factor that is proportional to ρ** (more motivation on next slide)
- Iterate
  1. $\mathbf{x}^{k+1} = \operatorname{argmin}_{\mathbf{x}} f(\mathbf{x}) + {\lambda^k}^\top (A\mathbf{x} - \mathbf{b}) + \frac{\rho}{2}\|A\mathbf{x} - \mathbf{b}\|^2$
     - ★ The last term here is kind of a barrier function. As we will see, in interior point or barrier methods applied to general inequality constraints, ρ will have to be reduced/changed at each step (but not necessarily here)
  2. $\lambda^{k+1} = \lambda^k + \rho(A\mathbf{x}^{k+1} - \mathbf{b})$
     - ★ Due to ρ (related to strong convexity) instead of t^k , we get better convergence

Augmented Lagrangian: Making dual methods more robust (contd.)

More motivation for replacing t^k with ρ:

- Using ρ instead of t^k, we must have
  $0 \in \partial \left(f(\mathbf{x}^{k+1}) \right) + A^T \left(\lambda^k + \rho(A\mathbf{x}^{k+1} - b) \right)$
- Considering $\widehat{\lambda}^{k+1} = \left(\lambda^k + \rho(A\mathbf{x}^{k+1} - b) \right)$, we get
  $0 \in \partial \left(f(\mathbf{x}^{k+1}) \right) + A^T \widehat{\lambda}^{k+1}$
  which is a necessary condition for our original problem
  - $\widehat{\lambda}^{k+1}$ in place of λ^*

ensures that we are on the KKT (necessary) solution path

Augmented Lagrangian: Making dual methods more robust (contd.)

More motivation for replacing t^k with ρ:

- Using ρ instead of t^k, we must have
  $0 \in \partial\left(f(\mathbf{x}^{k+1})\right) + A^T\left(\lambda^k + \rho(A\mathbf{x}^{k+1} - b)\right)$
- Considering $\widehat{\lambda}^{k+1} = \left(\lambda^k + \rho(A\mathbf{x}^{k+1} - b)\right)$, we get
  $0 \in \partial\left(f(\mathbf{x}^{k+1})\right) + A^T\widehat{\lambda}^{k+1}$
  which is a necessary condition for our original problem
  - $\widehat{\lambda}^{k+1}$ in place of λ^*
- What is the challenge in Applying Dual Decomposition to this Augmented Lagrangian?

||Ax-b|||^2 = (Ax-b)^T(Ax-b)=x^TA^TAx.....

Interactions across blocks of xi's creates non-decomposibility in SCATTER step

ADMM: Best of Several Worlds

- **Extend the decomposition idea to augmented Lagrangian.**
- Iteratively solve a smaller problem with respect to x_i by fixing variables x_j for $j \neq i$.
- Consider simpler case $N = 2$ (easily generalizable to N). $f(x) = f_1(x_1) + f_2(x_2)$ and augmented Lagrangian is

$$L_\rho(x_1, x_2, \lambda) = f_1(x_1) + f_2(x_2) + \lambda^T(A_1x_1 + A_2x_2 - b) + \frac{\rho}{2}\|A_1x_1 + A_2x_2 - \mathbf{b}\|_2^2. \quad (87)$$

ADMM solves each direction alternatively

ADMM takes the idea of dual ascent ahead to alternate between all the x's as well as alternate (like dual ascent, with lambda)

$$x_1^{t+1} = \arg\min_{x_1} L_\rho(x_1, x_2^t, \lambda^t) \quad (88)$$

$$x_2^{t+1} = \arg\min_{x_2} L_\rho(x_1^{t+1}, x_2, \lambda^t) \quad (89)$$

$$\lambda^{t+1} = \lambda^t + \rho(A_1x_1^{t+1} + A_2x_2^{t+1} - \mathbf{b}) \quad (90)$$

- Main difference wrt dual decomposition ascent:

ADMM: Best of Several Worlds

- **Extend the decomposition idea to augmented Lagrangian.**
- Iteratively solve a smaller problem with respect to x_i by fixing variables x_j for $j \neq i$.
- Consider simpler case $N = 2$ (easily generalizable to N). $f(x) = f_1(x_1) + f_2(x_2)$ and augmented Lagrangian is

$$L_\rho(x_1, x_2, \lambda) = f_1(x_1) + f_2(x_2) + \lambda^T(A_1x_1 + A_2x_2 - b) + \frac{\rho}{2}\|A_1x_1 + A_2x_2 - \mathbf{b}\|_2^2. \quad (87)$$

  ADMM solves each direction alternatively

$$x_1^{t+1} = \arg\min_{x_1} L_\rho(x_1, x_2^t, \lambda^t) \quad (88)$$

$$x_2^{t+1} = \arg\min_{x_2} L_\rho(x_1^{t+1}, x_2, \lambda^t) \quad (89)$$

$$\lambda^{t+1} = \lambda^t + \rho(A_1x_1^{t+1} + A_2x_2^{t+1} - \mathbf{b}) \quad (90)$$

- Main difference wrt dual decomposition ascent: ADMM updates x_i sequentially. Additional augmented term does not let us decompose the Lagrangian form into N components conditionally independent wrt λ

ADMM: Alternating Direction Method of Multipliers

1. Assume that functions f_1, f_2 are closed, proper, and convex (that is, they have closed, nonempty, and convex epigraphs)
2. Assume that the un-augmented Lagrangian $L_0(x_1, x_2, \lambda)$ has (critical) saddle points $\widehat{x}_1, \widehat{x}_2$ and $\widehat{\lambda}$ subject to
$$L_0(\widehat{x}_1, \widehat{x}_2, \lambda) \leq L_0(\widehat{x}_1, \widehat{x}_2, \widehat{\lambda}) \leq L_0(x_1, x_2, \widehat{\lambda}) \tag{91}$$
3. No need to assume that A_1, A_2 *etc.* have full column rank

Then when $t \to \infty$, one can prove that[15]

Residual convergence: $r^t = A_1 x_1^t + A_2 x_2^t - \mathbf{b} \to 0$

Objective convergence: $f_1(x_1^t) + f_2(x_2^t) \to f^*$

Dual variable convergence: $\lambda^t \to \lambda^*$

And the rate of convergence is Q-linear[16] (*i.e.*, $(f(\mathbf{x}^k) - p^*) \leq \rho^k (f(\mathbf{x}^0) - p^*)$)

15 https://web.stanford.edu/~boyd/papers/pdf/admm_distr_stats.pdf

16 https://arxiv.org/pdf/1502.02009.pdf

(Log) Barrier methods

Inspired by the Augmented Lagrangian method, how can we use the idea of a barrier to help solve constrained optimization problems while making use of unconstrained optimization techniques

Barrier Methods for Constrained Optimization

Consider a more general constrained optimization problem

Log barrier shoots to infinity even as we tend to violate the constraint. Hence, as iterations proceed and we are consistently in the feasible region, the Barrier function can be gradually ignored

==> 1/t --> 0

by letting t--> infinity as iterations proceed

$$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x})$$
$$\text{s.t.} g_i(\mathbf{x}) \leq 0 \; i = 1 \ldots m$$
$$\text{and } A\mathbf{x} = \mathbf{b}$$

Possibly reformulations of this problem include:

$$\min_x f(x) + \lambda B(x)$$

where B is a **barrier function** like

1. $B(x) = \frac{\rho}{2}\|A\mathbf{x} - \mathbf{b}\|^2$ (in Augmented Langragian - for a specific type of strong convexity wrt $\|.\|^2$))
2. $B(x) = \sum I_{g_i}(\mathbf{x})$ (Projected Gradient Descent: built on this & a linear approximation to $f(\mathbf{x})$)
3. $B(x) = \phi_{g_i}(\mathbf{x}) = -\frac{1}{t}\log\left(-g_i(\mathbf{x})\right)$
   - Here, $-\frac{1}{t}$ is used instead of λ. Lets discuss this in more details

Log barrier is a differentiable convex approximation to (2)

Barrier Method: Example

As a very simple example, consider the following inequality constrained optimization problem.

$$\begin{array}{ll} \text{minimize} & x^2 \\ \text{subject to} & x \geq 1 \end{array}$$

The logarithmic barrier formulation of this problem is

$$\text{minimize} \quad x^2 - \mu \ln(x-1)$$

The unconstrained minimizer for this convex logarithmic barrier function is $\widehat{\mathbf{x}}(\mu) = \frac{1}{2} + \frac{1}{2}\sqrt{1+2\mu}$. As $\mu \to 0$, the optimal point of the logarithmic barrier problem approaches the actual point of optimality $\widehat{\mathbf{x}} = 1$ (which, as we can see, lies on the boundary of the feasible region). The generalized idea, that as $\mu \to 0$, $f(\widehat{\mathbf{x}}) \to p^*$ (where p^* is the optimal for primal) will be proved next.

Homework

Barrier Method and Linear Program

Recap:

Problem type	**Objective Function**	**Constraints**	$L^*(\lambda)$	**Dual constraints**	**Strong duality**
Linear Program	$\mathbf{c}^T\mathbf{x}$	$A\mathbf{x} \leq \mathbf{b}$	$-\mathbf{b}^T\lambda$	$A^T\lambda + \mathbf{c} = \mathbf{0}$	Feasible primal

What are necessary conditions at primal-dual optimality?

- ..
- ..

Complementary Slackness ==> Barrier/Interior methods Force complementar
slackness to hold always while trying to attain feasibility
(eg: Using projection step) at point of optimality

(Primal/Dual) Feasibility==> Barrier/Interior methods Force feasibility
to hold always while trying to attain
complementary slackness at point of optimality

Log Barrier (Interior Point) Method

- The log barrier function is defined as

$$B(x) = \phi_{g_i}(\mathbf{x}) = -\frac{1}{t}\log\left(-g_i(\mathbf{x})\right)$$

- Approximates $\sum I_{g_i}(\mathbf{x})$ (better approximation as $t \to \infty$)
- $f(\mathbf{x}) + \sum_i \phi_{g_i}(\mathbf{x})$ is convex if f and g_i are convex
  Why? $\phi_{g_i}(\mathbf{x})$ is negative of monotonically increasing concave function (log) of a concave function $-g_i(\mathbf{x})$
- Let λ_i be lagrange multiplier associated with inequality constraint $g_i(\mathbf{x}) \leq 0$
- We've taken care of the inequality constraints, lets also consider an equality constraint $A\mathbf{x} = \mathbf{b}$ with corresponding langrage multipler (vector) ν

Log Barrier Method (contd.)

- Our objective becomes

$$\min_x f(x) + \sum_i \left(-\frac{1}{t}\right) \log\left(-g_i(x)\right)$$
$$\text{s.t. } Ax = b$$

- At different values of t, we get different $x^*(t)$
- Let $\lambda_i^*(t) =$
- First-order necessary conditions for optimality (and strong duality)[17] at $x^*(t), \lambda_i^*(t)$:
  1. ..
  2. ..
  3. ..
  4. ..
     - ★ ..
-

[17] of original problem