Accelerated Randomized Primal-Dual Coordinate Method for Empirical Risk Minimization

Lin Xiao (Microsoft Research)

Joint work with
Qihang Lin (CMU), Zhaosong Lu (Simon Fraser)
Yuchen Zhang (UC Berkeley)

Optimization without Borders
Les Houches, February 9, 2016

Outline

- empirical risk minimization for linear predictors
- brief overview of some efficient randomized algorithms
- accelerated randomized proximal coordinate gradient (APCG) method
- accelerated stochastic primal-dual coordinate (SPDC) method

Empirical Risk Minimization (ERM)

- a generic convex optimization problem in machine learning

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad \left\{ P(x) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \phi_i(a_i^T x) + g(x) \right\}$$

  - training examples: $(a_1, b_1), \ldots, (a_n, b_n)$, $a_i \in \mathbb{R}^d$, $b \in \mathbb{R}$
  - loss function ϕ_i measures prediction quality
  - regularization function $g(x)$ to reduce over-fitting

  both n and d can be very big ($\sim 10^9$), but each a_i very sparse

Empirical Risk Minimization (ERM)

- a generic convex optimization problem in machine learning

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad \left\{ P(x) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \phi_i(a_i^T x) + g(x) \right\}$$

  - training examples: $(a_1, b_1), \ldots, (a_n, b_n)$, $a_i \in \mathbb{R}^d$, $b \in \mathbb{R}$
  - loss function ϕ_i measures prediction quality
  - regularization function $g(x)$ to reduce over-fitting

  both n and d can be very big ($\sim 10^9$), but each a_i very sparse

- examples
  - SVM: $\phi_i(z) = \max\{0, 1 - b_i z\}$ and $g(x) = (\lambda/2)\|x\|_2^2$
  - logistic regression: $\phi_i(z) = \log(1 + \exp(-b_i z))$
  - ridge regression: $\phi_i(z) = (1/2)(z - b_i)^2$, $g(x) = (\lambda/2)\|x\|_2^2$
  - the Lasso: $\phi_i(z) = (1/2)(z - b_i)^2$ and $g(x) = \lambda\|x\|_1$

Minimizing finite average of convex functions

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

- **batch gradient method**

$$x^{(t+1)} = x^{(t)} - \alpha_t \nabla F(x^{(t)})$$

  - each step very expensive, (hopefully) fast convergence
  - can also use quasi-Newton or accelerated gradient methods

Minimizing finite average of convex functions

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

- **batch gradient method**

$$x^{(t+1)} = x^{(t)} - \alpha_t \nabla F(x^{(t)})$$

  - each step very expensive, (hopefully) fast convergence
  - can also use quasi-Newton or accelerated gradient methods

- **stochastic gradient method** (stochastic approximation)

$$x^{(t+1)} = x^{(t)} - \eta_t \nabla f_{i_t}(x^{(t)}) \qquad (i_t \text{ chosen randomly})$$

  - each iteration very cheap, but slow convergence
  - accelerated stochastic algorithms do not really help

Minimizing finite average of convex functions

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

- **batch gradient method**

$$x^{(t+1)} = x^{(t)} - \alpha_t \nabla F(x^{(t)})$$

  - each step very expensive, (hopefully) fast convergence
  - can also use quasi-Newton or accelerated gradient methods

- **stochastic gradient method** (stochastic approximation)

$$x^{(t+1)} = x^{(t)} - \eta_t \nabla f_{i_t}(x^{(t)}) \qquad (i_t \text{ chosen randomly})$$

  - each iteration very cheap, but slow convergence
  - accelerated stochastic algorithms do not really help

- recent advances in **randomized algorithms**:
  exploit finite average (sum) structure to get best of both worlds

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \left\{ P(x) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \phi_i(a_i^T x) + g(x) \right\}$$

- **assumptions:**
  - each ϕ_i is L-smooth: $|\phi_i'(\alpha) - \phi_i'(\beta)| \leq L|\alpha - \beta|$
  - regularizer g is λ-strongly convex

$$g(y) \geq g(x) + g'(y)^T(x - y) + \frac{\lambda}{2}\|x - y\|_2^2, \quad \forall\, x, y \in \mathbb{R}^n$$

- **examples**
  - squared loss $\phi_i(z) = (1/2)(z - b_i)^2$ is 1-smooth
  - logistic loss $\phi_i(z) = \log(1 + \exp(-b_i z))$ is 1/4-smooth

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \left\{ P(x) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \phi_i(a_i^T x) + g(x) \right\}$$

- **assumptions:**
  - each ϕ_i is L-smooth: $|\phi_i'(\alpha) - \phi_i'(\beta)| \leq L|\alpha - \beta|$
  - regularizer g is λ-strongly convex

$$g(y) \geq g(x) + g'(y)^T(x-y) + \frac{\lambda}{2}\|x-y\|_2^2, \quad \forall\, x, y \in \mathbb{R}^n$$

- **examples**
  - squared loss $\phi_i(z) = (1/2)(z - b_i)^2$ is 1-smooth
  - logistic loss $\phi_i(z) = \log(1 + \exp(-b_i z))$ is $1/4$-smooth
- **condition number**
  - let $A = [a_1, \ldots, a_n]$, assume $\max_i \|a_i\|_2 \leq R$, define $\kappa = \frac{L}{\lambda} R^2$
  - worst-case condition number for batch optimization: $1 + \kappa$

$$\frac{1}{n}\frac{L}{\lambda}\|A\|_2^2 \;\leq\; \frac{1}{n}\frac{L}{\lambda}\|A\|_F^2 \;\leq\; \frac{1}{n}\frac{L}{\lambda} n R^2 \;=\; \kappa$$

Condition number and batch complexity

- **condition number:** $\kappa = \frac{L}{\lambda} R^2$ (considering $\kappa \gg 1$)
- **batch complexity:** number of equivalent passes over dataset

Condition number and batch complexity

- **condition number:** $\kappa = \frac{L}{\lambda} R^2$ (considering $\kappa \gg 1$)
- **batch complexity:** number of equivalent passes over dataset

complexities to reach $\mathbf{E}[P(x^{(k)}) - P^\star] \leq \epsilon$

algorithm	iteration complexity	batch complexity
stochastic gradient	$(1+\kappa)/\epsilon$	$(1+\kappa)/(n\epsilon)$
full gradient (FG)	$(1+\kappa)\log(1/\epsilon)$	$(1+\kappa)\log(1/\epsilon)$
accelerated FG (Nesterov)	$(1+\sqrt{\kappa})\log(1/\epsilon)$	$(1+\sqrt{\kappa})\log(1/\epsilon)$
SDCA, SAG(A), SVRG, ...	$(n+\kappa)\log(1/\epsilon)$	$(1+\kappa/n)\log(1/\epsilon)$
A-SDCA, **APCG, SPDC**	$(n+\sqrt{\kappa n})\log(1/\epsilon)$	$(1+\sqrt{\kappa/n})\log(1/\epsilon)$

SDCA: Shalev-Shwartz & Zhang (2013)
SAG: Schmidt, Le Roux, & Bach (2012, 2013)
Finito: Defazio, Caetano & Domke (2014)
SVRG: Johnson & Zhang (2013), X. & Zhang (2014)
Quartz: Qu, Richtárik, & Zhang (2015)
Catalyst: Lin, Mairal, & Harchaoui (2015)
SAGA: Defazio, Bach & Lacoste-Julien (2014)
A-SDCA: Shalev-Shwartz & Zhang (2014)
MISO: Mairal (2015)
APCG: Lin, Lu & X. (2014)
SPDC: Zhang & X. (2015)
A-APPA Frostig, Ge, Kakade, &Sidford (2015)

lower bound: Agarwal & Bottou (2015), Lan (2015, RPDG method)

Outline

- empirical risk minimization for linear predictors
- **brief overview of some efficient randomized algorithms**
- accelerated randomized proximal coordinate gradient (APCG) method
- accelerated stochastic primal-dual coordinate (SPDC) method

Stochastic average gradient (SAG)

- batch gradient method

$$x^{(k+1)} = x^{(k)} - \frac{\alpha_k}{n} \sum_{i=1}^{n} \nabla f_i(x^{(k)})$$

Stochastic average gradient (SAG)

- batch gradient method

$$x^{(k+1)} = x^{(k)} - \frac{\alpha_k}{n} \sum_{i=1}^{n} \nabla f_i(x^{(k)})$$

- SAG method (Le Roux, Schmidt, Bach 2012)

$$x^{(k+1)} = x^{(k)} - \frac{\alpha_k}{n} \sum_{i=1}^{n} g_i^{(k)} \quad \text{where} \quad g_i^{(k)} = \begin{cases} \nabla f_i(x^{(k)}) & \text{if } i = i_k \\ g_i^{(k-1)} & \text{otherwise} \end{cases}$$

  - complexity (gradient evaluations): $O(\max\{n, \kappa\} \log \frac{1}{\epsilon})$
    cf. full gradient: $O(n\kappa \log \frac{1}{\epsilon})$ and stochastic gradient: $O(\frac{\kappa}{\epsilon})$
  - need to store most recent gradient of each component, but can be avoided for some structured problems

Stochastic variance reduced gradient (SVRG)

- SVRG (Johnson & Zhang 2013)
  - update form

$$x^{(k+1)} = x^{(k)} - \eta(\nabla f_{i_k}(x^{(k)}) - \nabla f_{i_k}(\tilde{x}) + \nabla F(\tilde{x}))$$

  - update $\tilde{x}$ periodically (every few passes)

Stochastic variance reduced gradient (SVRG)

- SVRG (Johnson & Zhang 2013)
  - update form

$$x^{(k+1)} = x^{(k)} - \eta(\nabla f_{i_k}(x^{(k)}) - \nabla f_{i_k}(\tilde{x}) + \nabla F(\tilde{x}))$$

  - update $\tilde{x}$ periodically (every few passes)
- still a stochastic gradient method

$$\mathbf{E}_{i_k}[\nabla f_{i_k}(x^{(k)}) - \nabla f_{i_k}(\tilde{x}) + \nabla F(\tilde{x})] = \nabla F(x^{(k)})$$

  - expected update direction is the same as $\mathbf{E}[\nabla f_{i_k}(x^{(k)})]$
  - variance can be diminishing if $\tilde{x}$ updated periodically
- complexity: $O\left((n+\kappa)\log\frac{1}{\epsilon}\right)$, cf. SAG: $O(\max\{n,\kappa\}\log\frac{1}{\epsilon})$
- SAGA (Defazio et al. 2014): unbiased variant of SAG

More structure: dual ERM problem

primal problem

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad \left\{ P(x) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \phi_i(a_i^T x) + g(x) \right\}$$

dual problem

$$\underset{y\in\mathbb{R}^n}{\text{maximize}} \quad \left\{ D(y) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} -\phi_i^*(y_i) - g^*\left(-\frac{1}{n}\sum_{i=1}^{n} y_i a_i\right) \right\}$$

where g^* and ϕ_i^* are convex conjugate functions

- $g^*(u) = \sup_{x\in\mathbb{R}^d}\{x^T u - g(x)\}$
- $\phi_i^*(y_i) = \sup_{z\in\mathbb{R}}\{y_i z - \phi_i(z)\}$, for $i = 1, \ldots, n$

Duality

- **assumptions:**
  - each ϕ_i is $1/\gamma$-smooth $\Longrightarrow$ ϕ_i^* is γ-strongly convex

$$|\phi_i'(\alpha) - \phi_i'(\beta)| \leq (1/\gamma)|\alpha - \beta|, \quad \forall\, \alpha, \beta \in \mathbb{R}$$

  - g is λ-strongly convex $\Longrightarrow$ g^* is $1/\lambda$-smooth

$$g(y) \geq g(x) + g'(y)^T(x - y) + \frac{\lambda}{2}\|x - y\|_2^2, \quad \forall\, x, y \in \mathbb{R}^n$$

- **weak duality**:
  - $P(x) \geq D(y)$ for all $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^n$
  - duality gap $P(x) - D(y) \geq P(x) - P(x^\star)$
- **strong duality**
  - there exist unique $(x^\star, y^\star)$ satisfying $P(x^\star) = D(y^\star)$
  - $x^\star = \nabla g^*\left(-\frac{1}{n}\sum_{i=1}^n y_i^\star a_i\right)$

Stochastic Dual Coordinate Ascent (SDCA)

$$\underset{y\in\mathbb{R}^n}{\text{maximize}} \left\{ D(y) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} -\phi_i^*(y_i) - g^*\left(-\frac{1}{n}\sum_{i=1}^{n} y_i a_i\right)\right\}$$

Initialize: $x^{(0)} = 0,\ y^{(0)} = 0,\ u^{(0)} = (1/n)\sum_{i=1}^n y_i^{(0)} a_i$

for $t = 0, 1, 2, \dots, T-1$

pick $k \in \{1, 2, \dots, n\}$ uniformly at random, and update

$$y_k^{(t+1)} = \arg\max_{\alpha} \left\{ -\phi_i^*(\alpha) + (a_k^T x^{(t)})(\alpha - y_k^{(t)}) - \frac{\|a_k\|_2^2}{2\lambda n}\left(\alpha - y_k^{(t)}\right)^2 \right\}$$

$$u^{(t+1)} = u^{(t)} + \frac{1}{n}(y_k^{(t+1)} - y_k^{(t)}) a_k$$

$$x^{(t+1)} = \nabla g^*(u^{(t+1)}) \qquad \left(x^{(t+1)} = \frac{1}{\lambda} u^{(t+1)} \text{ if } g(x) = \frac{\lambda}{2}\|x\|_2^2\right)$$

Hsieh, Chang, Lin, Keerthi & Sundararajan (2008), Shalev-Shwartz & Zhang (2013)

same complexity as SAG and SVRG: $O\left((n+\kappa)\log\frac{1}{\epsilon}\right)$

Outline

- empirical risk minimization for linear predictors
- brief overview of some efficient randomized algorithms
- **accelerated randomized proximal coordinate gradient (APCG) method**
  (Qihang Lin, Zhaosong Lu & X., SIOPT 2015)
- accelerated stochastic primal-dual coordinate (SPDC) method

(Block) coordinate descent method

- problem: minimize sum of two convex functions:

$$\underset{x \in \mathbb{R}^N}{\text{minimize}} \quad f(x) + \sum_{i=1}^{n} \Psi_i(x_i)$$

  - f smooth, Ψ_i may be nondifferentiable but $\text{prox}_{\Psi_i}(\cdot)$ simple
  - $x = (x_1, \ldots, x_n)$ with $x_i \in \mathbb{R}^{N_i}$ and $\sum_{i=1}^{n} N_i = N$

(Block) coordinate descent method

- problem: minimize sum of two convex functions:

$$\underset{x \in \mathbb{R}^N}{\text{minimize}} \quad f(x) + \sum_{i=1}^{n} \Psi_i(x_i)$$

  - f smooth, Ψ_i may be nondifferentiable but $\text{prox}_{\Psi_i}(\cdot)$ simple
  - $x = (x_1, \ldots, x_n)$ with $x_i \in \mathbb{R}^{N_i}$ and $\sum_{i=1}^n N_i = N$

- algorithm (Richtárik & Takáč, 2014): iterate for $k = 0, 1, 2, \ldots$
  1. choose coordinate i_k randomly
  2. update

$$x_i^{(k+1)} = \begin{cases} \text{prox}_{\eta \Psi_i}\left(x_i^{(k)} - \eta \nabla_{i_k} f(x^{(k)})\right) & \text{if } i = i_k \\ x_i^{(k)} & \text{if } i \neq i_k \end{cases}$$

(Block) coordinate descent method

- problem: minimize sum of two convex functions:

$$\underset{x \in \mathbb{R}^N}{\text{minimize}} \quad f(x) + \sum_{i=1}^{n} \Psi_i(x_i)$$

  - f smooth, Ψ_i may be nondifferentiable but $\text{prox}_{\Psi_i}(\cdot)$ simple
  - $x = (x_1, \ldots, x_n)$ with $x_i \in \mathbb{R}^{N_i}$ and $\sum_{i=1}^{n} N_i = N$

- algorithm (Richtárik & Takáč, 2014): iterate for $k = 0, 1, 2, \ldots$
  1. choose coordinate i_k randomly
  2. update

$$x_i^{(k+1)} = \begin{cases} \text{prox}_{\eta \Psi_i}\left(x_i^{(k)} - \eta \nabla_{i_k} f(x^{(k)})\right) & \text{if } i = i_k \\ x_i^{(k)} & \text{if } i \neq i_k \end{cases}$$

- accelerated randomized CD methods:
  - Nesterov (2012): minimizing smooth functions ($\Psi_i \equiv 0$)
  - Fercoq & Ricktárik (2014): accelerated sublinear rate

Accelerated proximal coordinate gradient (APCG)

input: $x^0 \in \mathrm{dom}(\Psi)$ and convexity parameter $\mu \geq 0$.

set $z^0 = x^0$, choose $0 < \gamma_0 \in [\mu, 1]$, and repeat for $k = 0, 1, 2, \ldots$

1. compute $\alpha_k \in (0, \frac{1}{n}]$ from the equation

$$n^2 \alpha_k^2 = (1 - \alpha_k)\,\gamma_k + \alpha_k \mu,$$

and set $\gamma_{k+1} = (1 - \alpha_k)\gamma_k + \alpha_k \mu, \qquad \beta_k = \frac{\alpha_k \mu}{\gamma_{k+1}}.$

2. compute $y^k = \frac{1}{\alpha_k \gamma_k + \gamma_{k+1}} \left(\alpha_k \gamma_k z^k + \gamma_{k+1} x^k \right).$

3. choose $i_k \in \{1, \ldots, n\}$ uniformly at random and compute

$$z^{k+1} = \arg\min_{x \in \mathbb{R}^N} \left\{ \frac{n\alpha_k}{2} \left\| x - (1 - \beta_k) z^k - \beta_k y^k \right\|_L^2 + \langle \nabla_{i_k} f(y^k), x_{i_k} \rangle + \Psi_{i_k}(x_{i_k}) \right\}$$

4. set $x^{k+1} = y^k + n\alpha_k (z^{k+1} - z^k) + \frac{\mu}{n}(z^k - y^k)$.

(if $\mu = 0$, APCG reduces to APPROX of Fercoq and Richtárik 2014)

Convergence analysis

- **assumptions:**
  - smoothness: $\|\nabla_i f(x + U_i v_i) - \nabla_i f(x)\|_2 \leq L_i \|v_i\|_2, \; i = 1, \ldots, n$
  - strong convexity: $f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{\mu}{2} \|y - x\|_L^2$

  (where $\|x\|_L^2 = \sum_{i=1}^n L_i \|x_i\|^2$, and $\mu \leq 1$ represents $1/\kappa$)

Convergence analysis

- **assumptions:**
  - smoothness: $\|\nabla_i f(x + U_i v_i) - \nabla_i f(x)\|_2 \leq L_i \|v_i\|_2, \; i = 1, \ldots, n$
  - strong convexity: $f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{\mu}{2}\|y - x\|_L^2$

  (where $\|x\|_L^2 = \sum_{i=1}^n L_i \|x_i\|^2$, and $\mu \leq 1$ represents $1/\kappa$)

- **theorem:** the sequenced $\{x^k\}$ generated by APCG satisfies

$$\mathbf{E}[F(x^k)] - F^\star \;\leq\; \min\left\{\left(1 - \frac{\sqrt{\mu}}{n}\right)^k, \; \left(\frac{2n}{2n + k\sqrt{\gamma_0}}\right)^2\right\}\left(F(x^0) - F^\star + \frac{\gamma_0}{2}R_0^2\right)$$

  where $R_0 \overset{\text{def}}{=} \min_{x^\star \in X^\star} \|x^0 - x^\star\|_L$ and $\|x\|_L^2 = \sum_{i=1}^n L_i \|x_i\|^2$

- comparisons
  - for $n = 1$, recover results for accelerated full gradient methods
  - for $n > 1$, faster than (un-accelerated) randomized CD methods

APCG with strong convexity

input: $x^0 \in \mathrm{dom}(\Psi)$ and convexity parameter $\mu > 0$

set $\alpha = \frac{\sqrt{\mu}}{n}$ and $z^0 = x^0$, and repeat for $k = 0, 1, 2, \ldots$

1. $y^{(k)} = \frac{x^{(k)} + \alpha z^{(k)}}{1+\alpha}$
2. choose $i_k \in \{1, \ldots, n\}$ uniformly at random and compute

$$z^{(k+1)} = \begin{cases} \mathrm{prox}_{\frac{1}{n\alpha}\Psi_i}\left((1-\alpha) z_i^{(k)} + \alpha y_i^{(k)} - \frac{1}{n\alpha}\nabla_i f(y^{(k)})\right) & \text{if } i = i_k \\ (1-\alpha) z_i^{(k)} + \alpha y_i^{(k)} & \text{if } i \neq i_k \end{cases}$$

3. $x^{(k+1)} = y^{(k)} + n\alpha(z^{(k+1)} - z^{(k)}) + \frac{n\alpha^2}{1+\alpha}(z^{(k)} - x^{(k)})$

convergence rate:

$$\mathbf{E}[F(x^{(k)})] - F^\star \ \le \ \left(1 - \frac{\sqrt{\mu}}{n}\right)^k \left(F(x^{(0)}) - F^\star + \frac{\mu}{2}\|x^{(0)} - x^\star\|_L^2\right)$$

Efficient implementation

input: $x^{(0)} \in \mathrm{dom}(\Psi)$ and convexity parameter $\mu > 0$.

set $\alpha = \frac{\sqrt{\mu}}{n}$ and $\rho = \frac{1-\alpha}{1+\alpha}$, and initialize $u^{(0)} = 0$ and $v^{(0)} = x^{(0)}$.

iterate: repeat for $k = 0, 1, 2, \ldots$

1. choose $i_k \in \{1, \ldots, n\}$ uniformly at random and compute

$$h_{i_k}^{(k)} = \arg\min_{h \in \mathbb{R}^{N_{i_k}}} \Big\{ \frac{n\alpha L_{i_k}}{2} \|h\|_2^2 + \Big\langle \nabla_{i_k} f(\rho^{k+1} u^{(k)} + v^{(k)}),\ h \Big\rangle + \Psi_{i_k}\Big(-\rho^{k+1} u_{i_k}^{(k)} + v_{i_k}^{(k)} + h\Big) \Big\}$$

2. let $u^{(k+1)} = u^{(k)}$ and $v^{(k+1)} = v^{(k)}$, and update

$$u_{i_k}^{(k+1)} = u_{i_k}^{(k)} - \frac{1 - n\alpha}{2\rho^{k+1}} h_{i_k}^{(k)}, \qquad v_{i_k}^{(k+1)} = v_{i_k}^{(k)} + \frac{1 + n\alpha}{2} h_{i_k}^{(k)}$$

equivalence:

$$x^{(k)} = \rho^k u^{(k)} + v^{(k)}, \quad y^{(k)} = \rho^{k+1} u^{(k)} + v^{(k)}, \quad z^{(k)} = -\rho^k u^{(k)} + v^{(k)}$$

Application to dual ERM problem

primal problem

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad \left\{ P(x) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \phi_i(a_i^T x) + g(x) \right\}$$

dual problem

$$\underset{y\in\mathbb{R}^n}{\text{maximize}} \quad \left\{ D(y) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} -\phi_i^*(y_i) - g^*\left(-\frac{1}{n}\sum_{i=1}^{n} y_i a_i \right) \right\}$$

assumptions:

- each ϕ_i is $1/\gamma$-smooth $\Longrightarrow$ ϕ_i^* is γ-strongly convex
- regularizer g is λ-strongly convex $\Longrightarrow$ g^* is $1/\lambda$-smooth

APCG for dual ERM

- relocate strong convexity in $F(y) = f(y) + \sum_{i=1}^{n} \Psi_i(y_i)$

$$f(y) = \lambda g^* \left(-\frac{1}{\lambda n} Ay \right) + \frac{\gamma}{2n} \|y\|_2^2, \quad \Psi_i(y_i) = \frac{1}{n} \left(\phi_i^*(y_i) - \frac{\gamma}{2} \|y_i\|_2^2 \right)$$

  - f is smooth and strongly convex
  - each Ψ_i, for $i = 1, \ldots, n$ still convex

APCG for dual ERM

- relocate strong convexity in $F(y) = f(y) + \sum_{i=1}^{n} \Psi_i(y_i)$

$$f(y) = \lambda g^* \left(-\frac{1}{\lambda n} Ay \right) + \frac{\gamma}{2n} \|y\|_2^2, \quad \Psi_i(y_i) = \frac{1}{n} \left(\phi_i^*(y_i) - \frac{\gamma}{2} \|y_i\|_2^2 \right)$$

  - f is smooth and strongly convex
  - each Ψ_i, for $i = 1, \ldots, n$ still convex

- **theorem:** to obtain $\mathbf{E}[D^\star - D(y^{(t)})] \le \epsilon$, it suffices to have

$$t \;\ge\; \left(n + \sqrt{\frac{nR^2}{\lambda\gamma}} \right) \log(C/\epsilon) \;=\; \left(n + \sqrt{n\kappa} \right) \log(C/\epsilon)$$

  where $R = \max_i \|a_i\|_2$ and $C = D^\star - D(y^{(0)}) + \frac{\gamma}{2n} \|y^{(0)} - y^\star\|_2^2$

- still need to recover primal solution, but complexity stay same

Outline

- empirical risk minimization for linear predictors
- brief overview of some efficient randomized algorithms
- accelerated randomized proximal coordinate gradient (APCG) method
- **accelerated stochastic primal-dual coordinate (SPDC) method**
  (Joint work with Yuchen Zhang)

Saddle-point formulation

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \left\{ P(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \phi_i(a_i^T x) + g(x) \right\}$$

using $\phi_i(a_i^T x) = \max_{y_i\in\mathbb{R}}\{y_i\langle a_i, x\rangle - \phi_i^*(y_i)\}$ to obtain

$$\min_{x\in\mathbb{R}^d} \left\{ \frac{1}{n}\sum_{i=1}^{n} \max_{y_i\in\mathbb{R}}\{y_i\langle a_i, x\rangle - \phi_i^*(y_i)\} + g(x) \right\}$$

$$= \min_{x\in\mathbb{R}^d} \max_{y\in\mathbb{R}^n} \left\{ f(x,y) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} y_i\langle a_i, x\rangle - \phi_i^*(y_i) + g(x) \right\}$$

Saddle-point formulation

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \left\{ P(x) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \phi_i(a_i^T x) + g(x) \right\}$$

using $\phi_i(a_i^T x) = \max_{y_i \in \mathbb{R}} \{ y_i \langle a_i, x \rangle - \phi_i^*(y_i) \}$ to obtain

$$\min_{x \in \mathbb{R}^d} \left\{ \frac{1}{n} \sum_{i=1}^{n} \max_{y_i \in \mathbb{R}} \{ y_i \langle a_i, x \rangle - \phi_i^*(y_i) \} + g(x) \right\}$$

$$= \min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^n} \left\{ f(x, y) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} y_i \langle a_i, x \rangle - \phi_i^*(y_i) + g(x) \right\}$$

- assumptions
  - each ϕ_i is $1/\gamma$-smooth $\Longrightarrow$ ϕ_i^* is γ-strongly convex
  - g is λ-strongly convex

  therefore, saddle-point $(x^\star, y^\star)$ exists and unique
- primal-dual algorithm: alternating between maximizing $f(x, y)$ over y and minimizing $f(x, y)$ over x

Primal-dual algorithm of Chambolle and Pock (2011)

- a class of convex-concave saddle point problem

$$\min_{x\in\mathbb{R}^d} \max_{y\in\mathbb{R}^n} \left\{\langle Kx, y\rangle + G(x) - F^*(y)\right\}$$

  - primal problem: $\min_x \; F(Kx) + G(x)$
  - dual problem: $\max_y -F^*(y) - G^*(-K^T y)$

- first-order primal-dual algorithm

$$y^{(t+1)} = \arg\max_{y\in\mathbb{R}^n} \left\{\langle K\overline{x}^{(t)}, y\rangle - F^*(y) - \frac{1}{2\sigma}\|y - y^{(t)}\|_2^2\right\}$$

$$x^{(t+1)} = \arg\min_{x\in\mathbb{R}^d} \left\{\langle K^T y^{(t+1)}, x\rangle + G(x) + \frac{1}{2\tau}\|x - x^{(t)}\|_2^2\right\}$$

$$\overline{x}^{(t+1)} = x^{(t+1)} + \theta(x^{(t+1)} - x^{(t)})$$

Basic ideas

- saddle point formulation of ERM

$$\min_{x\in\mathbb{R}^d}\ \max_{y\in\mathbb{R}^n}\ \left\{ f(x,y) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n}\big(y_i\langle a_i, x\rangle - \phi_i^*(y_i)\big) + g(x) \right\}$$

  define $K = \frac{1}{n}A,\ G(x) = g(x),\ F^*(y) = \frac{1}{n}\sum_{i=1}^{n}\phi_i^*(y_i)$ to match

$$\min_{x\in\mathbb{R}^d}\ \max_{y\in\mathbb{R}^n}\ \big\{\langle Kx, y\rangle + G(x) - F^*(y)\big\}$$

  - can apply Chambolle-Pock algorithm directly
  - same complexity as Nesterov's accelerated gradient method

Basic ideas

- saddle point formulation of ERM

$$\min_{x\in\mathbb{R}^d} \max_{y\in\mathbb{R}^n} \left\{ f(x,y) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n}\big(y_i\langle a_i, x\rangle - \phi_i^*(y_i)\big) + g(x) \right\}$$

  define $K = \frac{1}{n}A,\ G(x) = g(x),\ F^*(y) = \frac{1}{n}\sum_{i=1}^{n}\phi_i^*(y_i)$ to match

$$\min_{x\in\mathbb{R}^d} \max_{y\in\mathbb{R}^n} \big\{\langle Kx, y\rangle + G(x) - F^*(y)\big\}$$

  - can apply Chambolle-Pock algorithm directly
  - same complexity as Nesterov's accelerated gradient method

- SPDC alternates between
  - maximizing over a randomly chosen dual variable y_i
  - minimizing over the whole primal variable x

  better complexity than accelerated batch gradient method

Algorithm 1: SPDC

inputs: parameters $\tau, \sigma, \theta \in \mathbb{R}_+$, and initial points $x^{(0)}$ and $y^{(0)}$

Initialize: $\overline{x}^{(0)} = x^{(0)}$, $u^{(0)} = (1/n)\sum_{i=1}^{n} y_i^{(0)} a_i$

for $t = 0, 1, 2, \dots, T-1$

pick $k \in \{1, 2, \dots, n\}$ uniformly at random, and update

$$y_i^{(t+1)} = \begin{cases} \arg\max_{\beta\in\mathbb{R}} \left\{ \beta\langle a_i, \overline{x}^{(t)}\rangle - \phi_i^*(\beta) - \frac{1}{2\sigma}(\beta - y_i^{(t)})^2 \right\} & \text{if } i = k, \\ y_i^{(t)} & \text{if } i \neq k, \end{cases}$$

$$u^{(t+1)} = u^{(t)} + \frac{1}{n}(y_k^{(t+1)} - y_k^{(t)})a_k$$

$$x^{(t+1)} = \arg\min_{x\in\mathbb{R}^d} \left\{ g(x) + \left\langle u^{(t)} + n(u^{(t+1)} - u^{(t)})a_k,\ x \right\rangle + \frac{\|x - x^{(t)}\|_2^2}{2\tau} \right\}$$

$$\overline{x}^{(t+1)} = x^{(t+1)} + \theta(x^{(t+1)} - x^{(t)})$$

output: $x^{(T)}$ and $y^{(T)}$

Algorithm 2: Mini-batch SPDC

inputs: parameters $\tau, \sigma, \theta \in \mathbb{R}_+$, $x^{(0)}$ and $y^{(0)}$, mini-batch size m

Initialize: $\overline{x}^{(0)} = x^{(0)}$, $u^{(0)} = (1/n)\sum_{i=1}^{n} y_i^{(0)} a_i$

for $t = 0, 1, 2, \ldots, T-1$

pick $K \subset \{1, 2, \ldots, n\}$ randomly with $|K| = m$, and update

$$y_i^{(t+1)} = \begin{cases} \arg\max_{\beta \in \mathbb{R}} \left\{ \beta \langle a_i, \overline{x}^{(t)} \rangle - \phi_i^*(\beta) - \frac{1}{2\sigma}(\beta - y_i^{(t)})^2 \right\} & \text{if } i \in K \\ y_i^{(t)} & \text{if } i \notin K \end{cases}$$

$$u^{(t+1)} = u^{(t)} + \frac{1}{n} \sum_{k \in K} (y_k^{(t+1)} - y_k^{(t)}) a_k$$

$$x^{(t+1)} = \arg\min_{x \in \mathbb{R}^d} \left\{ g(x) + \left\langle u^{(t)} + \frac{n}{m}(u^{(t+1)} - u^{(t)}),\ x \right\rangle + \frac{\|x - x^{(t)}\|_2^2}{2\tau} \right\}$$

$$\overline{x}^{(t+1)} = x^{(t+1)} + \theta(x^{(t+1)} - x^{(t)})$$

output: $x^{(T)}$ and $y^{(T)}$

Convergence analysis

- assumptions
  - each ϕ_i is $1/\gamma$-smooth
  - g is λ-strongly convex
  - $\|a_i\|_2 \le R$ for $i = 1, \ldots, n$
- **theorem:** if the parameters for mini-batch SPDC are chosen as

$$\tau = \frac{1}{R}\sqrt{\frac{m\gamma}{n\lambda}}, \quad \sigma = \frac{1}{R}\sqrt{\frac{n\lambda}{m\gamma}}, \quad \theta = 1 - \frac{1}{(n/m) + \sqrt{\kappa(n/m)}},$$

where $\kappa = R^2/(\lambda\gamma)$, then we have

$$\mathbf{E}[\|x^{(T)} - x^\star\|_2^2] \le \epsilon \quad \text{and} \quad \mathbf{E}[\|y^{(T)} - y^\star\|_2^2] \le \epsilon,$$

whenever

$$T \ge \left(\frac{n}{m} + \sqrt{\kappa\frac{n}{m}}\right)\log\left(\frac{C}{\epsilon}\right)$$

- **convergence of duality gap:** in order to obtain

$$\mathbf{E}[P(x^{(T)}) - D(y^{(T)})] \leq \epsilon$$

(non-ergodic), it suffices to have

$$T \geq \left(\frac{n}{m} + \sqrt{\kappa \frac{n}{m}}\right) \log\left((1+\kappa)\frac{C'}{\epsilon}\right)$$

- **complexities** (hiding constants and $\log(1/\epsilon)$)

	iteration complexity	batch complexity
$1 \leq m \leq n$	$\mathcal{O}\left((n/m) + \sqrt{\kappa(n/m)}\right)$	$\mathcal{O}\left(1 + \sqrt{\kappa(m/n)}\right)$
$m = n$	$\mathcal{O}\left(1 + \sqrt{\kappa}\right)$	$\mathcal{O}\left(1 + \sqrt{\kappa}\right)$
$m = 1$	$\mathcal{O}\left(n + \sqrt{\kappa n}\right)$	$\mathcal{O}\left(1 + \sqrt{\kappa/n}\right)$

  - smaller batch size m leads to less number of passes over data
  - parallel computing: set m to match number of cores/threads

More careful comparison with batch methods

algorithm	τ	σ	θ	batch complexity
C-P batch	$\frac{\sqrt{n}}{\|A\|_2}\sqrt{\frac{\gamma}{\lambda}}$	$\frac{\sqrt{n}}{\|A\|_2}\sqrt{\frac{\lambda}{\gamma}}$	$1 - \frac{1}{1+\frac{\|A\|_2}{2\sqrt{n\lambda\gamma}}}$	$\left(1 + \frac{\|A\|_2}{2\sqrt{n\lambda\gamma}}\right)\log\frac{1}{\epsilon}$
SPDC $m = n$	$\frac{1}{R}\sqrt{\frac{\gamma}{\lambda}}$	$\frac{1}{R}\sqrt{\frac{\lambda}{\gamma}}$	$1 - \frac{1}{1+\frac{R}{\sqrt{\lambda\gamma}}}$	$\left(1 + \frac{R}{\sqrt{\lambda\gamma}}\right)\log\frac{1}{\epsilon}$
SPDC $m = 1$	$\frac{1}{R}\sqrt{\frac{\gamma}{n\lambda}}$	$\frac{1}{R}\sqrt{\frac{n\lambda}{\gamma}}$	$1 - \frac{1}{n+\frac{\sqrt{n}R}{\sqrt{\lambda\gamma}}}$	$\left(1 + \frac{R}{\sqrt{n\lambda\gamma}}\right)\log\frac{1}{\epsilon}$

C-P batch: Chambolle-Pock (2011)

More careful comparison with batch methods

algorithm	τ	σ	θ	batch complexity
C-P batch	$\frac{\sqrt{n}}{\|A\|_2}\sqrt{\frac{\gamma}{\lambda}}$	$\frac{\sqrt{n}}{\|A\|_2}\sqrt{\frac{\lambda}{\gamma}}$	$1 - \frac{1}{1+\frac{\|A\|_2}{2\sqrt{n\lambda\gamma}}}$	$\left(1 + \frac{\|A\|_2}{2\sqrt{n\lambda\gamma}}\right)\log\frac{1}{\epsilon}$
SPDC $m = n$	$\frac{1}{R}\sqrt{\frac{\gamma}{\lambda}}$	$\frac{1}{R}\sqrt{\frac{\lambda}{\gamma}}$	$1 - \frac{1}{1+\frac{R}{\sqrt{\lambda\gamma}}}$	$\left(1 + \frac{R}{\sqrt{\lambda\gamma}}\right)\log\frac{1}{\epsilon}$
SPDC $m = 1$	$\frac{1}{R}\sqrt{\frac{\gamma}{n\lambda}}$	$\frac{1}{R}\sqrt{\frac{n\lambda}{\gamma}}$	$1 - \frac{1}{n+\frac{\sqrt{n}R}{\sqrt{\lambda\gamma}}}$	$\left(1 + \frac{R}{\sqrt{n\lambda\gamma}}\right)\log\frac{1}{\epsilon}$

C-P batch: Chambolle-Pock (2011)

notice

$$\|A\|_2 \le \|A\|_F \le \sqrt{n}\max_i\{\|a_i\|_2\} = \sqrt{n}R$$

so worst-case complexity of C-P is same as SPDC with $m = n$

$$\widetilde{\mathcal{O}}\left(1 + R/\sqrt{\lambda\gamma}\right) = \widetilde{\mathcal{O}}\left(1 + \sqrt{\kappa}\right), \qquad \text{where } \kappa = R^2/(\lambda\gamma)$$

Non-smooth or non-strongly convex functions

- assumptions:
  - each ϕ_i and g are convex and Lipschitz continuous
  - $f(x, y)$ has a saddle point

- consider perturbed saddle point function

$$f_\delta(x, y) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i \langle a_i, x \rangle - \left(\phi_i^*(y_i) + \frac{\delta y_i^2}{2} \right) \right) + g(x) + \frac{\delta}{2} \|x\|_2^2$$

  - treat $\phi_i^* + \frac{\delta}{2}(\cdot)^2$ as ϕ_i^* and $g + \frac{\delta}{2}\|\cdot\|_2^2$ as g, all become δ-strongly convex
  - adding strongly convex perturbation on ϕ_i^* equivalent to smoothing ϕ_i, which becomes $(1/\delta)$-smooth

- apply SPDC to $f_\delta(x, y)$ with $\delta = O(\epsilon)$

Complexities under different assumptions

SPDC (stochastic primal-dual coordinate method)

ϕ_i	g	iteration complexity $\widetilde{\mathcal{O}}(\cdot)$
$(1/\gamma)$-smooth	λ-strongly convex	$n/m + \sqrt{(n/m)/(\lambda\gamma)}$
$(1/\gamma)$-smooth	non-strongly convex	$n/m + \sqrt{(n/m)/(\epsilon\gamma)}$
non-smooth	λ-strongly convex	$n/m + \sqrt{(n/m)/(\epsilon\lambda)}$
non-smooth	non-strongly convex	$n/m + \sqrt{n/m}/\epsilon$

- for last three cases, solve perturbed problem with $\delta = O(\epsilon)$
- last row: faster than SGD complexity $O(1/\epsilon^2)$ if $\epsilon < \sqrt{m/n}$

SPDC with non-uniform sampling

- potential problem
  - SPDC complexity depends on problem-specific constant

$$R = \max_{i=1,\ldots,n} \|a_i\|_2$$

  - may perform badly on unnormalized data

- solutions
  - can normalize to have $\|a_i\|_2 = 1$ for $i = 1, \ldots, n$
  - or use non-uniform sampling in choosing dual coordinates

$$p_k = (1-\alpha)\frac{1}{2n} + \alpha \frac{\|a_k\|_2}{\sum_{i=1}^n \|a_i\|_2}, \qquad k = 1, \ldots, n$$

with $0 < \alpha < 1$
(mixture of uniform and weighted sampling proportional to $L_i^{1/2}$)

Algorithm 3: SPDC with weighted sampling

inputs: parameters $\tau, \sigma, \theta \in \mathbb{R}_+$, and initial points $x^{(0)}$ and $y^{(0)}$

Initialize: $\overline{x}^{(0)} = x^{(0)}$, $u^{(0)} = (1/n)\sum_{i=1}^{n} y_i^{(0)} a_i$

for $t = 0, 1, 2, \ldots, T-1$

pick $k \in \{1, 2, \ldots, n\}$ with probability p_k, and update

$$y_i^{(t+1)} = \begin{cases} \arg\max_{\beta \in \mathbb{R}} \left\{ \beta \langle a_i, \overline{x}^{(t)} \rangle - \phi_i^*(\beta) - \frac{p_i n}{2\sigma}(\beta - y_i^{(t)})^2 \right\} & i = k, \\ y_i^{(t)} & i \neq k, \end{cases}$$

$$u^{(t+1)} = u^{(t)} + \frac{1}{n}(y_k^{(t+1)} - y_k^{(t)}) a_k$$

$$x^{(t+1)} = \arg\min_{x \in \mathbb{R}^d} \left\{ g(x) + \left\langle u^{(t)} + \frac{1}{p_k}(u^{(t+1)} - u^{(t)}) a_k,\, x \right\rangle + \frac{\|x - x^{(t)}\|_2^2}{2\tau} \right\}$$

$$\overline{x}^{(t+1)} = x^{(t+1)} + \theta(x^{(t+1)} - x^{(t)})$$

output: $x^{(T)}$ and $y^{(T)}$

Complexity analysis with non-uniform sampling

- **theorem:** if we choose

$$\tau = \frac{\alpha}{2\bar{R}}\sqrt{\frac{\gamma}{n\lambda}}, \qquad \sigma = \frac{\alpha}{2\bar{R}}\sqrt{\frac{n\lambda}{\gamma}}, \qquad \theta = 1 - \left(\frac{n}{1-\alpha} + \frac{\bar{R}}{\alpha}\sqrt{\frac{n}{\lambda\gamma}}\right)^{-1},$$

then $\mathbf{E}[\|x^{(T)} - x^\star\|_2^2] \leq \epsilon$ and $\mathbf{E}[\|y^{(T)} - y^\star\|_2^2] \leq \epsilon$ whenever

$$T \geq \left(\frac{n}{1-\alpha} + \frac{\sqrt{\bar{\kappa} n}}{\alpha}\right)\log\left(\frac{C}{\epsilon}\right) \quad \text{where } \bar{\kappa} = \frac{\left(\sum_{i=1}^n \|a_i\|_2/n\right)^2}{\gamma\lambda}$$

- From $\max_{i\in[n]} \|a_i\|_2$ (maximum) to $\sum_{i=1}^n \|a_i\|_2/n$ (average)

- optimal choice of parameter $\alpha^\star = \frac{1}{1+(n/\bar{\kappa})^{1/4}}$
  - $\alpha^\star = 1/2$ if $\bar{\kappa} = n$
  - larger $\alpha^\star$ (more weighted sampling) for ill-conditioned problems

Efficient implementation

- characteristics of big-data problems
  - both n and d can be very large (up to billions)
  - each feature vector $a_i \in \mathbb{R}^d$ are very sparse (nnz in hundreds)
- naive implementation of SPDC
  - costs $O(d)$ per iteration (for large d, it can be very slow!)
  - cannot scale to large datasets
- efficient implementation of SPDC
  - costs $O(\mathrm{nnz})$ per iteration (same as SGD or SDCA)
  - scales to huge datasets
  - derived for both ℓ_2 and $\ell_1 + \ell_2$ regularizations

Computational experiments

algorithms compared:

- two batch algorithms:
  - AFG: accelerated full gradient method with adaptive linear search (Nesterov 2013)
  - L-BFGS: low-memory BFGS quasi-Newton method
- three randomized incremental algorithms:
  - SDCA: stochastic dual coordinate ascent (Shalev-Shwartz & Zhang 2013)
  - SAG: stochastic averaged gradient (Schmidt, Le Roux, & Bach 2012)
  - A-SDCA: accelerated SDCA (Shalev-Shwartz & Zhang 2014)
- SPDC: stochastic primal-dual coordinate method

Classification with real datasets

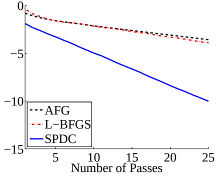
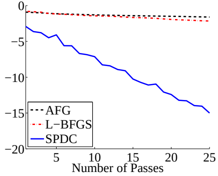
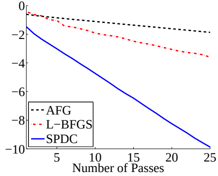
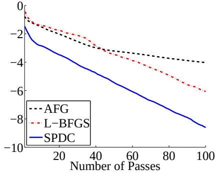
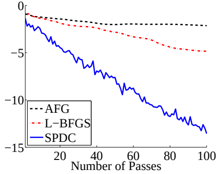
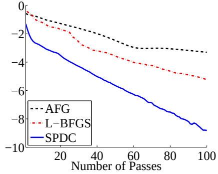
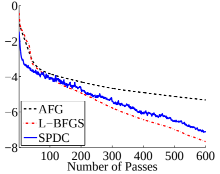
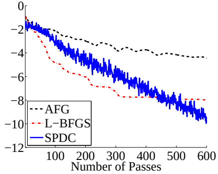
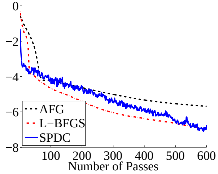
- binary classification with smoothed hinge loss

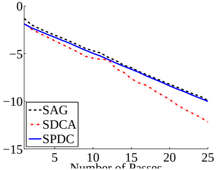
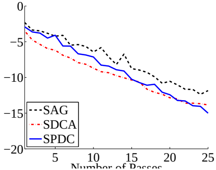
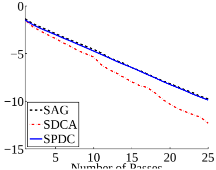
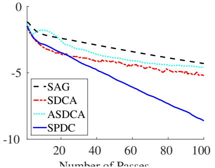
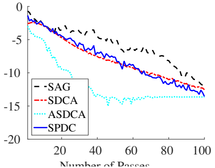
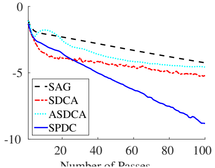
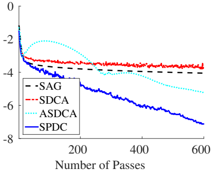
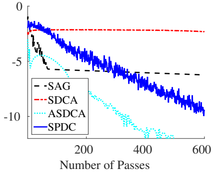
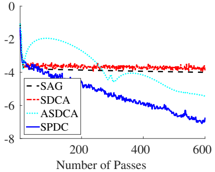
$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad \left\{ P(x) = \frac{1}{n} \sum_{i=1}^{n} \phi_i(a_i^T x) + \frac{\lambda}{2} \|x\|_2^2 \right\}$$

  - smoothed hinge loss $\phi_i(z) = \begin{cases} 0 & \text{if } b_i z \geq 1 \\ \frac{1}{2} - b_i z & \text{if } b_i z \leq 0 \\ \frac{1}{2}(1 - b_i z)^2 & \text{otherwise} \end{cases}$
  - $\phi_i^*(\beta) = b_i \beta + \frac{1}{2}\beta^2$ for $b_i \beta \in [-1, 0]$ and ∞ otherwise

- three real datasets

Dataset name	# samples n	# features d	sparsity
Covtype	581,012	54	22%
RCV1	20,242	47,236	0.16%
News20	19,996	1,355,191	0.04%

λ	RCV1	Covtype	News20
10^{-4}			
10^{-6}			
10^{-8}			

λ	RCV1	Covtype	News20
10^{-4}			
10^{-6}			
10^{-8}			

Summary

- exploiting finite-sum structure of regularized ERM

- two accelerated randomized coordinate update algorithms:
  - APCG: solving the dual ERM problem
  - SPDC: a primal-dual algorithm

  Lan (2015): primal only algorithm + lower complexity bound

- weighted sampling works better for unnormalized data
  (weighted coordinate sampling based on $L_i^{1/2}$)

- superior performance in experiments
  - for (relatively) small κ: much better than batch methods
  - for large $\kappa > n$: much better than SDCA, SAG and SVRG