Optimization in Machine Learning: Lecture 9

Solving Large-scale Problems II

by Xiaolin Huang xiaolinhuang@sjtu.edu.cn SEIEE 2-429

Institute of Image Processing and Pattern Recognition

http://www.pami.sjtu.edu.cn/

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

目录 Contents

目录 Contents

Conditional/reduced gradient

$$\min_{x} \quad f(x) \ \text{s.t.} \boxed{x \in D}$$

- f is convex and contiguously differentiable
- D is a compact convex set
- projected descent method

$$x^{k+1} = \underline{P_D}(x^k - t_k \Delta x_{nt})$$

- "*projection are not always easy*"

https://www.stat.cmu.edu/~ryantibs/convexopt/lectures/frank-wolfe.pdf

Conditional/reduced gradient

$$\min_{x} \quad f(x) \ \text{s.t.}\ x \in D$$

- f is convex and contiguously differentiable
- D is a compact convex set
- Frank-Wolfe method
  - linearize the objective function
  - solve an LP (if D is polynomial)
  - determine the conditional/reduced gradient
  - determine the stepsize

there is no projection

Algorithm 4 Fully-Corrective Variant, Re-Optimizing over all Previous Directions *(with* $\boldsymbol{s}^{(0)} := \boldsymbol{x}^{(0)}$*)*

… *as Algorithm 2, except replacing line* **b)** *with*

b') Update $\boldsymbol{x}^{(k+1)} := \underset{\boldsymbol{x} \in \operatorname{conv}(\boldsymbol{s}^{(0)}, \dots, \boldsymbol{s}^{(k+1)})}{\arg\min} f(\boldsymbol{x})$

Frank-Wolfe method

$$\min_x \quad f(x) \ \text{s.t.}\ x \in D$$

$$z^k = \underset{z}{\operatorname{argmin}} \quad z^\top \nabla f\left(x^k\right) \ \text{s.t.}\ z \in D$$

there is no projection

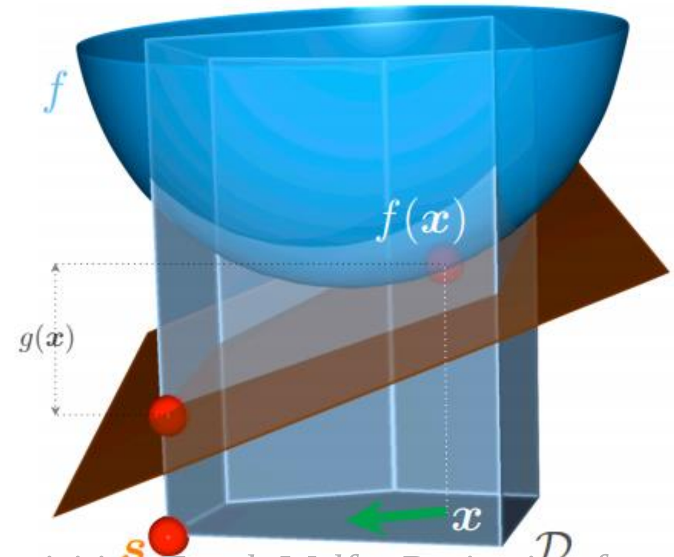
affine invariance

set the direction as $\Delta x^k = z^k - x^k$

set $t^k = \dfrac{2}{k+1}$ or do line search

$$x^{k+1} = x^k - t^k \Delta x^k$$

x^k is kept in the feasible domain



M. Jaggi, Revisiting Frank-Wolfe: Projection-free sparse convex optimization, ICML workshop 2013

Frank-Wolfe method

$$\min_{x} \quad f(x) \;\text{ s.t. } x \in D$$

- for example, when $D = \{x: ||x|| \le t\}$

$$z^k \in \operatorname*{argmin}_{z} \quad z^\top \nabla f(x^k) \;\text{ s.t. } ||z|| \le t$$

$$= -t\left(\operatorname*{arg\,max}_{\|z\|\le 1} \quad z^\top \nabla f(x^k)\right)$$

$$= -t\partial\left\|\nabla f(x^k)\right\|_*$$

  - this could be often simpler/cheaper than projection onto D

$$D = \{x: \|x\|_1 \le t\} \;\Rightarrow\; z^k \in -t\partial\left\|\nabla f(x^k)\right\|_\infty$$

$$D = \{x: \|x\|_p \le t\} \;\Rightarrow\; z_i^k = -t'\operatorname{sign}\left(\nabla f(x^k)\right)\left|\nabla f_i(x^k)\right|^{p/q}$$

Intuitive Explanation

$$z^k = \operatorname*{argmin}_{z} \quad z^\top \nabla f(x^k) \text{ s.t. } z \in D$$

- from the optimality of z^k, we have

$$\nabla f(x^k)^\top (z^k - x^k) \geq 0$$

- from convexity $\quad f(y) \geq f(x^k) + \nabla f(x^k)^\top (y - x^k), \forall x^k, y \in D$

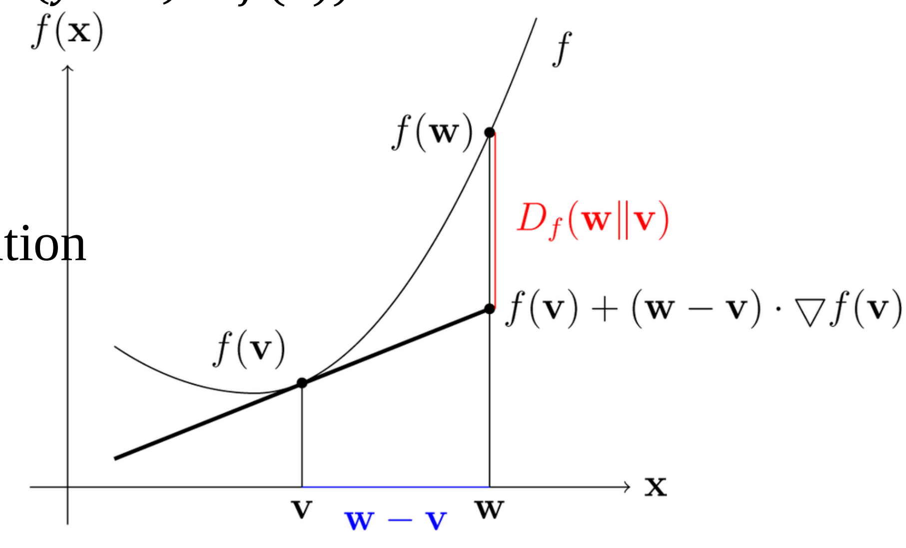
$$\begin{aligned} f(x^*) &\geq f(x^k) + \nabla f(x^k)^\top (x^* - x^k) \\ &\geq \min_{z \in D} \left\{ f(x^k) + \nabla f(x^k)^\top (z - x^k) \right\} \\ &\geq f(x^k) - \nabla f(x^k)^\top x^k + \min_{z \in D} z^\top \nabla f(x^k) \end{aligned}$$

Convergence analysis

- curvature constant to characterize the *non-linearity*

Bregman divergence

$$C_f \triangleq \sup_{\substack{x,s\in D \\ \gamma\in[0,1] \\ y=x+\gamma(s-x)}} \frac{2}{\gamma^2}\left(f(y)-f(x)-(y-x)^\top \nabla f(x)\right)$$



- C_f is closely related to Lipschitz condition

$$C_f \le \mathrm{diam}(D)^2 L$$

- additionally with convexity, we have

$$f(x^{k+1}) = f(x^k + t^k(z^k - x^k))$$

$$\le f(x^k) + t^k(z^k - x^k)^\top \nabla f(x) + \frac{\gamma^2}{2} C_f$$

Convergence analysis

$$f\left(x^{k+1}\right) \leq f\left(x^k\right) + t^k\left(z^k - x^k\right)^\top \nabla f(x) + \frac{\gamma^2}{2} C_f$$

allowing inaccuracy solution

$$\left(z^k - x^k\right)^\top \nabla f(x) \leq \underbrace{\min_{z \in D} z^\top \nabla f\left(x^k\right) - x^k \nabla f\left(x^k\right)}_{\text{duality gap} \quad -g(x^k)} + 2\delta C_f$$

$$f\left(x^{k+1}\right) \leq f\left(x^k\right) - t^k g(x^k) + \frac{\gamma^2 + \delta}{2} C_f$$

$$\min_x f(x) + I_D(x)$$ *indicator function*

$$\max_\alpha -f^*(\alpha) - I_D^*(-\alpha)$$ *supporting function*

$$f(x) + I_D(x) + f^*(\alpha) + I_D^*(-\alpha) \geq x^\top \alpha - I_D^*(-\alpha)$$

at $x = x^k, \alpha = \nabla f(x^k)$

$$x^k \nabla f\left(x^k\right) + \max_{z \in D} -z^\top \nabla f\left(x^k\right)$$

Convergence analysis

$$f(x^{k+1}) \le f(x^k) - t^k g(x^k) + \frac{\gamma^2 + \delta}{2} C_f$$

$$\Downarrow$$

$$f(x^{k+1}) - f^* \le f(x^k) - f^* - t^k g(x^k) + \frac{\gamma^2 + \delta}{2} C_f$$

$$\le f(x^k) - f^* - t^k (f(x^k) - f^*) + \frac{\gamma^2 + \delta}{2} C_f$$

$$\le (1 - t^k)(f(x^k) - f^*) + \frac{\gamma^2 + \delta}{2} C_f$$

- sublinear linear convergence

Theorem: Frank-Wolfe using step sizes $\gamma_k = 2/(k+1)$, $k = 1, 2, 3, \ldots$, and inaccuracy parameter $\delta \ge 0$, satisfies

$$f(x^{(k)}) - f^\star \le \frac{2M}{k+1}(1+\delta)$$

目录 Contents

LASSO

$$\min_x \quad \lambda\|x\|_1 + \|b - Ax\|_2^2$$

- **properties**
  - convex
  - one part is non-smooth and coordinately separable
  - one part is smooth and pointwise separable
- **decomposition into two parts**

$$\min_x \quad \lambda\|x\|_1 + \|b - Az\|_2^2$$

$$\text{s.t.} \quad x = z$$

Properties

- **dual ascent**

Lagrangian: $L(x,z;\alpha) = \lambda\|x\|_1 + \|b - Az\|_2^2 + \underline{\alpha^\top(x-z)}$

dual function: $g(\alpha) = \inf_{x,z} \ L(x,z;\alpha)$

dual problem: $\boxed{\alpha^* = \operatorname{argmax} g(\alpha)}$

primal-dual: $(x^*, z^*) = \operatorname{argmin}_{x,z} L(x,z;\alpha^*)$

- **gradient method for dual ascent**

$$\text{update} \quad \begin{cases} \alpha^{k+1} = \alpha^k + t^k \nabla g(\alpha^k) \\ \nabla g(\alpha^k) = \ \tilde{x} - \tilde{z} \quad \text{where} \quad (\tilde{x}, \tilde{z}) = \operatorname{argmin}_{x,z} \boxed{L(x,z;\alpha^k)} \end{cases}$$

separable

Dual ascent

- maximize the dual problem $g(\lambda, \nu)$, which is always convex

$$g(\lambda, \nu) = \min_x L(x, \lambda, \nu) = \min_x f_0(x) + \lambda^\top f(x) + \nu^\top h(x)$$

$$\leq f_0(x^k) + \lambda^\top f(x^k) + \nu^\top h(x^k)$$

$$= f_0(x^k) + {\lambda^k}^\top f(x^k) + (\lambda - \lambda^k)^\top f(x^k)$$

$$+{\nu^k}^\top h(x^k) + (\nu - \nu^k)^\top h(x^k)$$

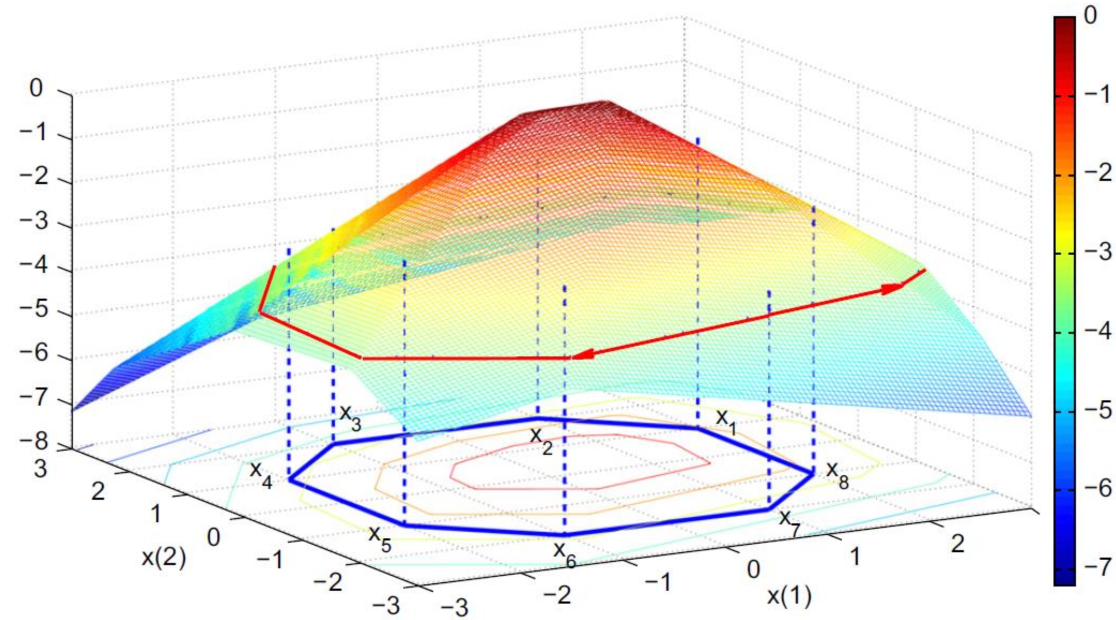
x^k is the minimizer of the Lagrangian at λ^k, ν^k

$$= g(\lambda^k, \nu^k) + (\lambda - \lambda^k)^\top f(x^k) + (\nu - \nu^k)^\top h(x^k)$$

$$\begin{bmatrix} f(x^k) \\ h(x^k) \end{bmatrix} \in \begin{bmatrix} \dfrac{\partial g(\lambda^k, \nu^k)}{\partial \lambda} \\ \dfrac{\partial g(\lambda^k, \nu^k)}{\partial \nu} \end{bmatrix}$$

Dual ascent

- dual ascent method
  - $x^{k+1} = \min_x f_0(x)$ -
  - $\lambda^{k+1} = \max\{\lambda^k + t$
  - $\nu^{k+1} = \nu^k + t^k h(x$



- DC (difference of convex functions) algorithm

$$\min_x \quad g(x) - h(x)$$

both are convex

$$\Downarrow$$

$$\max_y \quad g^*(y) - h^*(y)$$

concave optimization

$$\min_x \quad t - h(x), \text{s.t.}\, g(x) \le t$$

- update: $y^k = \partial h(x^k) = \operatorname{argmin}_y h^*(y) - g^*(y^{k-1}) - {x^k}^\top (y - y^{k-1})$

$$x^{k+1} = \partial g^*(y^k) = \operatorname{argmax}_x x^\top y^k - g(x)$$

Alternating direction method of multipliers

- **Dual decomposition: Lagrangian**

$$L(x,z;\alpha) = \lambda\|x\|_1 + \|b - Az\|_2^2 + \alpha^T(x - z)$$

- **Method of multipliers: augmented Lagrangian**

$$L_\rho(x,z;\alpha) = \lambda\|x\|_1 + \|b - Az\|_2^2 + \alpha^T(x - z) + \frac{\rho}{2}\boxed{\|x - z\|_2^2}$$

strong duality

condition number adjustment

- **ADMM**

$$x^{k+1} = \mathrm{argmin}_x \ L_\rho(x, v^k; \alpha^k)$$

$$z^{k+1} = \mathrm{argmin}_z \ L_\rho(x^{k+1}, z; \alpha^k)$$

$$\alpha^{k+1} = \alpha^k + \boxed{\rho}(x^{k+1} - z^{k+1})$$

ADMM for LASSO

$$\min_{x} \quad \lambda\|x\|_1 + \|b - Az\|_2^2$$

- **augmented Lagrangian**

$$L_\rho(x, z; \alpha) = \lambda\|x\|_1 + \|b - Az\|_2^2 + \alpha^T(x - z) + \frac{\rho}{2}\|x - z\|_2^2$$

- **x-update**

$$x^{k+1} = \operatorname{argmin}_x \; L_\rho\left(x, z^k; \alpha^k\right)$$

$$= \operatorname{argmin}_x \; \lambda\|x\|_1 + x^\top \alpha^k + \frac{\rho}{2}\left\|x - z^k\right\|_2^2$$

$$= S_{\lambda/\rho}\left(z^k + \alpha^k/\rho\right)$$

ADMM for LASSO

$$\min_{x} \quad \lambda\|x\|_1 + \|b - Az\|_2^2$$

- **augmented Lagrangian**

$$L_\rho(x, z; \alpha) = \lambda\|x\|_1 + \|b - Az\|_2^2 + \alpha^T(x - z) + \frac{\rho}{2}\|x - z\|_2^2$$

- **z-update**

$$\begin{aligned} z^{k+1} &= \operatorname{argmin}_z \ L_\rho\left(x^{k+1}, z; \alpha^k\right) \\ &= \operatorname{argmin}_z \|b - Az\|_2^2 + z^\top \alpha^k + \frac{\rho}{2}\left\|x^{k+1} - z\right\|_2^2 \\ &= (A^\top A + \rho I)^{-1}\left(A^\top b + \rho x^{k+1} - \alpha^k\right) \end{aligned}$$

ADMM for LASSO

- **Dual decomposition: Lagrangian**

$$L(x,z;\alpha) = \lambda\|x\|_1 + \|b - Az\|_2^2 + \alpha^T(x - z)$$

- **Method of multipliers: augmented Lagrangian**

$$L_\rho(x,z;\alpha) = \lambda\|x\|_1 + \|b - Az\|_2^2 + \alpha^T(x - z) + \frac{\rho}{2}\|x - z\|_2^2$$

- **ADMM**

$$x^{k+1} = \mathrm{argmin}_x\ L_\rho\left(x, v^k; \alpha^k\right)$$

analytical solution

$$z^{k+1} = \mathrm{argmin}_z\ L_\rho\left(x^{k+1}, z; \alpha^k\right)$$

analytical solution, inverse only once

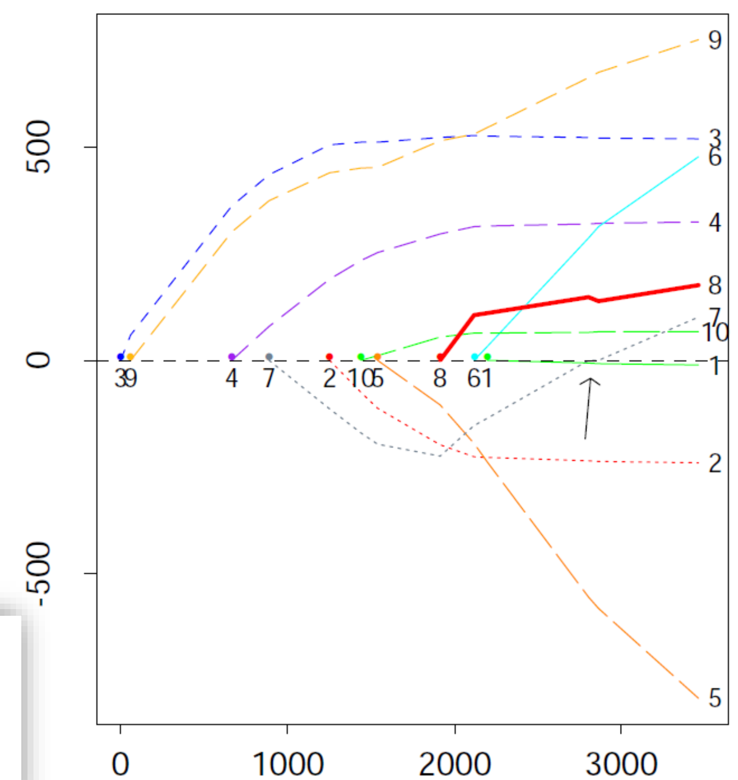
$$\alpha^{k+1} = \alpha^k + \rho\left(x^{k+1} - z^{k+1}\right)$$

LARS: Least Angle Regression

$$\min_x \quad \lambda \|x\|_1 + \|b - Ax\|_2^2$$

- **another interesting idea**
  - the optimal solution is a function of λ, denoted by $x(\lambda)$
  - $x(\lambda)$ is piecewise linear and thus, trackable
  - $x(\infty)$ is known

Lasso



Least Angle Regression

Bradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani
Statistics Department, Stanford University

Alternating direction method of multipliers

$$\begin{aligned} \min \quad & f(x) + g(z) \\ \text{s.t.} \quad & Ax + Bz = c \end{aligned}$$

- **Method of multipliers: augmented Lagrangian**

$$L_\rho(x, z; \alpha) = \lambda f(x) + g(z) + \alpha^T (Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2$$

- **ADMM**

$$x^{k+1} = \mathrm{argmin}_x \; L_\rho\left(x, z^k; \alpha^k\right)$$

$$z^{k+1} = \mathrm{argmin}_z \; L_\rho\left(x^{k+1}, z; \alpha^k\right)$$

$$\alpha^{k+1} = \alpha^k + \rho\left(Ax^{k+1} + Bz^{k+1} - c\right)$$

z^{k+1} minimize $L(x^{k+1}, z, y^k)$

$$\begin{aligned} \Rightarrow 0 &= \nabla_z L_\rho(x^{k+1}, z^{k+1}, y^k) \\ &= \nabla g(z^{k+1}) + B^T y^k + \rho B^T(Ax^{k+1} + Bz^{k+1} - c) \\ &= \nabla g(z^{k+1}) + B^T(y^k + \underbrace{\rho(Ax^{k+1} + Bz^{k+1} - c)}) \end{aligned}$$

$$(x^{k+1}, z^{k+1}, y^{k+1}) \quad = \nabla g(z^{k+1}) + B^T y^{k+1}$$

Satisfies 2nd dual feasibility.

Convergence on ADMM

$$\begin{array}{ll} \min & f(x) + g(z) \\ \text{s.t.} & Ax + Bz = c \end{array}$$

- **assumption**
  - f and g are convex, closed, proper
  - the Lagrangian has a saddle point
- **convergence**
  - approach feasibility
  - approach optimal solution

More discussions

- in-exact iteration

$$z^{k+1} = (A^\top A + \rho I)^{-1}\left(A^\top b + \rho x^{k+1} - \alpha^k\right)$$

  - e.g., solve by SGD

- update order

- multiple-block

$$\min_{x_i} \quad \Sigma_i h_i(x_i) \quad \text{s.t.} \;\; \Sigma_i A_i x_i = c$$

  - convergence discussion becomes hard
  - difficult to find hyper-parameters

Related algorithms

- operator splitting methods (Doulgas, Bregman, Lions and Mercier, …)
- proximal methods (Rochafellar, ...)

$$\mathbf{prox}_{\lambda f}(u) = \operatorname{argmin}_x (f(x) + 1/2\lambda \|x - u\|_2^2)$$

$$x^{k+1} = \mathbf{prox}_{\lambda f}\left(x^k - \lambda^k \nabla f(x^k)\right)$$

proximal gradient method

- Dykstra's alternating projections algorithm
- Singarn's method of partial inverses
- Rockafellar-Wets progressive hedging for scenario-based decomposition

https://stanford.edu/~boyd/papers/pdf/admm_slides.pdf

目录 Contents

Parallel computing by ADMM

$$\min_{x} \quad \Omega(x) + \Sigma_i l(a_i^\top x)$$

$$\Downarrow$$

$$\min_{x} \quad \Omega(x_0) + \Sigma_i f_i(x_i) \quad \text{s.t.} \quad x_i = x_0, \forall i$$

- augmented Lagrangian:

$$L_\rho(x_0, x_i; \alpha_i) = \Omega(x_0) + \Sigma_i f_i(x_i) + \Sigma_i \alpha_i^\top (x_i - x_0) + \frac{\rho}{2} \Sigma_i \|x_i - x_0\|_2^2$$

- update:

$$x_0^{k+1} = \operatorname{argmin}_{x_0} \; L_\rho\left(x_0, x_i^k; \alpha_i^k\right)$$

$$x_i^{k+1} = \operatorname{argmin}_{x_i} \; L_\rho\left(x_0^{k+1}, x_i; \alpha_i^k\right)$$

$$\alpha_i^{k+1} = \alpha_i^k + \rho\left(x_i^{k+1} - x_0^{k+1}\right)$$

Parallel computing by ADMM

$$\min_x \quad \Omega(x) + \Sigma_i l(a_i^\top x)$$

$$\Downarrow$$

$$\min_x \quad \Omega(x_0) + \Sigma_i f_i(x_i) \quad \text{s.t.} \quad x_i = x_0, \forall i$$

- augmented Lagrangian:

$$L_\rho(x_0, x_i; \alpha_i) = \Omega(x_0) + \Sigma_i f_i(x_i) + \Sigma_i \alpha_i^\top (x_i - x_0) + \frac{\rho}{2} \Sigma_i \|x_i - x_0\|_2^2$$

- update:

$$x_0^{k+1} = \operatorname{argmin}_{x_0} \ L_\rho\left(x_0, x_i^k; \alpha_i^k\right)$$

the master node takes charge of this part

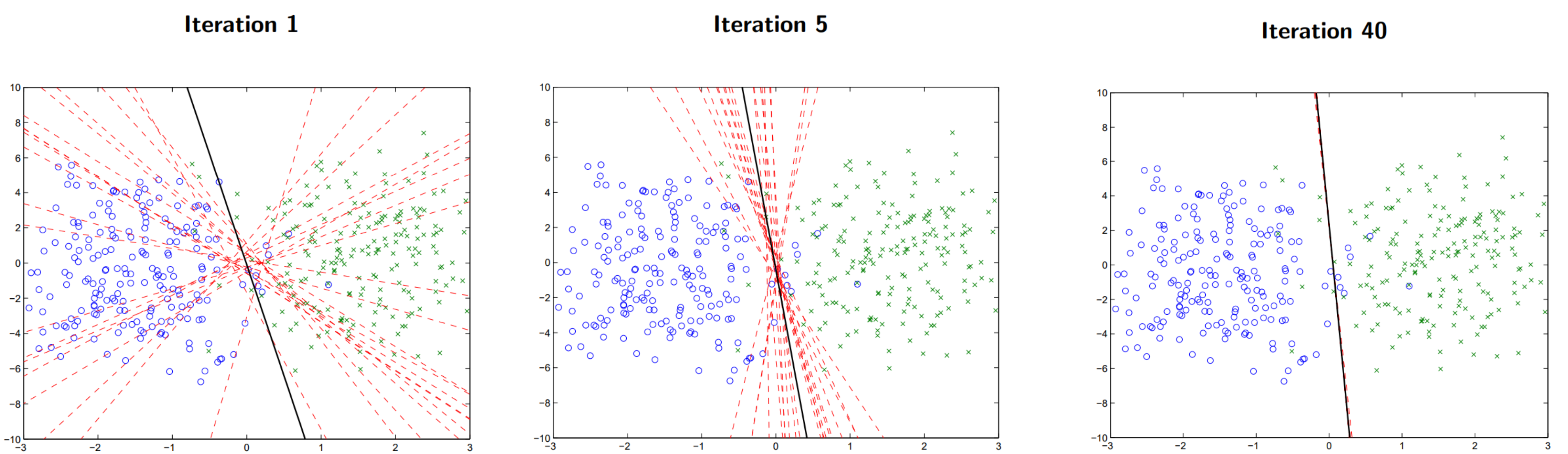
$$\left.\begin{aligned} x_i^{k+1} &= \operatorname{argmin}_{x_i} \ L_\rho\left(x_0^{k+1}, x_i; \alpha_i^k\right) \\ \alpha_i^{k+1} &= \alpha_i^k + \rho\left(x_i^{k+1} - x_0^{k+1}\right) \end{aligned}\right\}$$

each worker takes charge of this part

Parallel computing for SVM

$$\min_{x} \quad \lambda x^\top x + \Sigma_i \max\{0, b_i(a_i^\top x)\}$$

- toy example https://stanford.edu/~boyd/papers/pdf/admm_slides.pdf
  - 2 dimension, 400 samples
  - split into 20 groups (worst: each group contains examples in only on class)



Parallel computing by ADMM

$$\min_{x} \quad \Omega(x) + \Sigma_i l(a_i^\top x)$$

$$\Downarrow$$

$$\min_{x} \quad \Omega(x_0) + \Sigma_i f_i(x_i) \quad \text{s.t.} \quad x_i = x_0, \forall i$$

- augmented Lagrangian:

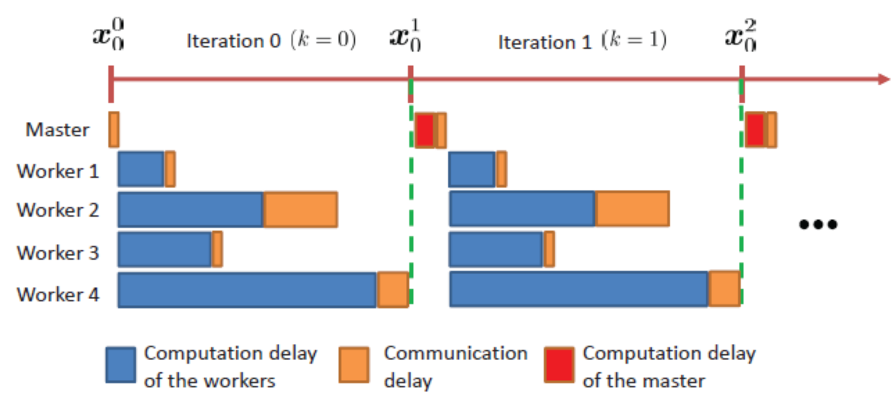
$$L_\rho(x_0, x_i; \alpha_i) = \Omega(x_0) + \Sigma_i f_i(x_i) + \Sigma_i \alpha_i^\top (x_i - x_0) + \frac{\rho}{2} \Sigma_i \|x_i - x_0\|_2^2$$

- update:

$$x_0^{k+1} = \text{argmin}_{x_0} \ L_\rho\left(x_0, x_i^k; \alpha_i^k\right)$$

$$x_i^{k+1} = \text{argmin}_{x_i} \ L_\rho\left(x_0^{k+1}, x_i; \alpha_i^k\right.$$

$$\alpha_i^{k+1} = \alpha_i^k + \rho\left(x_i^{k+1} - x_0^{k+1}\right)$$



Parallel computing by ADMM

$$\min_{x} \quad \Omega(x) + \Sigma_i l(a_i^\top x)$$

$$\min_{x} \quad \Omega(x_0) + \Sigma_i f_i(x_i) \quad \text{s.t.} \quad x_i = x_0, \forall i$$

- augmented Lagrangian:

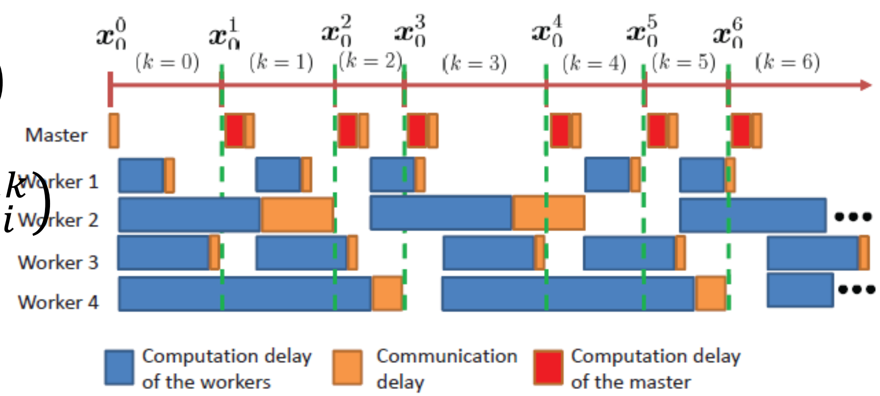
$$L_\rho(x_0, x_i; \alpha_i) = \Omega(x_0) + \Sigma_i f_i(x_i) + \Sigma_i \alpha_i^\top (x_i - x_0) + \frac{\rho}{2} \Sigma_i \|x_i - x_0\|_2^2$$

- update:

$$x_0^{k+1} = \text{argmin}_{x_0} \; L_\rho\left(x_0, x_i^k; \alpha_i^k\right)$$

$$x_i^{k+1} = \text{argmin}_{x_i} \; L_\rho\left(x_0^{k+1}, x_i; \alpha_i^k\right)$$

$$\alpha_i^{k+1} = \alpha_i^k + \rho\left(x_i^{k+1} - x_0^{k+1}\right)$$



Parallel computing by ADMM

$$\min_{x} \quad \Omega(x) + \Sigma_i l(a_i^\top x)$$

$$\Downarrow$$

$$\min_{x} \quad \Omega(x_0) + \Sigma_i f_i(x_i) \quad \text{s.t.} \quad x_i = x_0, \forall i$$

- augmented Lagrangian:

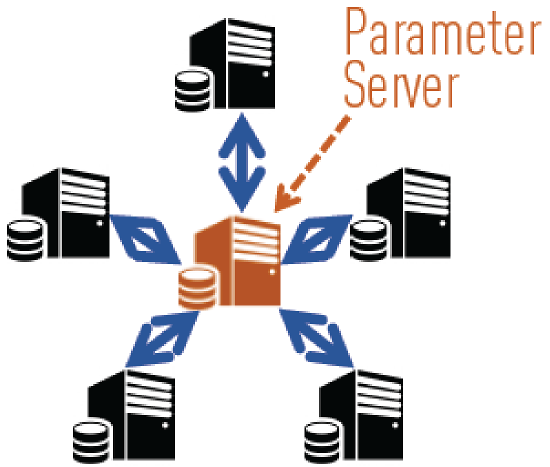
$$L_\rho(x_0, x_i; \alpha_i) = \Omega(x_0) + \Sigma_i f_i(x_i) + \Sigma_i \alpha_i^\top (x_i - x_0) + \frac{\rho}{2} \Sigma_i \|x_i - x_0\|_2^2$$

- update:

$$x_0^{k+1} = \text{argmin}_{x_0} \ L_\rho\left(x_0, x_i^k; \alpha_i^k\right)$$

$$x_i^{k+1} = \text{argmin}_{x_i} \ L_\rho\left(x_0^{k+1}, x_i; \alpha_i^k\right)$$

$$\alpha_i^{k+1} = \alpha_i^k + \rho\left(x_i^{k+1} - x_0^{k+1}\right)$$



Parallel computing by ADMM

$$\min_x \quad \Omega(x) + \Sigma_i l(a_i^\top x)$$

$$\min_x \quad \Omega(x_0) + \Sigma_i f_i(x_i) \quad \text{s.t.} \quad x_i = x_0, \forall i$$

- augmented Lagrangian:

$$L_\rho(x_0, x_i; \alpha_i) = \Omega(x_0) + \Sigma_i f_i(x_i) + \Sigma_i \alpha_i^\top (x_i - x_0) + \frac{\rho}{2} \Sigma_i \|x_i - x_0\|_2^2$$

- update:

$$x_0^{k+1} = \text{argmin}_{x_0} \ L_\rho\left(x_0, x_i^k; \alpha_i^k\right)$$

$$x_i^{k+1} = \text{argmin}_{x_i} \ L_\rho\left(x_0^{k+1}, x_i; \alpha_i^k\right)$$

$$\alpha_i^{k+1} = \alpha_i^k + \rho\left(x_i^{k+1} - x_0^{k+1}\right)$$

Decentralized strategy

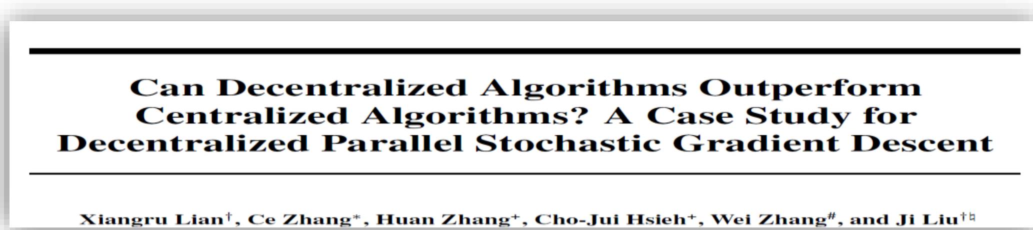
$$\min_{x} \quad \Omega(x) + \Sigma_i l(a_i^\top x)$$

- Decentralized SGD
  - compute local stochastic gradient
  - compute the neighborhood weighted average

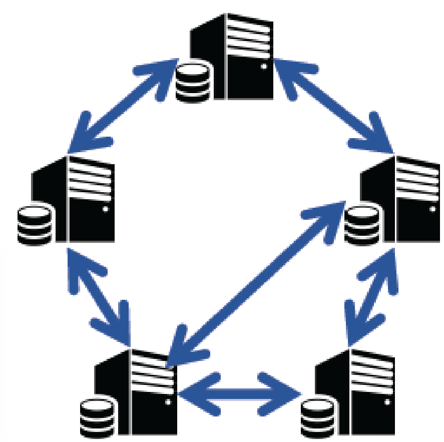
$$x_i^{k+\frac{1}{2}} = \Sigma_{i \in \mathrm{Nei}} a_i x_i^k$$

  - update local variable

$$x_i^{k+1} = x_i^{k+\frac{1}{2}} - t \nabla f_i(x_i)$$


Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent

Xiangru Lian[†], Ce Zhang[*], Huan Zhang[+], Cho-Jui Hsieh[+], Wei Zhang[#], and Ji Liu[†♮]



Decentralized strategy

$$\min_{x} \quad \Omega(x) + \Sigma_i l(a_i^\top x)$$

- Decentralized SGD
  - compute local stochastic gradient
  - compute the neighborhood weighted average

$$x_i^{k+\frac{1}{2}} = \Sigma_{i\in \mathrm{Nei}} a_i x_i^k$$

could be asynchronous

  - update local variable

$$x_i^{k+1} = x_i^{k+\frac{1}{2}} - t\nabla f_i(x_i)$$


Asynchronous Decentralized Parallel Stochastic Gradient Descent

Xiangru Lian [1*] Wei Zhang [2*] Ce Zhang [3] Ji Liu [4 1]

Decentralized strategy

$$\min_{x} \quad \Omega(x) + \Sigma_i l(a_i^\top x)$$

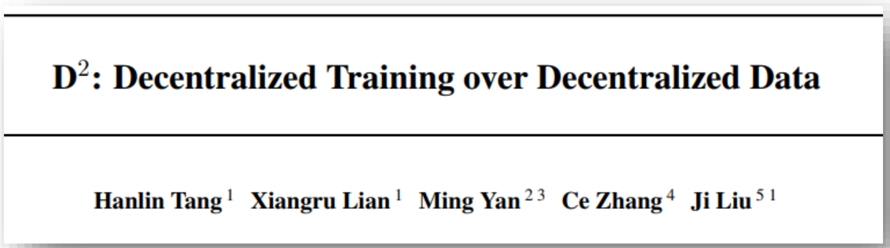
- Decentralized SGD
  - compute local stochastic gradient
  - compute the neighborhood weighted average

$$x_i^{k+\frac{1}{2}} = \Sigma_{i\in \mathrm{Nei}} a_i x_i^k$$

  - update local variable

$$x_i^{k+1} = x_i^{k+\frac{1}{2}} - t\nabla f_i(x_i)$$

when the data are really decentralized (there are variance)



D^2: Decentralized Training over Decentralized Data

Hanlin Tang[1] Xiangru Lian[1] Ming Yan[2,3] Ce Zhang[4] Ji Liu[5,1]

THANKS