SNU Cryptography Seminar: Convex Optimization & Distributed Learning via Alternating Direction Method of Multipliers

Sunghee Yun

VP, AI Technology & Product Strategy
Erudio Bio, Inc.

About the Speaker

- VP, AI Technology & Product Strategy @ Erudio Bio, Inc.
- Advisory Professor, Electrical Engineering and Computer Science @ DGIST
- Adjunct Professor, Electronic Engineering Department @ Sogang University
- Technology Consultant @ Gerson Lehrman Gruop (GLG), NYC

- CTO & Chief Applied Scientist - Senior Fellow @ Gauss Labs Inc. ~ 2023
- Senior Applied Scientist @ Amazon.com, Inc. ~ 2020
- Principal Engineer @ Software R&D Center of Samsung DS Division ~ 2017
- Principal Engineer @ Strategic Marketing Team of Memory Business Unit ~ 2016
- Principal Engineer @ Memory Design Technology Team of DRAM Development Lab. ~ 2015
- Senior Engineer @ CAE Team of Samsung Semiconductor ~ 2012
- M.S. & Ph.D. - Electrical Engineering (EE) @ Stanford University ~ 2004
- B.S. - Electrical Engineering (EE) @ Seoul National University ~ 1998

Today

- convex optimization (cvx opt) & machine learning
  - cvx opt definition
  - dual problem w/ examples & weak/strong dualities
  - KKT & complementary slackness
- distributed learning via alternating direction method of multipliers (ADMM)
  - dual decomposition & method of multipliers
  - ADMM definition & convergence
  - examples: constrained opt, consensus opt, consensus SVM, distributed lasso
- conclusions

Convex optimization

- many (supervised) machine learning (ML) depend on convex optimization (inherently)
- one of few optimization class that can be *actually solved*
- many engineering and scientific problems can be cast into convex optimization problems
- many more can be approximated by convex optimization
- convex optimization sheds lights on intrinsic property and structure of ML algorithms

Mathematical optimization

- mathematical optimization problem:

$$
\begin{array}{ll}
\text{minimize} & f_0(x) \\
\text{subject to} & f_i(x) \leq 0, \ i = 1, \ldots, m \\
& h_i(x) = 0, \ i = 1, \ldots, p
\end{array}
$$

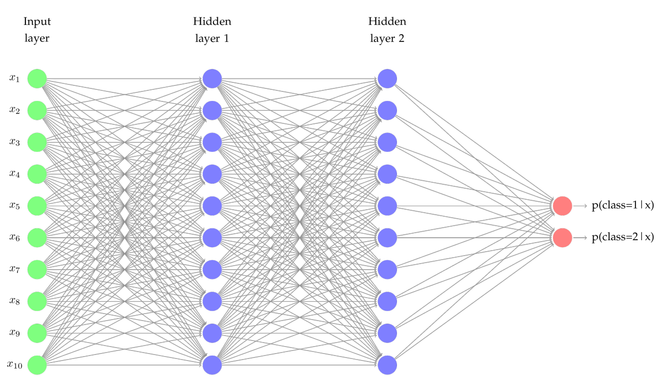
  - $x = \left[\begin{array}{ccc} x_1 & \cdots & x_n \end{array} \right]^T \in \mathbf{R}^n$ is the (vector) optimization variable
  - $f_0 : \mathbf{R}^n \to \mathbf{R}$ is the objective function
  - $f_i : \mathbf{R}^n \to \mathbf{R}$ are the inequality constraint functions
  - $h_i : \mathbf{R}^n \to \mathbf{R}$ are the equality constraint functions

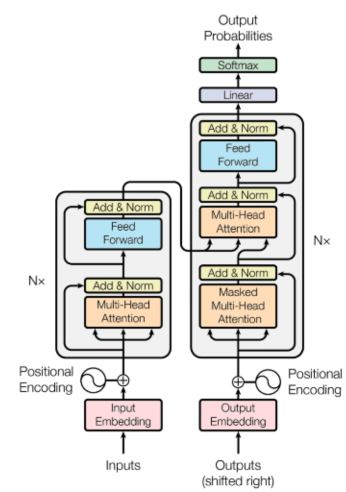
Optimization examples

- circuit optimization
  - optimization variables: transistor widths, resistances, capacitances, inductances
  - objective: operating speed (or equivalently, maximum delay)
  - constraints: area, power consumption
- portfolio optimization
  - optimization variables: amounts invested in different assets
  - objective: expected return
  - constraints: budget, overall risk, return variance

Optimization example - deep neural network (DNN)

- machine learning
  - optimization variables: model parameters, *e.g.*, connection weights, activation functions, number of layers
  - objective: loss function, *e.g.*, sum of squares of errors
  - constraints: network architecture, *e.g.*, fully-connected, transformer





Convex optimization

- canonical form:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \preceq_{K_i} 0, \;\; i = 1, \ldots, m \\ & Ax = b \end{array}$$

  where

  - $f_0(\lambda x + (1-\lambda) y) \leq \lambda f_0(x) + (1-\lambda) f_0(y)$ for all $x, y \in \mathbf{R}^n$ and $0 \leq \lambda \leq 1$
  - $f_i : \mathbf{R}^n \to \mathbf{R}^{k_i}$ are K_i-convex w.r.t. proper cone $K_i \subseteq \mathbf{R}^{k_i}$
  - all equality constraints are linear

Convex optimization

- algorithms
  - classical algorithms like simplex method still work well for many LPs
  - many state-of-the-art algorithms develoled for (even) large-scale convex optimization problems
    * barrier methods
    * primal-dual interior-point methods
- applications
  - many engineering and scientific problems are (or can be cast into) convex optimization problems
  - statistical parameter estimation, ML, signal processing, (variational) Bayesian inference, bioinformatics, chemical engineering, mechanical engineering

Why is convex optimization impactful?

- which one of these problems are easier to solve?
  - (generalized) geometric program with $n = 1,000,000$ variables and $m = 1,000$ constraints

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{p_0} \alpha_{0,i} x_1^{\beta_{0,i,1}} \cdots x_n^{\beta_{0,i,n}} \\ \text{subject to} & \sum_{i=1}^{p_j} \alpha_{j,i} x_1^{\beta_{j,i,1}} \cdots x_n^{\beta_{j,i,n}} \leq 1, \ j = 1, \ldots, m \end{array}$$

    with $\alpha_{j,i} \geq 0$ and $\beta_{j,i,k} \in \mathbf{R}$
    $\Rightarrow$ can be solved *globally* in your laptop computer
  - minimization of 10th order polynomial of $n = 20$ variables with no constraint

$$\text{minimize} \quad \sum_{i_1=1}^{10} \cdots \sum_{i_n=1}^{10} c_{i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n}$$

    with $c_{i_1,\ldots,i_n} \in \mathbf{R}$
    $\Rightarrow$ you *cannot* solve!

Convex optimization example - SVM*

- problem definition:
  - given $x^{(i)} \in \mathbf{R}^p$: input data, and $y^{(i)} \in \{-1, 1\}$: output labels
  - find hyperplane which separates two different classes as distinctively as possible (in some measure)
- (typical) formulation:

$$\begin{array}{ll} \mbox{minimize} & \|a\|_2^2 + \gamma \sum_{i=1}^m u_i \\ \mbox{subject to} & y^{(i)}(a^T x^{(i)} + b) \geq 1 - u_i, \ i = 1, \ldots, m \\ & u \succeq 0 \end{array}$$

  - convex optimization problem, hence stable and efficient algorithms exist even for very large problems
  - has worked extremely well in practice

Duality

- every (constrained) optimization problem has a *dual problem* (whether or not it's a convex optimization problem)
- every dual problem is a *convex optimization problem* (whether or not it's a convex optimization problem)
- duality provides *optimality certificate*, hence plays *central role* for modern optimization and machine learning algorithm implementation
- duality produces beautiful interpretations, $e.g.$,
  - entropy maximization is dual of (transformed) geometric program
  - exchange problem is dual of consensus problem
  - quadratic program is dual of support vector machine (SVM)
- (usually) solving one readily solves the other!

Lagrangian*

- standard form problem:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \ i = 1, \ldots, m \\ & h_i(x) = 0, \ i = 1, \ldots, p \end{array}$$

  where $x \in \mathbf{R}^n$ is optimization variable, $\mathcal{D}$ is domain, p^* is optimal value

- Lagrangian: $L : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ with $\mathbf{dom}\, L = \mathcal{D} \times \mathbf{R}^m \times \mathbf{R}^p$ defined by

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

  - λ_i: Lagrange multiplier associated with $f_i(x) \leq 0$
  - ν_i: Lagrange multiplier associated with $h_i(x) = 0$

Lagrange dual function*

- Lagrange dual function: $g : \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ defined by

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \nu_i h_i(x) \right)$$

  - g is *always* concave
  - $g(\lambda, \nu)$ can be $-\infty$
- lower bound property: if $\lambda \succeq 0$, then $g(\lambda, \nu) \leq p^*$

  Proof: If $\tilde{x}$ is feasible and $\lambda \succeq 0$, then $f_0(\tilde{x}) \geq L(\tilde{x}, \lambda, \nu) \geq \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = g(\lambda, \nu)$. Thus,

$$p^* = \inf_{x \in \mathcal{F}} f_0(x) \geq g(\lambda, \nu)$$

  where $\mathcal{F} = \{x \mid f_i(x) \leq 0 \text{ for } 1 \leq i \leq m,\ h_j(x) = 0 \text{ for } 1 \leq j \leq p\}$.

Dual problem

- Lagrange dual problem:

$$\begin{array}{ll} \text{maximize} & g(\lambda, \nu) \\ \text{subject to} & \lambda \succeq 0 \end{array}$$

  - very good lower bound on p^* (obtained from Lagrange dual function)
  - is a convex optimization problem
  - optimal value denoted by d^*
  - λ, ν are calle *dual feasible* if $\lambda \succeq 0$
- example: standard form LP and its dual

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \succeq 0 \end{array} \qquad \begin{array}{ll} \text{maximize} & -b^T \nu \\ \text{subject to} & A^T \nu + c \succeq 0 \end{array}$$

Weak duality

- weak duality implies $d^* \leq p^*$
  - always true (by construction of dual problem)
  - provides *nontrivial* lower bounds, especially, for difficult problems, $e.g.$, solving the following SDP:

$$\begin{array}{ll} \mbox{maximize} & -\mathbf{1}^T \nu \\ \mbox{subject to} & W + \mathbf{diag}(\nu) \succeq 0 \end{array}$$

gives a lower bound for max-cut problem

$$\begin{array}{ll} \mbox{minimize} & x^T W x \\ \mbox{subject to} & x_i^2 = 1,\ i = 1, \ldots, n \end{array}$$

Strong duality

- strong duality implies $d^* = p^*$
  - not necessarily hold; does not hold in general
  - *usually* holds for convex optimization problems
  - conditions which guarantee strong duality in convex problems called *constraint qualifications*

Slater's constraint qualification*

- strong duality holds for a convex optimization problem:

$$\begin{array}{ll}\text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0,\ i = 1, \ldots, m \\ & Ax = b\end{array}$$

  - if it is strictly feasible, *i.e.*, there exists $x \in \mathbf{R}^n$ such that

$$f_i(x) < 0,\ i = 1, \ldots, m,\ Ax = b$$

- Slater's condition
  - also guarantees the dual optimum is attained (if $p^* > -\infty$)
  - linear inequalities do not need to hold with strict inequalities

Duality example: LP

- primal problem:

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b\end{array}$$

- dual function:

$$g(\lambda) = \inf_x \left(\left(c + A^T \lambda\right)^T x - b^T \lambda \right) = \begin{cases} -b^T \lambda & \text{if } A^T \lambda + c = 0 \\ -\infty & \text{otherwise} \end{cases}$$

- dual problem:

$$\begin{array}{ll}\text{maximize} & -b^T \lambda \\ \text{subject to} & A^T \lambda + c = 0 \\ & \lambda \succeq 0\end{array}$$

  - Slater's condition implies that $p^* = d^*$ if $A\tilde{x} \prec b$ for some $\tilde{x}$
  - truth is, $p^* = d^*$ except when both primal and dual are infeasible

Duality example: QP*

- primal problem (assuming $P \in \mathbf{S}^n_{++}$):

$$\begin{array}{ll} \text{minimize} & x^T P x \\ \text{subject to} & Ax \preceq b \end{array}$$

- dual function:

$$g(\lambda) = \inf_x \left(x^T P x + \lambda^T (Ax - b) \right) = -\frac{1}{4} \lambda^T A P^{-1} A^T \lambda - b^T \lambda$$

- dual problem:

$$\begin{array}{ll} \text{maximize} & -\lambda^T A P^{-1} A^T \lambda / 4 - b^T \lambda \\ \text{subject to} & \lambda \succeq 0 \end{array}$$

  - Slater's condition implies that $p^* = d^*$ if $A\tilde{x} \prec b$ for some $\tilde{x}$
  - truth is, $p^* = d^*$ always!

Complementary slackness*

- assume strong dualtiy holds, x^* is primal optimal, and (λ^*, ν^*) is dual optimal

$$
\begin{aligned}
f_0(x^*) &= g(\lambda^*, \nu^*) = \inf_x \left(f_0(x) + \sum_{i=1}^m \lambda_i^* f_i(x) + \sum_{i=1}^p \nu_i^* h_i(x) \right) \\
&\leq f_0(x^*) + \sum_{i=1}^m \lambda_i^* f_i(x^*) + \sum_{i=1}^p \nu_i^* h_i(x^*) \\
&\leq f_0(x^*)
\end{aligned}
$$

- thus, all inequalities are tight, $i.e.$, they hold with equalities
  - x^* minimizes $L(x, \lambda^*, \nu^*)$
  - $\lambda_i^* f_i(x^*) = 0$ for all i, known as *complementary slackness*

$$
\lambda_i^* > 0 \Rightarrow f_i(x^*) = 0, \qquad f_i(x^*) < 0 \Rightarrow \lambda_i^* = 0
$$

Karush-Kuhn-Tucker (KKT) conditions*

- KKT (optimality) conditions consist of
  - primal feasibility: $f_i(x) \leq 0$ for all $1 \leq i \leq m$, $h_i(x) = 0$ for all $1 \leq i \leq p$
  - dual feasibility: $\lambda \succeq 0$
  - complementary slackness: $\lambda_i f_i(x) = 0$
  - zero gradient of Lagrangian: $\nabla f_0(x) + \sum_{i=1}^m \lambda_i \nabla f_i(x) + \sum_{i=1}^p \nu_i \nabla h_i(x) = 0$
- if strong daulity holds and x^*, λ^*, and ν^* are optimal, they satisfy KKT condtions!

KKT conditions for convex optimization problem*

- if $\tilde{x}$, $\tilde{\lambda}$, and $\tilde{\nu}$ satisfy KKT for convex optimization problem, then they are optimal!
  - complementary slackness implies $f_0(\tilde{x}) = L(\tilde{x}, \tilde{\lambda}, \tilde{\nu})$
  - last conidtion together with convexity implies $g(\tilde{\lambda}, \tilde{\nu}) = L(\tilde{x}, \tilde{\lambda}, \tilde{\nu})$
- thus, for example, if Slater's condition is satisfied, x is optimal if and only if there exist λ, ν that satisfy KKT conditions
  - Slater's condition implies strong dualtiy, hence dual optimum is attained
  - this generalizes optimality condition $\nabla f_0(x) = 0$ for unconstrained problem

Alternating Direction Method of Multipliers (ADMM)

REFERENCE:

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein

Distributed optimization and statistical learning via the alternating direction method of multipliers

What is ADMM for?

- ADMM is for
  - ML with huge data sets
  - distributed optimization where
  - MANY local agents solving large problem by iteratively solving small problems while being coordinated by ONE central agent

Dual ascent method

- consider convex equality-constrained optimization problem:

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

- Lagrangian defined by $L(x, y) = f(x) + y^T(Ax - b)$
- dual function defined by

$$g(y) = \inf_x L(x, y) = -\sup_x((-A^T y)^T x - f(x)) - b^T y = -f^*(-A^T y) - b^T y$$

- dual probem defined by

$$\text{maximize} \quad g(y)$$

Dual ascent method

- gradient method for dual problem:

$$y^{k+1} = y^k + \alpha^k \nabla g(y^k)$$

  where $\nabla g(y) = A\tilde{x} - b$ with $\tilde{x} = \mathrm{argmin}_x L(x, y)$

- this fact induces the following *dual ascent method*:

$$\begin{aligned} x^{k+1} &:= \underset{x}{\mathrm{argmin}}\, L(x, y^k) \\ y^{k+1} &:= y^k + \alpha^k (Ax^{k+1} - b) \end{aligned}$$

  - consists of two stes; x-minimization and dual update

Dual decomposition

- suppose that f is separable in $x_1, \ldots, x_N$, *i.e.*,

$$f(x) = f_1(x_1) + \cdots + f_N(x_N)$$

where $x = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^T$

- then, L is separable, too, since

$$L(x,y) = \sum_{i=1}^N f_i(x_i) + y^T \left(\sum_{i=1}^N A_i x_i - b \right) = \sum_{i=1}^N (f_i(x_i) + y^T A_i x_i) - b^T y$$

- thus, x-minimization step splits into N separate minimizations:

$${x_i}^{k+1} = \operatorname*{argmin}_{x_i} L_i(x_i, y^k) = \operatorname*{argmin}_{x_i} (f_i(x_i) + y^T A_i x_i)$$

- parallelism can be employed!

Method of multipliers

- dual ascent fails, $e.g.$, when f is an affine function in x!
- one solution: *augmented Lagrangian*

$$L_\rho(x, y) = f(x) + y^T(Ax - b) + (\rho/2)\|Ax - b\|_2^2$$

- method of multipliers:

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

Optimality condition*

- optimality conditions: $Ax^* - b = 0, \; \nabla f(x^*) + A^T y^* = 0$
- x^{k+1} minimizes $L_\rho(x, y^k)$, hence

$$\begin{aligned} 0 &= \nabla_x L_\rho(x^{k+1}, y^k) \\ &= \nabla_x f(x^{k+1}) + A^T(y^k + \rho(Ax^{k+1} - b)) \\ &= \nabla_x f(x^{k+1}) + A^T y^{k+1} \end{aligned}$$

- thus, *dual feasibility* achieved!
- *primal feasibility* achieved in limit: $\lim_{k\to\infty} Ax^{k+1} = b$

Pros and cons of method of multipliers

- pros: it works even for nondifferentiable or affine f possibly with $+\infty$ value
- cons: the penalty term deprives it of its capability of parallelism!

ADMM

- ADMM
  - retains the robustness of method of multipliers
    * can deal with nondifferentiable f
    * can deal with affine f
    * can deal with f with $+\infty$ value
  - supports decomposition, hence parallelism
- also called "robust dual decomposition" or "decomposable method of multipliers"

ADMM formulation and algorithm

- ADMM formulation:

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

  where f and g convex

- then, *augmented* Lagrangian defined by

$$L_\rho(x,z,y) = f(x) + g(z) + y^T(Ax+Bz-c) + (\rho/2)\|Ax+Bz-c\|_2^2$$

- finally, ADMM steps:

$$\begin{array}{ll} x\mbox{-minimization:} & x^{k+1} := \mathop{\rm argmin}_x L_\rho(x, z^k, y^k) \\ z\mbox{-minimization:} & z^{k+1} := \mathop{\rm argmin}_z L_\rho(x^{k+1}, z, y^k) \\ \mbox{dual update:} & y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \end{array}$$

Optimality conditions*

- optimality conditions

$$\begin{array}{ll} \text{primal feasibility:} & Ax + Bz - c = 0 \\ \text{dual feasibility:} & \nabla f(x) + A^T y = 0,\ \nabla g(z) + B^T y = 0 \end{array}$$

- since z^{k+1} minimizes $L_\rho(x^{k+1}, z, y^k)$,

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

  - thus, $(x^{k+1}, z^{k+1}, y^{k+1})$ satisfies the second dual feasibility condition!
- primal feasibility and the first dual feasibility are achieved as $k \to \infty$

ADMM in scaled form*

- rewrite augmented Lagrangian with $r = Ax + Bz - c$ and $u = (1/\rho)y$:

$$
\begin{array}{rcl}
L_\rho(x,z,y) & = & f(x) + g(z) + y^T(Ax+Bz-c) + (\rho/2)\|Ax+Bz-c\|_2^2 \\
& = & f(x) + g(z) + (\rho/2)(\|r\|_2^2 + (2/\rho)y^T r) \\
& = & f(x) + g(z) + (\rho/2)(\|r + (1/\rho)y\|_2^2 - \|(1/\rho)y\|_2^2) \\
& = & f(x) + g(z) + (\rho/2)\|Ax+Bz-c+u\|_2^2 - (\rho/2)\|u\|_2^2
\end{array}
$$

- ADMM in scaled form: (with $u^k := (1/\rho)y^k$)

$$
\begin{array}{ll}
x\text{-minimization:} & x^{k+1} := \mathop{\rm argmin}_x(f(x) + (\rho/2)\|Ax + Bz^k - c + u^k\|_2^2) \\
z\text{-minimization:} & z^{k+1} := \mathop{\rm argmin}_z(g(z) + (\rho/2)\|Ax^{k+1} + Bz - c + u^k\|_2^2) \\
\text{dual update:} & u^{k+1} := u^k + (Ax^{k+1} + Bz^{k+1} - c)
\end{array}
$$

- Note that $u^k = u^0 + \sum_{i=1}^k r^i$ with $r^k = Ax^k + Bz^k - c$

Convergence

- assuming that
  - f and g are convex, closed, proper, $i.e.$,

$$\{(x,t) \in \mathbf{R}^n \times \mathbf{R} \mid f(x) \leq t\},\ \{(z,t) \in \mathbf{R}^n \times \mathbf{R} \mid g(x) \leq t\}$$

    are closed, nonempty, convex sets
  - L_0 has a saddle point, $i.e.$, existence of (x^*, z^*, y^*) such that

$$L_0(x^*, z^*, y) \leq L_0(x^*, z^*, y^*) \leq L_0(x, z, y^*)$$

    holds for all x, z, y
- ADMM converges:
  - iterates approach feasibility: $Ax^k + Bz^k - c \to 0$
  - objective approaches optimal value: $f(x^k) + g(z^k) \to p^*$

Related algorithms*

- Douglas, Peaceman, Rachford, Lions, Mercier: operator spliting methods (1950s, 1979)
- Rockafellar: proximal point algorithm (1976)
- Dykstra's alternating projections algorithm (1983)
- Spingarn's method of partial inverses (1985)
- Rockafellar-Wets progressive hedging (1991)
- Rockafellar, et al.: proximal methods (1976–Present)
- Bregman iterative methods (2008–Present)

Common patterns

- x-update step requires minimizing

$$f(x) + (\rho/2)\|Ax + v^k\|_2^2$$

where $v^k = Bz^k - c + u^k$

- z-update step requires minimizing

$$g(z) + (\rho/2)\|Bz + w^k\|_2^2$$

where $w^k = Ax^{k+1} - c + u^k$

- a few special cases enable the simplification of these updates (by exploting special structures)

Decomposition

- suppose
  - f is block-separable:

$$f(x) = f_1(x_1) + \cdots + f_N(x_N)$$

  - A comformably block separable, $i.e.$, $A^T A$ is block diagonal

$$A^T A = \begin{bmatrix} A_1^T \\ \vdots \\ A_N^T \end{bmatrix} \begin{bmatrix} A_1 & \cdots & A_N \end{bmatrix} = \begin{bmatrix} A_1^T A_1 & 0 & \cdots & 0 \\ 0 & A_2^T A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_N^T A_N \end{bmatrix}$$

- then, x-update splits into N parallel updates of x_i
- the very same thing can be applied to z-udpate

Proximal operator*

- when $A = I$, x-update becomes

$$x^+ = \operatorname*{argmin}_x \left(f(x) + (\rho/2)\|x - v\|_2^2 \right) = \operatorname*{\mathbf{prox}}_{f,\rho}(v)$$

- furthermore,
  - if $f = I_C$, $i.e.$, f is indicator function of $C \subseteq \mathbf{R}^n$, then

$$x^+ := \Pi_C(v),$$

    $i.e.$, projection onto C.
  - if $f = \lambda\|\cdot\|_1$, $i.e.$, f is l_1 norm, then

$$x_i^+ := S_{\lambda/\rho}(v_i),$$

    $i.e.$, soft thresholding where $S_a(v) = (v-a)_+ - (-v-a)_+$

Quadratic objective function*

- assume $f(x) = (1/2)x^T Px + q^T x + r$
- then, x-update becomes

$$
\begin{aligned}
x^+ &= \operatorname*{argmin}_x \left((1/2)x^T Px + q^T x + r + (\rho/2)\|Ax - v\|_2^2\right) \\
&= (P + \rho A^T A)^{-1}(\rho A^T v - q)
\end{aligned}
$$

- matrix inversion lemma implies

$$
(P + \rho A^T A)^{-1} = P^{-1} - \rho P^{-1} A^T (I + \rho A P^{-1} A^T)^{-1} A P^{-1}
$$

- if direct method is used, cache factorization of $P + \rho A^T A$ or $I_+ \rho A P^{-1} A^T$ cen save tremendous of computation efforts

Solutions for general objective functions

- if f is smooth,
- standard methods can be used:
  - Newton's method, gradient method, quasi-Newton's method
  - preconditioned CG, limited-memory BFGS (scale to very large problems)
- other techniques:
  - warm start
  - early stopping with variant (or adaptive) tolerances as algorithm proceeds

Constrained optimization

- generic constrained optimization:

$$\begin{array}{ll}\text{minimize} & f(x)\\ \text{subject to} & x \in \mathcal{C}\end{array}$$

- ADMM form:

$$\begin{array}{ll}\text{minimize} & f(x) + g(z)\\ \text{subject to} & x - z = 0\end{array}$$

where $g(z) = I_{\mathcal{C}}(z)$

- then, ADMM iterations become:

$$\begin{array}{lll}x^{k+1} & := & \mathrm{argmin}_x \left(f(x) + (\rho/2)\left\|x - z^k + u^k\right\|_2^2\right)\\ z^{k+1} & := & \Pi_{\mathcal{C}}\left(x^{k+1} + u^k\right)\\ u^{k+1} & := & u^k + x^{k+1} - z^{k+1}\end{array}$$

Lasso formulation

- problem formulation:

$$\text{minimize} \quad (1/2)\|Ax - b\|_2^2 + \lambda\|x\|_1$$

- ADMM form:

$$\begin{array}{ll} \text{minimize} & (1/2)\|Ax - b\|_2^2 + \lambda\|z\|_1 \\ \text{subject to} & x - z = 0 \end{array}$$

- ADMM iterations:

$$\begin{array}{lll} x^{k+1} & := & \left(A^T A + \rho I\right)^{-1} \left(A^T b + \rho z^k - y^k\right) \\ z^{k+1} & := & S_{\lambda/\rho} \left(x^{k+1} + y^k/\rho\right) \\ y^{k+1} & := & y^k + \rho \left(x^{k+1} - z^{k+1}\right) \end{array}$$

Lasso computational example

- for dense $A \in \mathbf{R}^{1500\times 5000}$, $i.e.$, 5000 predictors and 1500 measurements

- computation efforts:
  - 1.32 seconds for factorization
  - 0.03 seconds for ADMM iterations
  - 2.97 seconds for lasso solve
  - 4.45 seconds for full regularization path, $e.g.$, 30 λs

- only takes short sciprt

Sparse inverse covariance selection (SICS)*

- S: empirical covariance of samples from $\mathcal{N}(0, \Sigma)$, with Σ^{-1} sparse, $e.g.$, Gaussian Markov random field

- estimate Σ^{-1} via l_1 regularized maximum likelihood:

$$\text{minimize} \quad \mathbf{Tr}(SX) - \log\det X + \lambda \|X\|_1$$

- methods: COVSEL (Banerjee et al 2008) or graphical lasso (Friedman, Hastie, and Tibshirani, 2007)

SICS via ADMM*

- SICS problem:

$$\text{minimize} \quad \mathbf{Tr}(SX) - \log\det X + \lambda\|X\|_1$$

- ADMM form:

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(SX) - \log\det X + \lambda\|Z\|_1 \\ \text{subject to} & X - Z = 0 \end{array}$$

- ADMM iterations:

$$\begin{array}{rcl} X^{k+1} & := & \mathrm{argmin}_X \left(\mathbf{Tr}(SX) - \log\det X + (\rho/2)\|X - Z^k + U^k\|_F^2\right) \\ Z^{k+1} & := & S_{\lambda/\rho}\left(X^{k+1} + U^k\right) \\ U^{k+1} & := & U^k + (X^{k+1} - Z^{k+1}) \end{array}$$

Solution for X-update*

- eigenvalue decomposition:

$$\rho(Z^k - U^k) - S = Q\Lambda Q^T$$

- diagonal matrix forming:

$$\tilde{X}_{ii} = \frac{\lambda_i + \sqrt{\lambda_i^2 + 4\rho}}{2\rho}$$

- then, X-udpate can be achieved by

$$X^{k+1} = Q\tilde{X}Q^T$$

SICS example

- for $\Sigma^{-1} \in \mathbf{R}^{1000\times 10000}$ with 10000 nonzero entries

- ADMM takes 3–10 minutes

- for comparision,
  - COVSEL takes > 25 minutes when Σ^{-1} is 400×400 tridiagonal matrix

Consensus optimization (CO)

- sum of N functions as objective

$$\text{minimize} \quad \sum_{i=1}^N f_i(x)$$

  - for example, f_i could be the loss function of ith training data block

- ADMM form:

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

  - x_i is ith local variable
  - z is the global variable
  - $x_i - z = 0$ are *consistency* or *consensus constraints*
  - regularization can be added via $g(z)$

CO using ADMM

- Lagrangian:

$$L_\rho(x, z, y) = \sum_{i=1}^{N} \left(f_i(x_i) + y_i^T (x_i - z) + (\rho/2)\|x_i - z\|_2^2 \right)$$

- ADMM iterations:

$$\begin{aligned} x_i^{k+1} &:= \mathop{\mathrm{argmin}}_{x_i} \left(f_i(x_i) + {y_i^k}^T (x_i - z) + (\rho/2)\|x_i - z\|_2^2 \right) \\ z_i^{k+1} &:= \frac{1}{N} \sum_{i=1}^{N} \left(x_i^{k+1} + (1/\rho) y_i^k \right) \\ y_i^{k+1} &:= y_i^k + \rho(x_i^{k+1} - z^{k+1}) \end{aligned}$$

Consensus classification

- given data set, (a_i, b_i), $i = 1, \ldots, N$ where $a_i \in \mathbf{R}^n$, $b_i \in \{-1, 1\}$
- linear classifier $\mathbf{sign}(a^T w + v)$ with *(vector) weight* or *support vector* w, offset v
- margin for ith data is $b_i(a_i^T w + v)$
- loss for ith data is $l\left(b_i(a_i^T w + v)\right)$ where l is loss function, $e.g.$, hinge, logistic, probit, exponential, $etc.$
- choose w, v so as to minimize

$$\frac{1}{N} \sum_{i=1}^{N} l(b_i(a_i^T w + v)) + r(w)$$

  – $r(w)$ is regularization term, $e.g.$, l_2, l_1, l_p, $etc.$
- split data and use ADMM consensus to solve the optimization problem

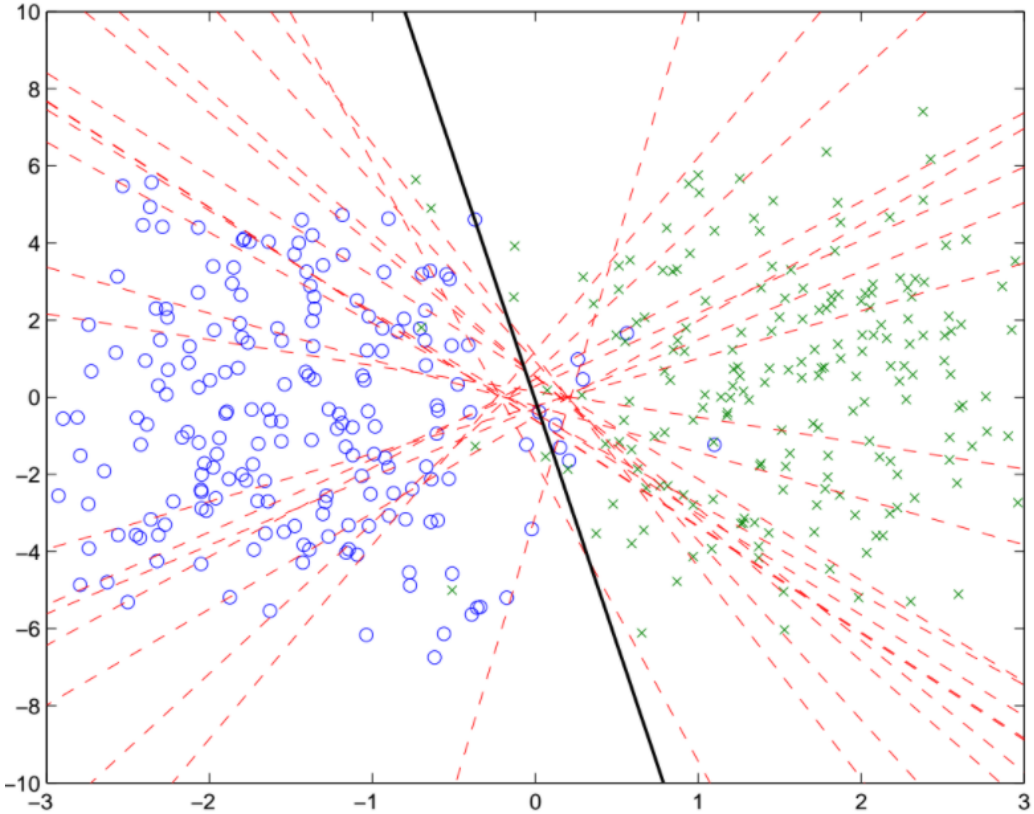
Consensus SVM example

- hinge loss $l(u) = (l - u)_+$ with l_2 regularization

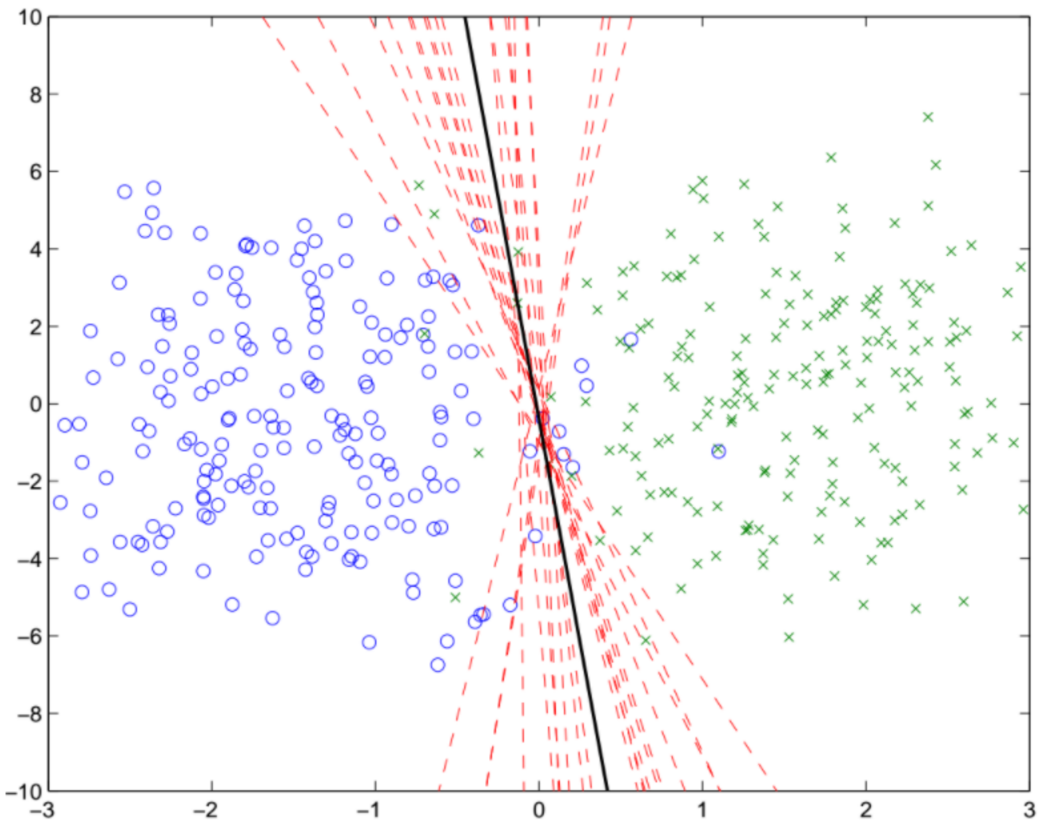
- toy problem with $n = 2$, $N = 400$ to illustrate

- data split into 20 groups, in worst possible way: each group contains only positive or negative data

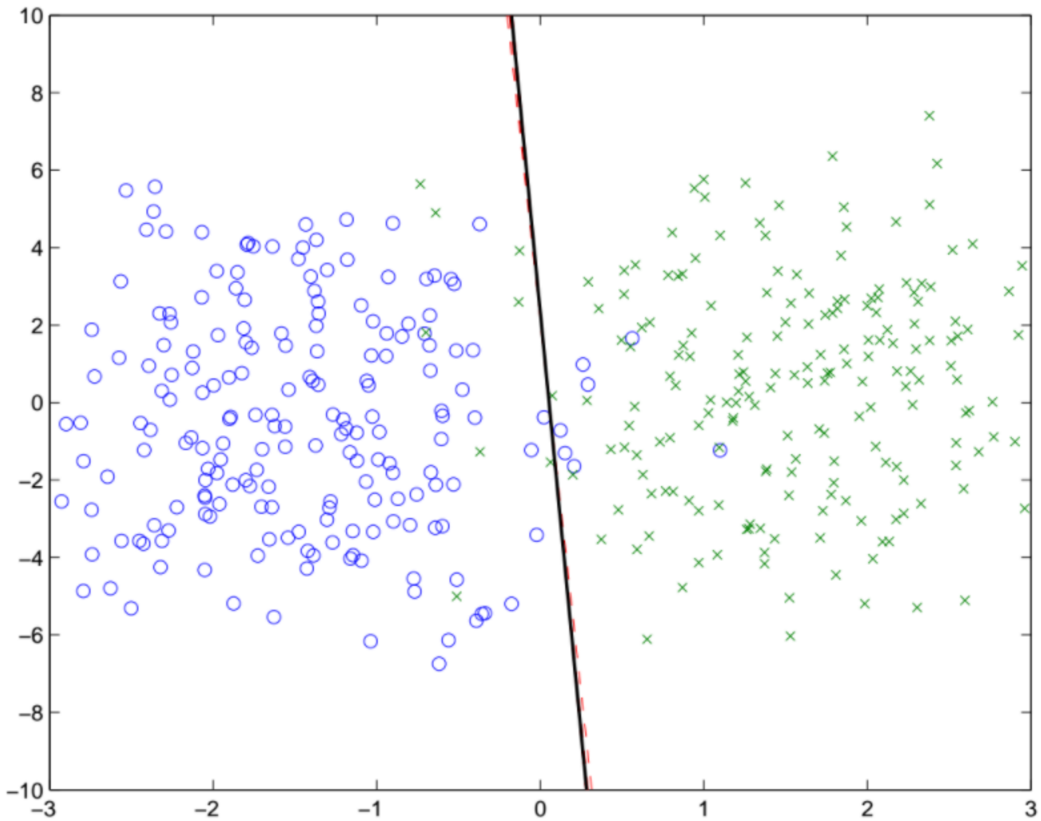
The 1st Epoch



The 5th Epoch



The 40th Epoch



Distributed lasso

- example with *dense* $A \in \mathbf{R}^{m \times n}$ where $m = 400,000$ and $n = 8,000$
  - distributed solver written in C using MPI and GSL
  - no optimization or tuned libraries (like ATLAS, MKL)
  - split into 80 subsystems across 10 (8-core) machines
- computation efforts:
  - 30 seconds for loading data
  - 5 seconds for factorization
  - 2 seconds for subsequent ADMM iterations
  - 6 seconds for lasso solve (~ 15 ADMM iterations)

Exchange problem*

- typical problem formulation:

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

  - dual of consensus
  - constraint is called *equilibrium* or *market clearing* constraint
- one interpretation: N agents exchanging n items so as to minimize total cost
  - $(x_i)_j > 0 \Leftrightarrow$ agent i *receives* $(x_i)_j$ of item j from exchange
  - $(x_i)_j < 0 \Leftrightarrow$ agent i *contributes* $-(x_i)_j$ of item j to exchange
- duality interpretation:
  - y^*, *i.e.*, optimal dual variable, can be interpreted as *valid prices* for items
  - real (or virtual) cash payment $(y^*)^T x_i$ by agent i

ADMM conclusions

- ADMM
  - provides single algorithm framework competitive with special algorithms
  - induces *systematic distributed* algorithms with convergence proof
  - whereas all federated learning based on (asynchronous) update does not provide systematic learning
  - can be easily applied to non-convex cases
- the underlying idea can be used for many ML areas
  - computer vision (CV), natural language processing (NLP), classical statistical learning