Proximal Stochastic Dual Coordinate Ascent

Shai Shalev-Shwartz and Tong Zhang

Statistics Department
Rutgers University

Motivation: regularized loss minimization

Assume we want to solve the Lasso problem:

$$\min_w \left[\frac{1}{n} \sum_{i=1}^{n} (w^\top x_i - y_i)^2 + \lambda \|w\|_1 \right]$$

Motivation: regularized loss minimization

Assume we want to solve the Lasso problem:

$$\min_w \left[\frac{1}{n} \sum_{i=1}^{n} (w^\top x_i - y_i)^2 + \lambda \|w\|_1 \right]$$

or the ridge regression problem:

$$\min_w \left[\underbrace{\frac{1}{n} \sum_{i=1}^{n} (w^\top x_i - y_i)^2}_{\text{loss}} + \underbrace{\frac{\lambda}{2} \|w\|_2^2}_{\text{regularization}} \right]$$

Our goal: solve regularized loss minimization problems as fast as we can.

Motivation: regularized loss minimization

Assume we want to solve the Lasso problem:

$$\min_w \left[\frac{1}{n} \sum_{i=1}^{n} (w^\top x_i - y_i)^2 + \lambda \|w\|_1 \right]$$

or the ridge regression problem:

$$\min_w \left[\underbrace{\frac{1}{n} \sum_{i=1}^{n} (w^\top x_i - y_i)^2}_{\text{loss}} + \underbrace{\frac{\lambda}{2} \|w\|_2^2}_{\text{regularization}} \right]$$

Our goal: solve regularized loss minimization problems as fast as we can.

- Problem is deterministic optimization
- But a good solution leads to stochastic algorithm called *proximal Stochastic Dual Coordinate Ascent* (Prox-SDCA).
- We show: fast convergence of SDCA for many regularized loss minimization problems in machine learning.

Outline

- Loss Minimization with L_2 Regularization
  - dual formulation
  - Dual Coordinate Ascent (DCA) and Stochastic Gradient Descent
  - fast convergence Properties of SDCA
  - the importance of randomization
- General regularization
  - duality
  - Prox-SDCA algorithm
  - fast convergence and comparison to other methods
- Highlevel proof ideas

Loss Minimization with L_2 Regularization

$$\min_w P(w) := \left[\frac{1}{n}\sum_{i=1}^{n}\phi_i(w^\top x_i) + \frac{\lambda}{2}\|w\|^2\right].$$

Loss Minimization with L_2 Regularization

$$\min_w P(w) := \left[\frac{1}{n}\sum_{i=1}^{n}\phi_i(w^\top x_i) + \frac{\lambda}{2}\|w\|^2\right].$$

Examples:

	$\phi_i(z)$	Lipschitz	smooth
SVM	$\max\{0, 1 - y_i z\}$	✓	✗
Logistic regression	$\log(1 + \exp(-y_i z))$	✓	✓
Abs-loss regression	$\lvert z - y_i\rvert$	✓	✗
Square-loss regression	$(z - y_i)^2$	✗	✓

Dual Formulation

Primal problem:

$$w_* = \arg\min_w P(w) := \left[\frac{1}{n}\sum_{i=1}^{n}\phi_i(w^\top x_i) + \frac{\lambda}{2}\|w\|^2\right]$$

Dual problem:

$$\alpha_* = \max_{\alpha\in\mathbb{R}^n} D(\alpha) := \left[\frac{1}{n}\sum_{i=1}^{n} -\phi_i^*(-\alpha_i) - \frac{\lambda}{2}\left\|\frac{1}{\lambda n}\sum_{i=1}^{n}\alpha_i x_i\right\|^2\right],$$

and the convex conjugate (dual) is defined as:

$$\phi_i^*(a) = \sup_z(az - \phi_i(z)).$$

Relationship of Primal and Dual Solutions

Weak duality: $P(w) \geq D(\alpha)$ for all w and α
Strong duality: $P(w_*) = D(\alpha_*)$ with the relationship

$$w_* = \frac{1}{\lambda n} \sum_{i=1}^{n} \alpha_{*,i} \cdot x_i, \quad \alpha_{*\,i} = -\phi_i'(w_*^\top x_i).$$

Relationship of Primal and Dual Solutions

Weak duality: $P(w) \geq D(\alpha)$ for all w and α
Strong duality: $P(w_*) = D(\alpha_*)$ with the relationship

$$w_* = \frac{1}{\lambda n} \sum_{i=1}^{n} \alpha_{*,i} \cdot x_i, \quad \alpha_{*\,i} = -\phi_i'(w_*^\top x_i).$$

Duality gap: for any w and α:

$$\underbrace{P(w) - D(\alpha)}_{\text{duality gap}} \geq \underbrace{P(w) - P(w_*)}_{\text{primal sub-optimality}} .$$

Example: Linear Support Vector Machine

- Primal formulation:

$$P(w) = \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - w^\top x_i y_i) + \frac{\lambda}{2}\|w\|_2^2$$

- Dual formulation:

$$D(\alpha) = \frac{1}{n}\sum_{i=1}^{n} \alpha_i y_i - \frac{1}{2\lambda n^2}\left\|\sum_{i=1}^{n} \alpha_i x_i y_i\right\|_2^2, \quad \alpha_i y_i \in [0, 1].$$

- Relationship:

$$w_* = \frac{1}{\lambda n}\sum_{i=1}^{n} \alpha_{*,i} x_i$$

Dual Coordinate Ascent (DCA)

Solve the dual problem using coordinate ascent

$$\max_{\alpha \in \mathbb{R}^n} D(\alpha),$$

and keep the corresponding primal solution using the relationship

$$w = \frac{1}{\lambda n} \sum_{i=1}^{n} \alpha_i x_i.$$

- DCA: At each iteration, optimize $D(\alpha)$ w.r.t. a single coordinate, while the rest of the coordinates are kept in tact.
- Stochastic Dual Coordinate Ascent (SDCA): Choose the updated coordinate uniformly at random

Dual Coordinate Ascent (DCA)

Solve the dual problem using coordinate ascent

$$\max_{\alpha \in \mathbb{R}^n} D(\alpha),$$

and keep the corresponding primal solution using the relationship

$$w = \frac{1}{\lambda n} \sum_{i=1}^{n} \alpha_i x_i.$$

- DCA: At each iteration, optimize $D(\alpha)$ w.r.t. a single coordinate, while the rest of the coordinates are kept in tact.
- Stochastic Dual Coordinate Ascent (SDCA): Choose the updated coordinate uniformly at random

SMO (John Platt), Liblinear (Hsieh et al) etc implemented DCA.

SDCA vs. SGD — update rule

Stochastic Gradient Descent (SGD) update rule:

$$w^{(t+1)} = \left(1 - \tfrac{1}{t}\right) w^{(t)} - \frac{\phi_i'(w^{(t)\top} x_i)}{\lambda t} x_i$$

SDCA update rule:

$$1. \quad \Delta_i = \operatorname*{argmax}_{\Delta \in \mathbb{R}} D(\alpha^{(t)} + \Delta_i e_i)$$

$$2. \quad w^{(t+1)} = w^{(t)} + \frac{\Delta_i}{\lambda n} x_i$$

- Rather similar update rules.
- SDCA has several advantages:
  - Stopping criterion: duality gap smaller than a value
  - No need to tune learning rate

SDCA vs. SGD — update rule — Example

SVM with the hinge loss: $\phi_i(w) = \max\{0, 1 - y_i w^\top x_i\}$

SGD update rule:

$$w^{(t+1)} = \left(1 - \tfrac{1}{t}\right) w^{(t)} - \frac{\mathbf{1}[y_i\, x_i^\top w^{(t)} < 1]}{\lambda\, t}\, x_i$$

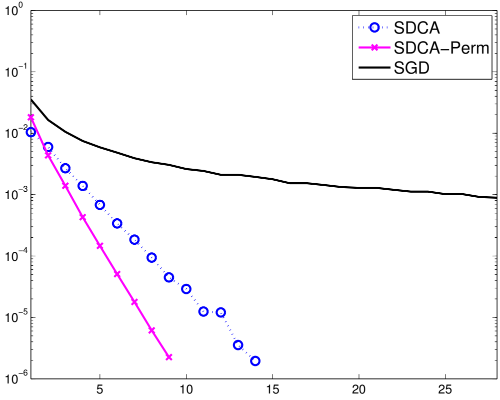
SDCA update rule:

$$1.\ \Delta_i = y_i \max\left(0, \min\left(1, \frac{1 - y_i\, x_i^\top w^{(t-1)}}{\|x_i\|_2^2/(\lambda n)} + y_i\, \alpha_i^{(t-1)}\right)\right) - \alpha_i^{(t-1)}$$

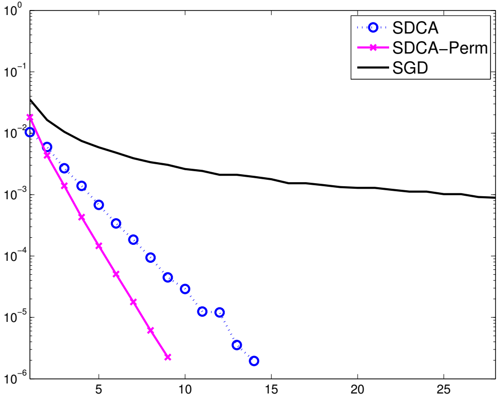
1. $\alpha^{(t+1)} = \alpha^{(t)} + \Delta_i\, e_i$
2. $w^{(t+1)} = w^{(t)} + \dfrac{\Delta_i}{\lambda\, n}\, x_i$

SDCA vs. SGD — experimental observations

- On CCAT dataset, $\lambda = 10^{-6}$, smoothed loss



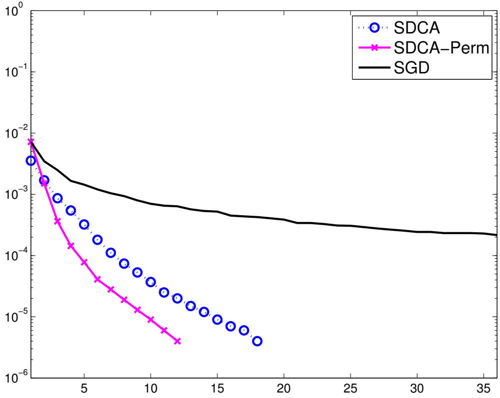
- On CCAT dataset, $\lambda = 10^{-6}$, smoothed loss



The convergence of SDCA is shockingly fast! How to explain this?

SDCA vs. SGD — experimental observations

- On CCAT dataset, $\lambda = 10^{-5}$, hinge-loss



How to understand the convergence behavior?

SDCA vs. SGD — Current analysis is unsatisfactory

How many iterations are required to guarantee $P(w^{(t)}) \leq P(w^*) + \epsilon$?

- For SGD: $\tilde{O}\left(\frac{1}{\lambda\epsilon}\right)$
- For SDCA:
  - Hsieh et al. (ICML 2008), following Luo and Tseng (1992): $O\left(\frac{1}{\nu}\log(1/\epsilon)\right)$, but, ν can be arbitrarily small
  - Shalev-Schwartz and Tewari (2009), Nesterov (2010):
    - $O(n/\epsilon)$ for general n-dimensional coordinate ascent
    - Can apply it to the dual problem
    - Resulting rate is slower than SGD
    - And, the analysis does not hold for logistic regression (it requires smooth dual)
  - Analysis is for dual sub-optimality

SDCA vs. SGD — Current analysis is unsatisfactory

How many iterations are required to guarantee $P(w^{(t)}) \leq P(w^*) + \epsilon$?

- For SGD: $\tilde{O}\left(\frac{1}{\lambda \epsilon}\right)$
- For SDCA:
  - Hsieh et al. (ICML 2008), following Luo and Tseng (1992): $O\left(\frac{1}{\nu}\log(1/\epsilon)\right)$, but, ν can be arbitrarily small
  - Shalev-Schwartz and Tewari (2009), Nesterov (2010):
    - $O(n/\epsilon)$ for general n-dimensional coordinate ascent
    - Can apply it to the dual problem
    - Resulting rate is slower than SGD
    - And, the analysis does not hold for logistic regression (it requires smooth dual)
  - Analysis is for dual sub-optimality
  - What we need: duality gap and primal sub-optimality

Dual vs. Primal sub-optimality

Good dual sub-optimality does not imply good primal sub-optimality!

Dual vs. Primal sub-optimality

Good dual sub-optimality does not imply good primal sub-optimality!

- Take data which is linearly separable using a vector w_0
- Set $\lambda = 2\epsilon/\|w_0\|^2$ and use the hinge-loss
- $P(w^*) \leq P(w_0) = \epsilon$
- Take dual solution 0 and the corresponding primal solution $w(0) = 0$
- $D(0) = 0 \;\Rightarrow\; D(\alpha^*) - D(0) = P(w^*) - D(0) \leq \epsilon$
- $P(w(0)) - P(w^*) = 1 - P(w^*) \geq 1 - \epsilon$

Conclusion: it is important to study the convergence of duality gap.

Our Results: to achieve ϵ accuracy

- For $(1/\gamma)$-smooth loss:

$$\tilde{O}\left(\left(n + \frac{1}{\gamma\lambda}\right)\log\frac{1}{\epsilon}\right)$$

- For L-Lipschitz loss:

$$\tilde{O}\left(n + \frac{L^2}{\lambda\,\epsilon}\right)$$

- For "almost smooth" loss functions (e.g. the hinge-loss):

$$\tilde{O}\left(n + \frac{L}{\lambda\,(\epsilon/L)^{1/(1+\nu)}}\right)$$

where $\nu > 0$ is a data dependent quantity

Compare to Batch Gradient Descent Algorithm

Number of examples needed needed to achieve ϵ accuracy:

- $(1/\gamma)$-smooth loss:
  - Batch GD: $\tilde{O}(n \cdot 1/(\gamma\lambda)\log(1/\epsilon))$
  - SDCA: $\tilde{O}(n + 1/(\gamma\lambda)\log(1/\epsilon))$
- L-Lipschitz loss:
  - Batch GD: $\tilde{O}(n \cdot L^2/(\lambda\epsilon))$
  - SDCA: $\tilde{O}(n + L^2/(\lambda\epsilon))$

Compare to Batch Gradient Descent Algorithm

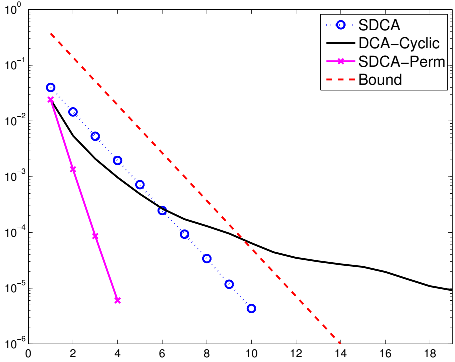
Number of examples needed needed to achieve ϵ accuracy:

- $(1/\gamma)$-smooth loss:
  - Batch GD: $\tilde{O}(n \cdot 1/(\gamma\lambda) \log(1/\epsilon))$
  - SDCA: $\tilde{O}(n + 1/(\gamma\lambda) \log(1/\epsilon))$
- L-Lipschitz loss:
  - Batch GD: $\tilde{O}(n \cdot L^2/(\lambda\epsilon))$
  - SDCA: $\tilde{O}(n + L^2/(\lambda\epsilon))$

The gain of SDCA over batch algorithm is significant when n is large.

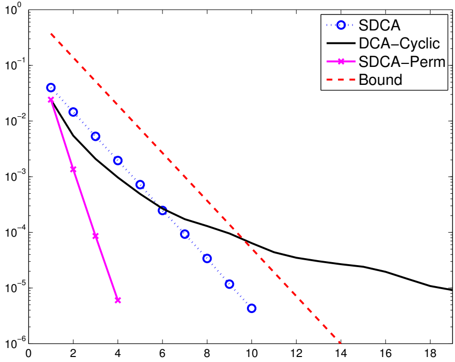
SDCA vs. DCA — Randomization is Crucial!

- On CCAT dataset, $\lambda = 10^{-4}$, smoothed hinge-loss



SDCA vs. DCA — Randomization is Crucial!

- On CCAT dataset, $\lambda = 10^{-4}$, smoothed hinge-loss

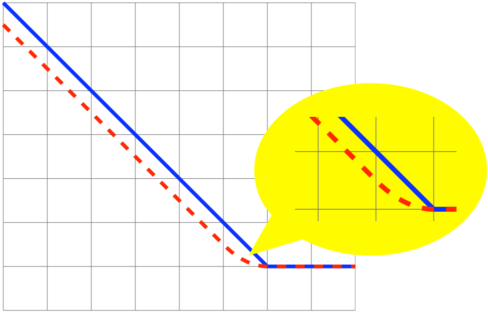


Randomization is crucial!

- In particular, the bound of Luo and Tseng holds for cyclic order, hence must be inferior to our bound

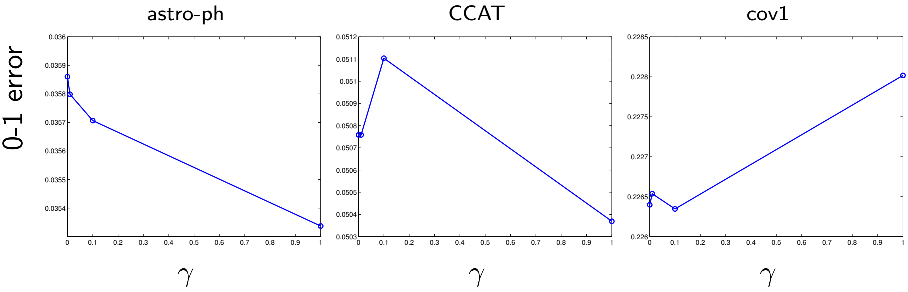
Smoothing the hinge-loss

$$\phi(x) = \begin{cases} 0 & x > 1 \\ 1 - x - \gamma/2 & x < 1 - \gamma \\ \frac{1}{2\gamma}(1-x)^2 & \text{o.w.} \end{cases}$$



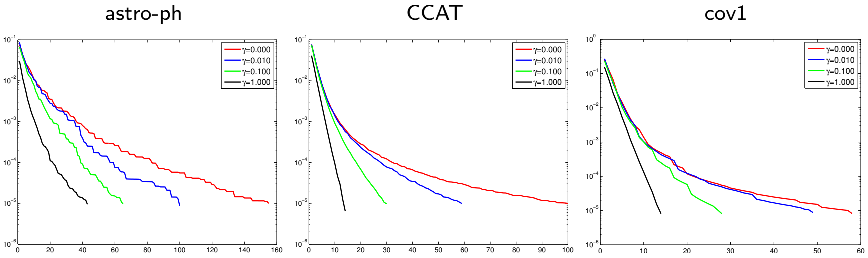
Smoothing the hinge-loss

- Mild effect on 0-1 error



Smoothing the hinge-loss

- Improves training time



- Duality gap as a function of runtime for different smoothing parameters

Additional related work

- Collins et al (2008): For smooth loss, similar bound to ours (for smooth loss) but for a more complicated algorithm (Exponentiated Gradient on dual)
- Lacoste-Julien, Jaggi, Schmidt, Pletscher (preprint on Arxiv):
  - Study Frank-Wolfe algorithm for the dual of structured prediction problems.
  - Boils down to SDCA for the case of binary hinge-loss.
  - Same bound as our bound for the Lipschitz case
- Le Roux, Schmidt, Bach (NIPS 2012): A variant of SGD for smooth loss and finite sample. Also obtain $\log(1/\epsilon)$.

Proximal SDCA for General Regularizer

Want to solve:

$$\min_w P(w) := \left[\frac{1}{n} \sum_{i=1}^{n} \phi_i(X_i^\top w) + \lambda g(w) \right],$$

where X_i are matrices; $g(\cdot)$ is strongly convex.
Examples:

- Multi-class logistic loss

$$\phi_i(X_i^\top w) = \ln \sum_{\ell=1}^{K} \exp(w^\top X_{i,\ell}) - w^\top X_{i,y_i}.$$

- $L_1 - L_2$ regularization

$$g(w) = \frac{1}{2}\|w\|_2^2 + \frac{\sigma}{\lambda}\|w\|_1$$

Dual Formulation

Primal:

$$\min_w P(w) := \left[\frac{1}{n} \sum_{i=1}^{n} \phi_i(X_i^\top w) + \lambda g(w) \right],$$

Dual:

$$\max_\alpha D(\alpha) := \left[\frac{1}{n} \sum_{i=1}^{n} -\phi_i^*(-\alpha_i) - \lambda g^* \left(\frac{1}{\lambda n} \sum_{i=1}^{n} X_i \alpha_i \right) \right]$$

with the relationship

$$w = \nabla g^* \left(\frac{1}{\lambda n} \sum_{i=1}^{n} X_i \alpha_i \right).$$

Prox-SDCA: extension of SDCA for arbitrarily strongly convex $g(w)$.

Prox-SDCA

Dual:

$$\max_{\alpha} D(\alpha) := \left[\frac{1}{n}\sum_{i=1}^{n} -\phi_i^*(-\alpha_i) - \lambda g^*(v)\right], \quad v = \frac{1}{\lambda n}\sum_{i=1}^{n} X_i \alpha_i.$$

Assume $g(w)$ is strongly convex in norm $\|\cdot\|_P$ with dual norm $\|\cdot\|_D$.

Prox-SDCA

Dual:

$$\max_\alpha D(\alpha) := \left[\frac{1}{n}\sum_{i=1}^n -\phi_i^*(-\alpha_i) - \lambda g^*(v)\right], \quad v = \frac{1}{\lambda n}\sum_{i=1}^n X_i\alpha_i .$$

Assume $g(w)$ is strongly convex in norm $\|\cdot\|_P$ with dual norm $\|\cdot\|_D$.
For each α, and the corresponding v and w, define prox-dual

$$\tilde{D}_\alpha(\Delta\alpha) = \left[\frac{1}{n}\sum_{i=1}^n -\phi_i^*(-(\alpha_i + \Delta\alpha_i)) \right.$$

$$\left. -\lambda\left(\underbrace{g^*(v) + \nabla g^*(v)^\top \frac{1}{\lambda n}\sum_{i=1}^n X_i\Delta\alpha_i + \frac{1}{2}\left\|\frac{1}{\lambda n}\sum_{i=1}^n X_i\Delta\alpha_i\right\|_D^2}_{\text{upper bound of } g^*(\cdot)}\right)\right]$$

Prox-SDCA

Dual:

$$\max_{\alpha} D(\alpha) := \left[\frac{1}{n}\sum_{i=1}^{n} -\phi_i^*(-\alpha_i) - \lambda g^*(v)\right], \quad v = \frac{1}{\lambda n}\sum_{i=1}^{n} X_i \alpha_i .$$

Assume $g(w)$ is strongly convex in norm $\|\cdot\|_P$ with dual norm $\|\cdot\|_D$.
For each α, and the corresponding v and w, define prox-dual

$$\tilde{D}_{\alpha}(\Delta\alpha) = \left[\frac{1}{n}\sum_{i=1}^{n} -\phi_i^*(-(\alpha_i + \Delta\alpha_i)) \right.$$
$$\left. -\lambda \left(\underbrace{g^*(v) + \nabla g^*(v)^\top \frac{1}{\lambda n}\sum_{i=1}^{n} X_i \Delta\alpha_i + \frac{1}{2}\left\|\frac{1}{\lambda n}\sum_{i=1}^{n} X_i \Delta\alpha_i\right\|_D^2}_{\text{upper bound of } g^*(\cdot)} \right) \right]$$

Prox-SDCA: randomly pick i and update $\Delta\alpha_i$ by maximizing $\tilde{D}_{\alpha}(\cdot)$.

Example: $L_1 - L_2$ Regularized Logistic Regression

Primal:

$$P(w) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\ln(1+e^{-w^\top X_i Y_i})}_{\phi_i(w)} + \underbrace{\frac{\lambda}{2} w^\top w + \sigma\|w\|_1}_{\lambda g(w)} .$$

Dual: with $\alpha_i Y_i \in [0,1]$

$$D(\alpha) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{-\alpha_i Y_i \ln(\alpha_i Y_i) - (1-\alpha_i Y_i)\ln(1-\alpha_i Y_i)}_{\phi_i^*(-\alpha_i)} - \frac{\lambda}{2}\|\mathrm{trunc}(v, \sigma/\lambda)\|_2^2$$

s.t. $v = \frac{1}{\lambda n}\sum_{i=1}^{n} \alpha_i X_i; \qquad w = \mathrm{trunc}(v, \sigma/\lambda)$

where

$$\mathrm{trunc}(u,\delta)_j = \begin{cases} u_j - \delta & \text{if } u_j > \delta \\ 0 & \text{if } |u_j| \le \delta \\ u_j + \delta & \text{if } u_j < -\delta \end{cases}$$

Proximal-SDCA for L_1-L_2 Regularization

Algorithm:

Keep dual α and $v = (\lambda n)^{-1} \sum_i \alpha_i X_i$

- Randomly pick i
- Find Δ_i by approximately maximizing:

$$-\phi_i^*(\alpha_i + \Delta_i) - \mathrm{trunc}(v, \sigma/\lambda)^\top X_i \, \Delta_i - \frac{1}{2\lambda n} \|X_i\|_2{}^2 \Delta_i^2,$$

where $\phi_i^*(\alpha_i + \Delta) = (\alpha_i + \Delta) Y_i \ln((\alpha_i + \Delta) Y_i) + (1 - (\alpha_i + \Delta) Y_i) \ln(1 - (\alpha_i + \Delta) Y_i)$

- $\alpha = \alpha + \Delta_i \cdot e_i$
- $v = v + (\lambda n)^{-1} \Delta_i \cdot X_i$.

Let $w = \mathrm{trunc}(v, \sigma/\lambda)$.

Proximal-SDCA for L_1-L_2 Regularization

Algorithm:

Keep dual α and $v = (\lambda n)^{-1} \sum_i \alpha_i X_i$

- Randomly pick i
- Find Δ_i by approximately maximizing:

$$-\phi_i^*(\alpha_i + \Delta_i) - \mathrm{trunc}(v, \sigma/\lambda)^\top X_i \, \Delta_i - \frac{1}{2\lambda n} \|X_i\|_2{}^2 \Delta_i^2,$$

  where $\phi_i^*(\alpha_i + \Delta) = (\alpha_i + \Delta) Y_i \ln((\alpha_i + \Delta) Y_i) + (1 - (\alpha_i + \Delta) Y_i) \ln(1 - (\alpha_i + \Delta) Y_i)$

- $\alpha = \alpha + \Delta_i \cdot e_i$
- $v = v + (\lambda n)^{-1} \Delta_i \cdot X_i$.

Let $w = \mathrm{trunc}(v, \sigma/\lambda)$.

Closely related to Lin Xiao (2010): Dual Averaging Method for Regularized Stochastic Learning and Online Optimization

Convergence rate

The same as the non-proximal version of SDCA: number of iterations needed to achieve ϵ accuracy

- For $(1/\gamma)$-smooth loss:

$$\tilde{O}\left(\left(n + \frac{1}{\gamma\lambda}\right)\log\frac{1}{\epsilon}\right)$$

- For L-Lipschitz loss:

$$\tilde{O}\left(n + \frac{L^2}{\lambda\,\epsilon}\right)$$

    - asymptotically faster rate for "almost smooth" loss functions (e.g. the hinge-loss)

Solving L_1 with Smooth Loss

Assume we want to solve L_1 regularization to accuracy ϵ with smooth ϕ_i:

$$\frac{1}{n}\sum_{i=1}^{n}\phi_i(w) + \sigma\|w\|_1.$$

Apply Prox-SDCA with extra term $0.5\lambda\|w\|_2^2$, where $\lambda = O(\epsilon)$:

- number of iterations needed is $\tilde{O}(n + 1/\epsilon)$.

Solving L_1 with Smooth Loss

Assume we want to solve L_1 regularization to accuracy ϵ with smooth ϕ_i:

$$\frac{1}{n}\sum_{i=1}^{n}\phi_i(w) + \sigma\|w\|_1 .$$

Apply Prox-SDCA with extra term $0.5\lambda\|w\|_2^2$, where $\lambda = O(\epsilon)$:

- number of iterations needed is $\tilde{O}(n + 1/\epsilon)$.

Compare to Dual Averaging SGD (Xiao):

- number of iterations needed is $\tilde{O}(1/\epsilon^2)$.

Solving L_1 with Smooth Loss

Assume we want to solve L_1 regularization to accuracy ϵ with smooth ϕ_i:

$$\frac{1}{n}\sum_{i=1}^{n}\phi_i(w) + \sigma\|w\|_1 .$$

Apply Prox-SDCA with extra term $0.5\lambda\|w\|_2^2$, where $\lambda = O(\epsilon)$:

- number of iterations needed is $\tilde{O}(n + 1/\epsilon)$.

Compare to Dual Averaging SGD (Xiao):

- number of iterations needed is $\tilde{O}(1/\epsilon^2)$.

Compare to batch accelerated proximal gradient (Nesterov):

- number of iterations needed is $\tilde{O}(n/\sqrt{\epsilon})$.

Prox-SDCA wins in the statistically interesting regime: $\epsilon > \Omega(1/n^2)$

Analysis of SDCA: Highlevel Idea

- Main lemma: for any t and $s \in [0, 1]$,

$$\underbrace{\mathbb{E}[D(\alpha^{(t)}) - D(\alpha^{(t-1)})]}_{\text{dual suboptimality improvement}} \geq \frac{s}{n} \underbrace{\mathbb{E}[P(w^{(t-1)}) - D(\alpha^{(t-1)})]}_{\text{duality gap}} - \left(\frac{s}{n}\right)^2 \frac{G^{(t)}}{2\lambda}$$

Improvement of dual can be estimated from duality gap

Analysis of SDCA: Highlevel Idea

- Main lemma: for any t and $s \in [0,1]$,

$$\underbrace{\mathbb{E}[D(\alpha^{(t)}) - D(\alpha^{(t-1)})]}_{\text{dual suboptimality improvement}} \geq \frac{s}{n} \underbrace{\mathbb{E}[P(w^{(t-1)}) - D(\alpha^{(t-1)})]}_{\text{duality gap}} - \left(\frac{s}{n}\right)^2 \frac{G^{(t)}}{2\lambda}$$

Improvement of dual can be estimated from duality gap

- $G^{(t)} = O(1)$ for Lipschitz losses:

$$\mathbb{E}[D(\alpha^{(t)}) - D(\alpha^{(t-1)})] \geq \frac{s}{n} \mathbb{E}[P(w^{(t-1)}) - D(\alpha^{(t-1)})] - \frac{A}{\lambda} \left(\frac{s}{n}\right)^2$$

- With appropriate s, $G^{(t)} \leq 0$ for smooth losses

$$\mathbb{E}[D(\alpha^{(t)}) - D(\alpha^{(t-1)})] \geq \frac{s}{n} \mathbb{E}[P(w^{(t-1)}) - D(\alpha^{(t-1)})]$$

Proof Idea: smooth loss

- Main lemma: for any t and $s \in [0,1]$,

$$\mathbb{E}[D(\alpha^{(t)}) - D(\alpha^{(t-1)})] \geq \frac{s}{n}\,\mathbb{E}[P(w^{(t-1)}) - D(\alpha^{(t-1)})]$$

- Bounding dual sub-optimality: the above lemma yields

$$\mathbb{E}[D(\alpha^{(t)}) - D(\alpha^{(t-1)})] \geq \frac{s}{n}\,\mathbb{E}[D(\alpha_*) - D(\alpha^{(t-1)})],$$

  which implies linear convergence of dual sub-optimality

- Bounding duality gap: Summing the inequality for iterations $T_0+1,\ldots,T$ and choosing a random $t \in \{T_0+1,\ldots,T\}$ yields,

$$\mathbb{E}\left[(P(w^{(t-1)}) - D(\alpha^{(t-1)}))\right] \leq \frac{n}{s(T-T_0)}\,\mathbb{E}[D(\alpha^{(T)}) - D(\alpha^{(T_0)})]$$

Summary

- Prox-SDCA algorithm:
  - solves loss minimization problems with regularization such as L_1 or L_2
- Works very well in practice
  - it is important to use SDCA, which is superior to cyclic DCA:
  - one cannot just randomize the order once and apply cyclic DCA

Summary

- Prox-SDCA algorithm:
  - solves loss minimization problems with regularization such as L_1 or L_2
- Works very well in practice
  - it is important to use SDCA, which is superior to cyclic DCA:
  - one cannot just randomize the order once and apply cyclic DCA
- Our analysis shows that SDCA is superior to traditional methods in many interesting scenarios

Summary

- Prox-SDCA algorithm:
  - solves loss minimization problems with regularization such as L_1 or L_2
- Works very well in practice
  - it is important to use SDCA, which is superior to cyclic DCA:
  - one cannot just randomize the order once and apply cyclic DCA
- Our analysis shows that SDCA is superior to traditional methods in many interesting scenarios
- What we learn:
  - goal is to solve a determnistic optimization problem
  - but good solution leads to a truly stochastic algorithm

Summary

- Prox-SDCA algorithm:
  - solves loss minimization problems with regularization such as L_1 or L_2
- Works very well in practice
  - it is important to use SDCA, which is superior to cyclic DCA:
  - one cannot just randomize the order once and apply cyclic DCA
- Our analysis shows that SDCA is superior to traditional methods in many interesting scenarios
- What we learn:
  - goal is to solve a determnistic optimization problem
  - but good solution leads to a truly stochastic algorithm

Final question: is there a determnistic algorithm with similar fast convergence properites?