Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization

Shai Shalev-Shwartz and Tong Zhang

School of CS and Engineering,
The Hebrew University of Jerusalem

"Optimization and Big Data",
Edinburgh 2013

Outline

1. Stochastic Dual Coordinate Ascent (SDCA)
   - Works great in practice
   - Lack of adequate theoretical understanding
   - New theoretical analysis + interesting implications
2. Proximal versions
   - Structured Output Learning
   - General regularizers (e.g. ℓ_1)
3. Acceleration and parallel implementation
   - Interpolating between accelerated gradient and SDCA
   - How to parallelize ?

Regularized Loss Minimization

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

Regularized Loss Minimization

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

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 Coordinate Ascent (DCA)

Primal problem:

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

Dual problem:

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

- 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

SDCA vs. SGD — update rule

Stochastic Gradient Descent (SGD) update rule:

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

SDCA update rule:

$$\begin{aligned}
&1.\ \Delta_i = \operatorname*{argmax}_{\Delta \in \mathbb{R}} D(\alpha^{(t)} + \Delta e_i) \\
&2.\ \alpha_i^{(t+1)} = \alpha_i^{(t)} + \Delta_i \\
&3.\ \ w^{(t+1)} = w^{(t)} + \frac{\Delta_i}{\lambda\, n}\, x_i
\end{aligned}$$

- Often, the update rules are rather similar
- SDCA has several advantages:
    - Stopping criterion
    - 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:

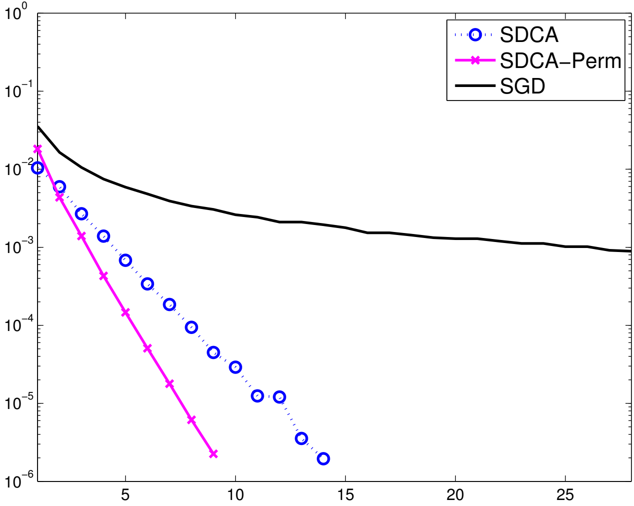
1. $\Delta_i = y_i \max\left(0, \min\left(1, \frac{1 - y_i\, x_i^\top w^{(t-1)}}{\|x_i\|_2^2/(\lambda n)} + y_i\, \alpha_i^{(t-1)}\right)\right) - \alpha_i^{(t-1)}$
2. $\alpha_i^{(t+1)} = \alpha_i^{(t)} + \Delta_i$
3. $w^{(t+1)} = w^{(t)} + \frac{\Delta_i}{\lambda\, n}\, x_i$

Outline

1. Stochastic Dual Coordinate Ascent (SDCA)
   - Works great in practice
   - Lack of adequate theoretical understanding
   - New theoretical analysis + interesting implications
2. Proximal versions
   - Structured Output Learning
   - General regularizers (e.g. ℓ_1)
3. Acceleration and parallel implementation
   - Interpolating between accelerated gradient and SDCA
   - How to parallelize ?

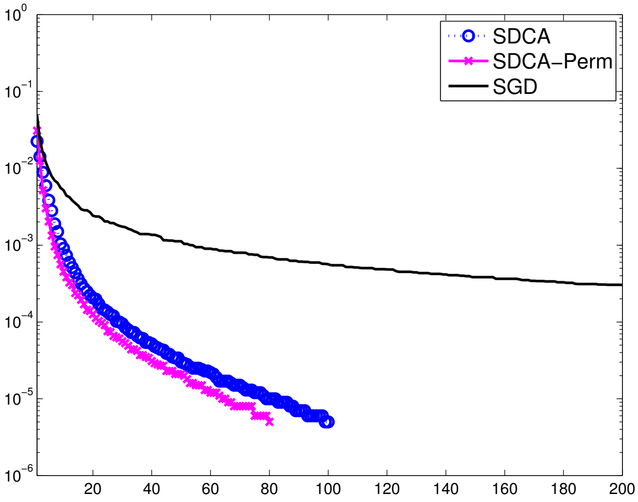
SDCA vs. SGD — experimental observations

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



SDCA vs. SGD — experimental observations

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



Outline

1. Stochastic Dual Coordinate Ascent (SDCA)
   - Works great in practice
   - Lack of adequate theoretical understanding
   - New theoretical analysis + interesting implications
2. Proximal versions
   - Structured Output Learning
   - General regularizers (e.g. ℓ_1)
3. Acceleration and parallel implementation
   - Interpolating between accelerated gradient and SDCA
   - How to parallelize ?

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
  - S 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)
  - Collins et al (2008): For smooth loss, similar bound to ours (for smooth loss) but for a more complicated algorithm (Exponentiated Gradient on dual)
  - Peter Richtarik and Martin Takac: Analysis of randomized coordinate descent with good rates (does not hold for logistic regression)
- Analysis is for dual sub-optimality

Dual vs. 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
- $P(w^*) \leq P(w_0) = \epsilon$
- $D(0) = 0 \;\Rightarrow\; D(\alpha^*) - D(0) = P(w^*) - D(0) \leq \epsilon$
- But, $w(0) = 0$ so $P(w(0)) - P(w^*) = 1 - P(w^*) \geq 1 - \epsilon$
- Conclusion: In the "interesting" regime, ϵ-sub-optimality on the dual can be meaningless w.r.t. the primal !

Very recent related work

- Lacoste-Julien, Jaggi, Schmidt, Pletscher (ICML 2013):
    - Study Frank-Wolfe algorithm for the dual of structured prediction problems.
    - Interestingly, 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.

Outline

1. Stochastic Dual Coordinate Ascent (SDCA)
   - Works great in practice
   - Lack of adequate theoretical understanding
   - New theoretical analysis + interesting implications
2. Proximal versions
   - Structured Output Learning
   - General regularizers (e.g. ℓ_1)
3. Acceleration and parallel implementation
   - Interpolating between accelerated gradient and SDCA
   - How to parallelize ?

Our results

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

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

- For L-Lipschitz loss:

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

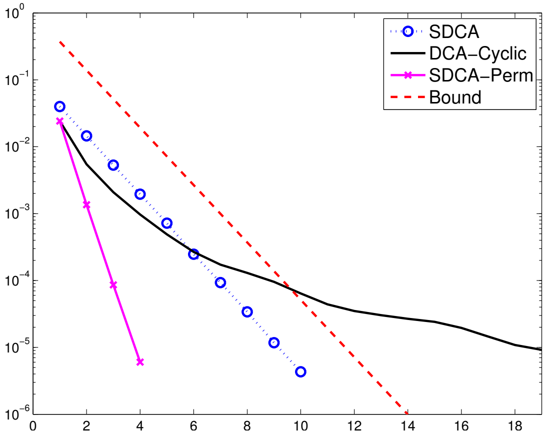
- For "almost smooth" loss functions (e.g. the hinge-loss):

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

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

SDCA vs. DCA — Randomization is crucial

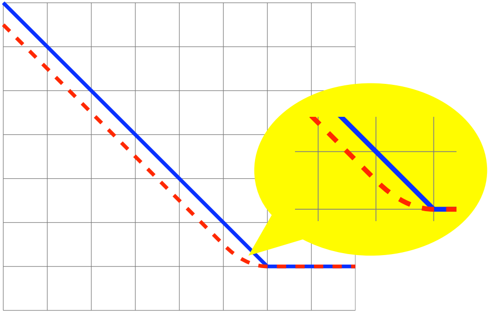
- On CCAT dataset, $\lambda = 10^{-4}$, smoothed hinge-loss



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

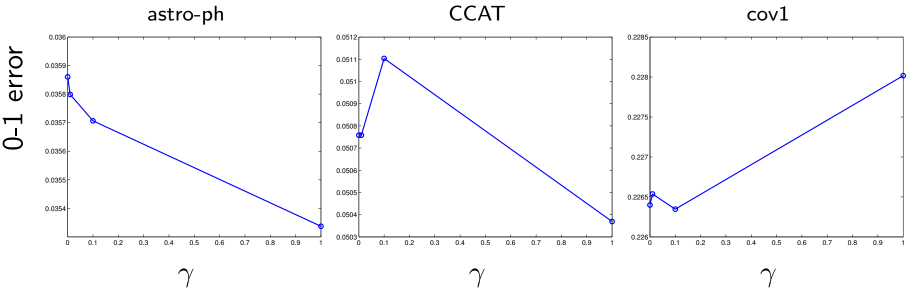
Smoothing the hinge-loss

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



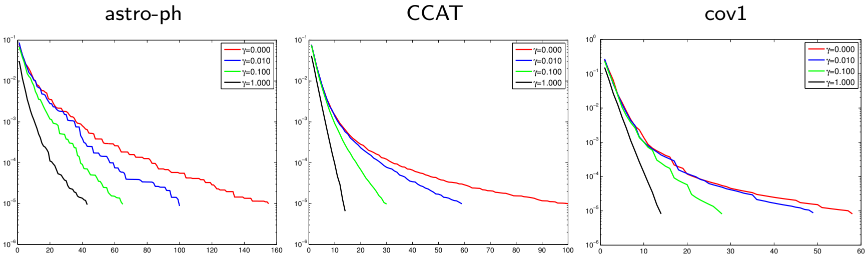
Smoothing the hinge-loss

- Mild effect on 0-1 error



Smoothing the hinge-loss

- Improves training time



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

Proof Idea

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

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

- $G^{(t)} = O(1)$ for Lipschitz losses
- With appropriate s, $G^{(t)} \leq 0$ for smooth losses

Proof Idea

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

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

- Bounding dual sub-optimality:
  Since $P(w^{(t-1)}) \geq D(\alpha^*)$, the above lemma yields a convergence rate for the 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,

$$\mathbb{E}\left[(P(w^{(t-1)}) - D(\alpha^{(t-1)}))\right] \leq \frac{n}{s(T - T_0)}\,\mathbb{E}[D(\alpha^{(T)}) - D(\alpha^{(T_0)})] + \frac{s\,G}{2\lambda n}$$

Outline

1. Stochastic Dual Coordinate Ascent (SDCA)
   - Works great in practice
   - Lack of adequate theoretical understanding
   - New theoretical analysis + interesting implications
2. Proximal versions
   - Structured Output Learning
   - General regularizers (e.g. ℓ_1)
3. Acceleration and parallel implementation
   - Interpolating between accelerated gradient and SDCA
   - How to parallelize ?

Proximal version

Primal problem:

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

Dual problem:

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

Main Results

Assumptions:

- ϕ_i is L-Lipschitz or $1/\gamma$-smooth w.r.t. $\|\cdot\|_P$
- g is strongly convex w.r.t. $\|\cdot\|_{P'}$
- $\|X_i\| \le R$ where $\|X_i\| = \sup_{u \neq 0} \frac{\|X_i u\|_{D'}}{\|u\|_D}$

Results (bound on the number of iterations):

- For smooth losses: $\tilde{O}\left(\left(n + \frac{R^2}{\lambda\gamma}\right)\log(1/\epsilon)\right)$.
- For Lipschitz losses: $O\left(n + \frac{(RL)^2}{\lambda\epsilon}\right)$
- Approximate dual maximization suffices (can obtain closed form approximate solutions while still maintaining same guarantees on the convergence rate)

Structured Output Learning

- The optimization problem:

$$\min_w \left[\frac{\lambda}{2}\|w\|_2^2 + \frac{1}{n}\sum_{i=1}^{n}\left(\max_{y'} \delta(y', y_i) - w^\top \psi(x_i, y_i) + w^\top \psi(x_i, y')\right)\right] .$$

- Let $g(w) = \frac{1}{2}\|w\|_2^2$, let the j'th column of X_i be $\psi(x_i, j)$ and let,

$$\phi_i(v) = \max_j \left(\delta(j, y_i) - v_{y_i} + v_j\right) .$$

- Then g is 1-strongly convex w.r.t. $\|\cdot\|_2$ and ϕ_i is 2-Lipschitz w.r.t. norm $\|\cdot\|_\infty$. Therefore,

$$\|X_i\| = \sup_{u \neq 0} \frac{\|X_i u\|_2}{\|u\|_1} = \sup_{u:\|u\|_1=1} \|X_i u\|_2 = \max_j \|\psi(x_i, j)\|_2 \leq R .$$

- The resulting algorithm is equivalent to Lacoste-Julien et al [2012].

Outline

1. Stochastic Dual Coordinate Ascent (SDCA)
   - Works great in practice
   - Lack of adequate theoretical understanding
   - New theoretical analysis + interesting implications
2. Proximal versions
   - Structured Output Learning
   - General regularizers (e.g. ℓ_1)
3. Acceleration and parallel implementation
   - Interpolating between accelerated gradient and SDCA
   - How to parallelize ?

ℓ_1 regularization, instances of low ℓ_2 norm

- Solve:

$$\min_w \left[\frac{1}{n} \sum_{i=1}^n \phi_i(x_i^\top w) + \sigma \|w\|_1 \right] ,$$

- Assume: $R = \max_i \|x_i\|_2 = O(1)$
- Set $g(w) = \frac{1}{2}\|w\|_2^2 + \frac{\sigma}{\lambda}\|w\|_1$
- Then, $\nabla_i g^*(v) = \mathrm{sign}(v_i) \left[|v_i| - \frac{\sigma}{\lambda} \right]_+$ corresponds to soft-thresholding
- Runtime (smooth case):

Ours	SGD	FISTA	Primal SCD
$nd + \frac{d}{\epsilon \sigma^2}$	$\frac{d}{\epsilon^2 \sigma^2}$	$\frac{nd}{\sqrt{\epsilon}\sigma}$	$\frac{dn}{\epsilon \sigma^2}$

ℓ_1 regularization, instances of low ℓ_∞ norm

- Now assume $\max_i \|x_i\|_\infty = R = O(1)$
- Use $g(w) = \frac{3\log(d)}{2}\|w\|_q^2 + \frac{\sigma}{\lambda}\|w\|_1$, with $q = \frac{\log(d)}{\log(d)-1}$
- $$\nabla_i g^*(v) = \begin{cases} \mathrm{sign}(v_i)\ \left(a\ \left(|v_i| - \frac{\sigma}{\lambda}\right)\right)^{\frac{1}{q-1}} & \text{if } |v_i| > \frac{\sigma}{\lambda} \\ 0 & \text{otherwise} \end{cases}$$
  for some easy to calculate a
- Similar analysis as before

Outline

1. Stochastic Dual Coordinate Ascent (SDCA)
   - Works great in practice
   - Lack of adequate theoretical understanding
   - New theoretical analysis + interesting implications
2. Proximal versions
   - Structured Output Learning
   - General regularizers (e.g. ℓ_1)
3. Acceleration and parallel implementation
   - Interpolating between accelerated gradient and SDCA
   - How to parallelize ?

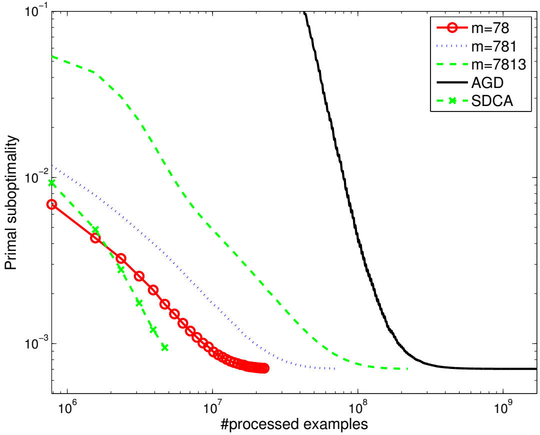
Interpolating between accelerated gradient and SDCA

- SDCA rate $\tilde{O}\left(n + \frac{1}{\lambda}\right)$
- Nesterov's Accelerated (deterministic) Gradient Descent (AGD): $\tilde{O}\left(\frac{1}{\sqrt{\lambda}}\right)$
- ASDCA with mini-batches (of size m):

Algorithm	$\gamma\lambda n = \Theta(1)$	$\gamma\lambda n = \Theta(1/m)$	$\gamma\lambda n = \Theta(m)$
SDCA	n	nm	n
ASDCA	$n/\sqrt{m}$	n	n/m
AGD	$\sqrt{n}$	$\sqrt{nm}$	$\sqrt{n/m}$

Experimental demonstration

- On CCAT dataset, $\lambda = 1/n$, smoothed hinge-loss



How to parallelize ?

- n examples, d features, s computing nodes
- Divide by examples: each node gets n/s examples

How to parallelize ?

- n examples, d features, s computing nodes
- Divide by examples: each node gets n/s examples
- Divide by features: each node gets d/s features of all examples

How to parallelize ?

- n examples, d features, s computing nodes
- Divide by examples: each node gets n/s examples
- Divide by features: each node gets d/s features of all examples
- Runtime per iteration:

Algorithm	partition type	runtime	communication time
SDCA	features	d/s	$s\log^2(s)$
ASDCA	features	dm/s	$ms\log^2(s)$
ASDCA	examples	dm/s	$d\log(s)$
AGD	examples	dn/s	$d\log(s)$

How to parallelize ?

- n examples, d features, s computing nodes
- Assume: $\lambda n = \Theta(1)$
- Total runtime:

Algorithm	partition type	runtime	communication time
SDCA	features	dn/s	$ns\log^2(s)$
ASDCA	features	$dnm^{1/2}/s$	$nm^{1/2}s\log^2(s)$
ASDCA	examples	$dnm^{1/2}/s$	$ndm^{-1/2}\log(s)$
AGD	examples	$dn^{3/2}/s$	$n^{1/2}d\log(s)$

Summary

- SDCA works very well in practice
- So far, theoretical guarantees were unsatisfactory
- Our analysis shows that SDCA is an excellent choice in many scenarios
- Proximal versions
- Accelerated SDCA with mini-batches