Mini-Batch Primal and Dual Methods for Support Vector Machines

Peter Richtárik (Edinburgh)

Coauthors: M. Takáč, A. Bijral and N. Srebro

arXiv:1303.2314 + International Conference on Machine Learning 2013

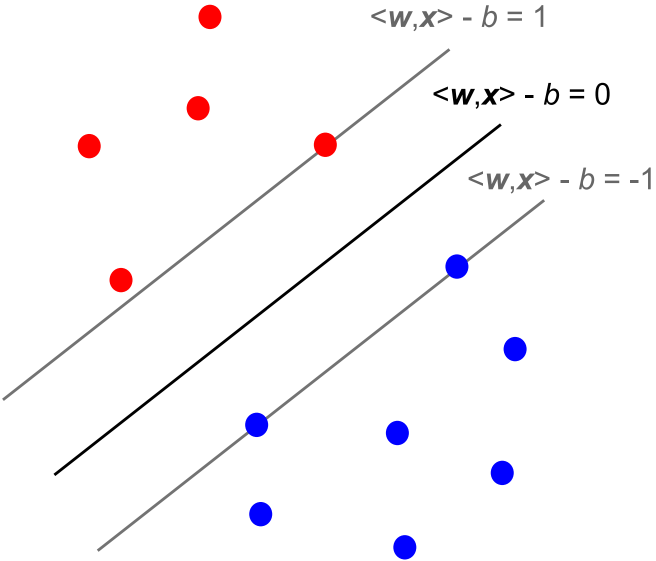
Shai Shalev-Schwartz, Tong Zhang
Stochastic dual coordinate ascent methods for regularized loss minimization
JMLR 2013

Shai Shalev-Schwartz, Yoram Singer, Nathan Srebro
Pegasos: Primal estimated subgradient solver for SVM
ICML 2007

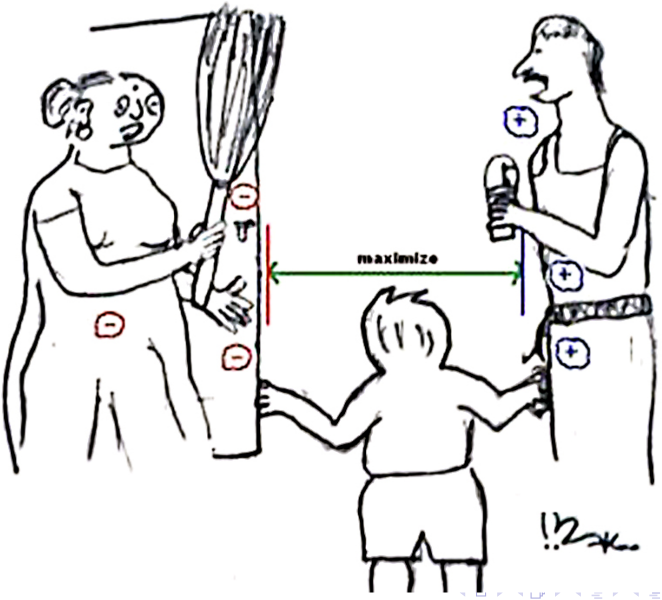
Peter Richtárik, Martin Takáč
Parallel coordinate descent methods for big data optimization
2012 (arXiv:1212.0873)

Martin Takáč, Avleen Bijral, Peter Richtárik, Nathan Srebro
Mini-batch primal and dual methods for SVMs
ICML 2013

Support Vector Machine



Family Support Machine



PART I:

Stochastic Gradient Descent (SGD)

SVM: Primal Problem

Data:

$$\{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \{+1,-1\} \;:\; i \in S \stackrel{\text{def}}{=} \{1,2,\ldots,n\}\}$$

- Examples: $\mathbf{x}_1,\ldots,\mathbf{x}_n$ (assumption: $\max_i \|\mathbf{x}_i\|_2 \leq 1$)
- Labels: $y_i \in \{+1,-1\}$

Optimization formulation of SVM:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \left\{ \mathcal{P}_S(\mathbf{w}) \stackrel{\text{def}}{=} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \hat{L}_S(\mathbf{w}) \right\}, \tag{P}$$

where

- $\hat{L}_A(\mathbf{w}) \stackrel{\text{def}}{=} \frac{1}{|A|}\sum_{i\in A} \ell(y_i\langle \mathbf{w}, \mathbf{x}_i\rangle)$ (average hinge loss on examples in A)
- $\ell(\zeta) \stackrel{\text{def}}{=} \max\{0, 1-\zeta\}$ (hinge loss)

Pegasos (SGD)

Algorithm

1. Choose $\mathbf{w}_1 = 0 \in \mathbb{R}^d$
2. Iterate for $t = 1, 2, \ldots, T$
   2.1 Choose $A_t \subset S = \{1, 2, \ldots, n\}$, $|A_t| = b$, uniformly at random
   2.2 Set stepsize $\eta_t \leftarrow \frac{1}{\lambda t}$
   2.3 Update $\quad \mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \partial \mathcal{P}_{A_t}(\mathbf{w}_t)$

Theorem

For $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}_t$ we have:

$$\mathbf{E}[\mathcal{P}(\bar{\mathbf{w}})] \leq \mathcal{P}(\mathbf{w}^*) + c\log(T) \cdot \frac{1}{\lambda T},$$

where $c = (\sqrt{\lambda} + 1)^2$.

Shai Shalev-Shwartz, Yoram Singer and Nathan Srebro

Pegasos: Primal Estimated sub-GrAdient SOlver for SVM, ICML 2007

Pegasos (SGD)

Algorithm

1. Choose $\mathbf{w}_1 = 0 \in \mathbb{R}^d$
2. Iterate for $t = 1, 2, \dots, T$
   2.1 Choose $A_t \subset S = \{1, 2, \dots, n\}$, $|A_t| = b$, uniformly at random
   2.2 Set stepsize $\eta_t \leftarrow \frac{1}{\lambda t}$
   2.3 Update $\quad \mathbf{w}_{t+1} \leftarrow (1 - \eta_t \lambda)\mathbf{w}_t + \frac{\eta_t}{b} \sum_{i \in A_t \,:\, y_i \langle \mathbf{w}_t, \mathbf{x}_i \rangle < 1} y_i \mathbf{x}_i$

Theorem

For $\bar{\mathbf{w}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{w}_t$ we have:

$$\mathbf{E}[\mathcal{P}(\bar{\mathbf{w}})] \le \mathcal{P}(\mathbf{w}^*) + c \log(T) \cdot \frac{1}{\lambda T},$$

where $c = (\sqrt{\lambda} + 1)^2$.

Shai Shalev-Shwartz, Yoram Singer and Nathan Srebro

Pegasos: Primal Estimated sub-GrAdient SOlver for SVM, ICML 2007

Pegasos (SGD)

Algorithm

1. Choose $\mathbf{w}_1 = 0 \in \mathbb{R}^d$
2. Iterate for $t = 1, 2, \ldots, T$
   2.1 Choose $A_t \subset S = \{1, 2, \ldots, n\}$, $|A_t| = b$, uniformly at random
   2.2 Set stepsize $\eta_t \leftarrow \frac{1}{\lambda t}$
   2.3 Update $\mathbf{w}_{t+1} \leftarrow (1 - \eta_t \lambda)\mathbf{w}_t + \frac{\eta_t}{b} \sum_{i \in A_t \,:\, y_i \langle \mathbf{w}_t, \mathbf{x}_i \rangle < 1} y_i \mathbf{x}_i$

Theorem 1

For $\bar{\mathbf{w}} = \frac{2}{T} \sum_{t=\lfloor T/2 \rfloor + 1}^{T} \mathbf{w}_t$ we have:

$$\mathbf{E}[\mathcal{P}(\bar{\mathbf{w}})] \leq \mathcal{P}(\mathbf{w}^*) + \frac{30\beta_b}{b} \cdot \frac{1}{\lambda T},$$

where $\beta_b = 1 + \frac{(b-1)(n\sigma^2 - 1)}{n-1}$, $\sigma^2 \stackrel{\text{def}}{=} \frac{1}{n}\|\mathbf{Q}\|$ and $\mathbf{Q}_{ij} = \langle y_i \mathbf{x}_i, y_j \mathbf{x}_j \rangle$

Martin Takáč, Avleen Bijral, P. R. and Nathan Srebro

Mini-batch primal and dual methods for SVMs, ICML 2013

Insight into $\frac{\beta_b}{b}$

$$\frac{\beta_b}{b} = \frac{1 + \frac{(b-1)(n\sigma^2 - 1)}{n-1}}{b}$$

Insight into $\frac{\beta_b}{b}$

$$\frac{\beta_b}{b} = \frac{1 + \frac{(b-1)(n\sigma^2-1)}{n-1}}{b}$$

Letting $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, $\mathbf{Z} = [y_1\mathbf{x}_1, \ldots, y_n\mathbf{x}_n]$ and assuming $\|\mathbf{x}_i\| = 1$ for all i, we have

$$\begin{aligned} n\sigma^2 &\stackrel{\text{def}}{=} \|\mathbf{Q}\| = \|\mathbf{Z}\mathbf{Z}^T\| = \|\mathbf{Z}^T\mathbf{Z}\| \\ &= \lambda_{max}(\mathbf{Z}^T\mathbf{Z}) \in [\tfrac{\mathrm{tr}(\mathbf{Z}^T\mathbf{Z})}{n}, \mathrm{tr}(\mathbf{Z}^T\mathbf{Z})] = [\underbrace{\tfrac{\mathrm{tr}(\mathbf{X}^T\mathbf{X})}{n}}_{=1}, \underbrace{\mathrm{tr}(\mathbf{X}^T\mathbf{X})}_{=n}] \end{aligned}$$

Insight into $\frac{\beta_b}{b}$

$$\frac{\beta_b}{b} = \frac{1 + \frac{(b-1)(n\sigma^2-1)}{n-1}}{b}$$

Letting $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, $\mathbf{Z} = [y_1\mathbf{x}_1, \ldots, y_n\mathbf{x}_n]$ and assuming $\|\mathbf{x}_i\| = 1$ for all i, we have

$$\begin{aligned} n\sigma^2 &\stackrel{\text{def}}{=} \|\mathbf{Q}\| = \|\mathbf{Z}\mathbf{Z}^T\| = \|\mathbf{Z}^T\mathbf{Z}\| \\ &= \lambda_{max}(\mathbf{Z}^T\mathbf{Z}) \in [\tfrac{\mathrm{tr}(\mathbf{Z}^T\mathbf{Z})}{n}, \mathrm{tr}(\mathbf{Z}^T\mathbf{Z})] = [\underbrace{\tfrac{\mathrm{tr}(\mathbf{X}^T\mathbf{X})}{n}}_{=1}, \underbrace{\mathrm{tr}(\mathbf{X}^T\mathbf{X})}_{=n}] \end{aligned}$$

- $n\sigma^2 = n \Rightarrow \frac{\beta_b}{b} = 1$
  (no parallelization speedup; mini-batching does not help)
- $n\sigma^2 = 1 \Rightarrow \frac{\beta_b}{b} = \frac{1}{b}$
  (speedup equal to batch size!)

Insight into $\frac{\beta_b}{b}$

$$\frac{\beta_b}{b} = \frac{1 + \frac{(b-1)(n\sigma^2-1)}{n-1}}{b}$$

Letting $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, $\mathbf{Z} = [y_1\mathbf{x}_1, \ldots, y_n\mathbf{x}_n]$ and assuming $\|\mathbf{x}_i\| = 1$ for all i, we have

$$n\sigma^2 \overset{\text{def}}{=} \|\mathbf{Q}\| = \|\mathbf{Z}\mathbf{Z}^T\| = \|\mathbf{Z}^T\mathbf{Z}\|$$
$$= \lambda_{max}(\mathbf{Z}^T\mathbf{Z}) \in [\tfrac{\mathrm{tr}(\mathbf{Z}^T\mathbf{Z})}{n}, \mathrm{tr}(\mathbf{Z}^T\mathbf{Z})] = [\underbrace{\tfrac{\mathrm{tr}(\mathbf{X}^T\mathbf{X})}{n}}_{=1}, \underbrace{\mathrm{tr}(\mathbf{X}^T\mathbf{X})}_{=n}]$$

- $n\sigma^2 = n \Rightarrow \frac{\beta_b}{b} = 1$
  (no parallelization speedup; mini-batching does not help)
- $n\sigma^2 = 1 \Rightarrow \frac{\beta_b}{b} = \frac{1}{b}$
  (speedup equal to batch size!)

Similar expression appears in

P. R. and Martin Takáč

Parallel coordinate descent methods for big data optimization, 2012

with $n\sigma^2$ replaced by ω (degree of partial separability of the loss function)

Computing β_b: SVM Datasets

To run mini-batched SGD (and SDCA), we need to compute β_b:

$$\beta_b = 1 + \frac{(b-1)(n\sigma^2 - 1)}{n-1}, \quad \text{where} \quad n\sigma^2 = \lambda_{\max}(\mathbf{Z}^T\mathbf{Z})$$

Two options:

- Compute the largest eigenvalue (e.g., power method)
- Replace $n\sigma^2$ by an upper bound: degree* of partial separability ω
  General SDCA methods based on ω described here:

P. R. and Martin Takáč

Parallel coordinate descent methods for big data optimization, 2012

Dataset	# Examples (d)	# Features (n)	$\|A\|_0$	$n\sigma^2 \in [1, n]$	ω
a1a	1,605	123	22,249	13.879	14
a9a	32,561	123	451,592	13.885	14
rcv1	20,242	47,236	1,498,952	105.570	980
real-sim	72,309	20,958	3,709,191	44.253	3,484
news20	19,996	1,355,191	9,097,958	9,674.184	16,423
url	2,396,130	3,231,961	277,058,644	114.956	414
webspam	350,000	16,609,143	29,796,333	50.436	127
kdda2010	8,407,752	20,216,830	305,613,510	47.459	85
kddb2010	19,264,097	29,890,095	566,345,888	41.467	75

Where does β_b Come from?

Lemma 1

Consider any symmetric $\mathbf{Q} \in \mathbb{R}^{n\times n}$, random subset $A \subset \{1, 2, \ldots, n\}$ with $|A| = b$ and $\mathbf{v} \in \mathbb{R}^n$. Then

$$\mathbf{E}[\mathbf{v}_{[A]}^T \mathbf{Q} \mathbf{v}_{[A]}] = \frac{b}{n}\left[\left(1 - \frac{b-1}{n-1}\right)\sum_{i=1}^{n} \mathbf{Q}_{ii}\mathbf{v}_i^2 + \frac{b-1}{n-1}\mathbf{v}^T\mathbf{Q}\mathbf{v}\right].$$

Moreover, if $\mathbf{Q}_{ii} \leq 1$ for all i, then we get the following ESO (Expected Separable Overapproximation):

$$\mathbf{E}[\mathbf{v}_{[A]}^T \mathbf{Q} \mathbf{v}_{[A]}] \leq \frac{b}{n}\beta_b \|\mathbf{v}\|^2.$$

Where does β_b Come from?

Lemma 1

Consider any symmetric $\mathbf{Q} \in \mathbb{R}^{n\times n}$, random subset $A \subset \{1,2,\ldots,n\}$ with $|A| = b$ and $\mathbf{v} \in \mathbb{R}^n$. Then

$$\mathbf{E}[\mathbf{v}_{[A]}^T \mathbf{Q} \mathbf{v}_{[A]}] = \frac{b}{n}\left[\left(1 - \frac{b-1}{n-1}\right)\sum_{i=1}^{n} \mathbf{Q}_{ii}\mathbf{v}_i^2 + \frac{b-1}{n-1}\mathbf{v}^T\mathbf{Q}\mathbf{v}\right].$$

Moreover, if $\mathbf{Q}_{ii} \leq 1$ for all i, then we get the following ESO (Expected Separable Overapproximation):

$$\mathbf{E}[\mathbf{v}_{[A]}^T \mathbf{Q} \mathbf{v}_{[A]}] \leq \frac{b}{n}\beta_b \|\mathbf{v}\|^2.$$

Remark: ESO inequalities are systematically developed in

P. R. and Martin Takáč

Parallel coordinate descent methods for big data optimization, 2012

Insight into the Analysis

Classical Pegasos Analysis

Uses the inequality:

$$\|\nabla \hat{L}_{A_t}(\mathbf{w})\|^2 \leq 1$$

which holds for any $A_t \subset S = \{1, 2, \ldots, n\}$

Insight into the Analysis

Classical Pegasos Analysis

Uses the inequality:

$$\|\nabla \hat{L}_{A_t}(\mathbf{w})\|^2 \leq 1$$

which holds for any $A_t \subset S = \{1, 2, \ldots, n\}$

New Analysis

Uses the inequality:

$$\mathbf{E}\|\nabla \hat{L}_{A_t}(\mathbf{w})\|^2 \leq \frac{\beta_b}{b}$$

which holds for A_t, $|A_t| = b$, chosen uniformly at random (established by previous lemma)

PART II:

Stochastic Dual Coordinate Ascent (SDCA)

Stochastic Dual Coordinate Ascent (SDCA)

Problem:

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n,\ 0\leq\boldsymbol{\alpha}_i\leq 1} \left\{ \mathcal{D}(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\alpha}_i - \frac{1}{2\lambda n^2}\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha} \right\} \qquad \text{(D)}$$

Algorithm

1. Choose $\boldsymbol{\alpha}_0 = 0 \in \mathbb{R}^n$
2. For $t = 0, 1, 2, \ldots$ iterate:
   - 2.1 Choose $i \in \{1, \ldots, n\}$, uniformly at random
   - 2.2 Set $\delta^* \leftarrow \arg\max\{\mathcal{D}(\boldsymbol{\alpha}_t + \delta\mathbf{e}_i) \ : \ 0 \leq \boldsymbol{\alpha}_i + \delta \leq 1\}$
   - 2.3 $\boldsymbol{\alpha}_{t+1} \leftarrow \boldsymbol{\alpha}_t + \delta^*\mathbf{e}_i$

Stochastic Dual Coordinate Ascent (SDCA)

Problem:

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n,\ 0\le\boldsymbol{\alpha}_i\le 1} \left\{ \mathcal{D}(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\alpha}_i - \frac{1}{2\lambda n^2}\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha} \right\} \qquad \text{(D)}$$

Algorithm

1. Choose $\boldsymbol{\alpha}_0 = 0 \in \mathbb{R}^n$
2. For $t = 0, 1, 2, \ldots$ iterate:
   2.1 Choose $i \in \{1, \ldots, n\}$, uniformly at random
   2.2 Set $\delta^* \leftarrow \arg\max\{\mathcal{D}(\boldsymbol{\alpha}_t + \delta\mathbf{e}_i)\ :\ 0 \le \boldsymbol{\alpha}_i + \delta \le 1\}$
   2.3 $\boldsymbol{\alpha}_{t+1} \leftarrow \boldsymbol{\alpha}_t + \delta^*\mathbf{e}_i$

First proposed for SVM by

C.-J. Hsieh K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan
A dual coordinate descent method for large-scale linear SVM, ICML 2008

General analysis in

P. R. and M. Takáč
Iteration complexity of randomized block-coordinate descent methods . . . , MAPR 2012
[INFORMS Computing Society Best Student Paper Prize (runner-up), 2012]

"Naive" Mini-Batching / Parallelization

Problem:

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^n,\ 0 \le \boldsymbol{\alpha}_i \le 1} \left\{ \mathcal{D}(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\alpha}_i - \frac{1}{2\lambda n^2} \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} \right\} \qquad \text{(D)}$$

Algorithm

1. Choose $\boldsymbol{\alpha}_0 = 0 \in \mathbb{R}^n$
2. For $t = 0, 1, 2, \ldots$ iterate:
   - 2.1 Choose $A_t \subset \{1, \ldots, n\}$, $|A_t| = b$, uniformly at random
   - 2.2 Set $\boldsymbol{\alpha}_{t+1} \leftarrow \boldsymbol{\alpha}_t$
   - 2.3 For $i \in A_t$ do
     - 2.3.1 Set $\delta^* \leftarrow \arg\max\{\mathcal{D}(\boldsymbol{\alpha} + \delta \mathbf{e}_i) \ : \ 0 \le \boldsymbol{\alpha}_i + \delta \le 1\}$
     - 2.3.2 $\boldsymbol{\alpha}_{t+1} \leftarrow \boldsymbol{\alpha}_{t+1} + \delta^* \mathbf{e}_i$

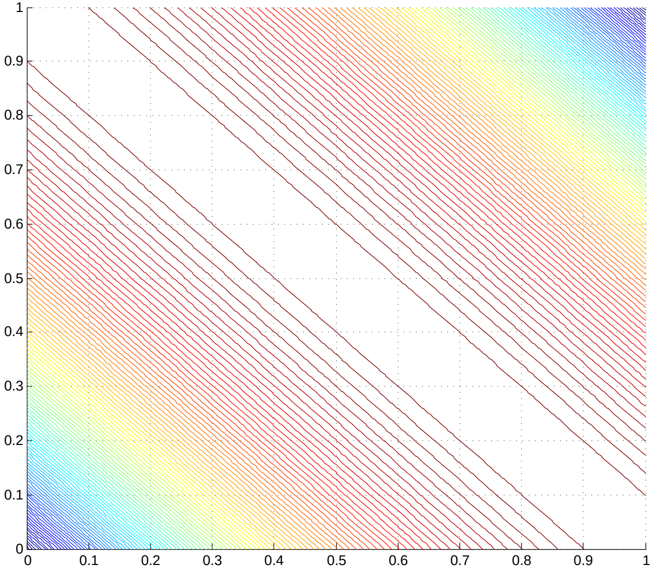
Analyzed in

Joseph K. Bradley, Aapo Kyrola, Danny Bickson, and Carlos Guestrin

Parallel coordinate descent for L1-regularized loss minimization, ICML 2011

- Convergence guaranteed only for "small" b ($\beta_b \le 2$)
- Analysis does not cover SVM dual (D)

Example: Failure of Naive Parallelization



Example: Details

Problem:

$$\mathbf{Q} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad \lambda = \frac{1}{n} = \frac{1}{2}, \quad b = 2.$$

$$\Rightarrow \qquad \mathcal{D}(\boldsymbol{\alpha}) = \frac{1}{2}\mathbf{e}^T\boldsymbol{\alpha} - \frac{1}{4}\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha}$$

The naive approach will produce the sequence:

- $\boldsymbol{\alpha}_0 = (0,0)^T$ with $\mathcal{D}(\boldsymbol{\alpha}_0) = 0$
- $\boldsymbol{\alpha}_1 = (1,1)^T$ with $\mathcal{D}(\boldsymbol{\alpha}_1) = 0$
- $\boldsymbol{\alpha}_2 = (0,0)^T$ with $\mathcal{D}(\boldsymbol{\alpha}_2) = 0$
- $\boldsymbol{\alpha}_3 = (1,1)^T$ with $\mathcal{D}(\boldsymbol{\alpha}_3) = 0$
- $\ldots$

Optimal solution: $\mathcal{D}(\boldsymbol{\alpha}^*) = \mathcal{D}((\frac{1}{2}, \frac{1}{2})^T) = 0.25$

Safe Mini-Batching

Instead of choosing δ "naively" via maximizing the original function

$$\mathcal{D}(\boldsymbol{\alpha}+\boldsymbol{\delta}) := -\frac{(\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\delta} + \boldsymbol{\delta}^T\mathbf{Q}\boldsymbol{\delta})}{2\lambda n^2} + \sum_{i=1}^{n} \frac{\boldsymbol{\alpha}_i + \boldsymbol{\delta}_i}{n},$$

work with its Expected Separable Underapproximation:

$$\mathcal{H}_{\beta_b}(\boldsymbol{\delta},\boldsymbol{\alpha}) := -\frac{(\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\delta} + \beta_b\|\boldsymbol{\delta}\|^2)}{2\lambda n^2} + \sum_{i=1}^{n} \frac{\boldsymbol{\alpha}_i + \boldsymbol{\delta}_i}{n},$$

That is, instead of

$$\delta^* \leftarrow \arg\max\{\mathcal{D}(\boldsymbol{\alpha} + \delta\mathbf{e}_i, \boldsymbol{\alpha}) \ : \ 0 \le \boldsymbol{\alpha}_i + \delta \le 1\}, \quad i \in A_t$$

do

$$\delta^* \leftarrow \arg\max\{\mathcal{H}_{\beta}(\delta\mathbf{e}_i, \boldsymbol{\alpha}) \ : \ 0 \le \boldsymbol{\alpha}_i + \delta \le 1\}, \quad i \in A_t$$

Safe Mini-Batching: General Theory

Developed in

P. R. and Martin Takáč

Parallel coordinate descent methods for big data optimization, 2012

Definition (Expected Separable Overapproximation)

Let

- $f : \mathbb{R}^n \to \mathbb{R}$ be convex and smooth,
- $\hat{S}$ be a random subset of $\{1, 2, \ldots, n\}$ s.t. for all i

$$\mathbf{P}(i \in \hat{S}) = const,$$

- $w \in \mathbb{R}^n_{++}$ define $\|x\|_w \stackrel{\text{def}}{=} (\sum_{i=1}^n w_i x_i^2)^{1/2}$.

Then we say that f admits a (β, w)-ESO w.r.t. $\hat{S}$ if for all $x, h \in \mathbb{R}^n$:

$$\mathbf{E}[f(x + h_{[\hat{S}]})] \leq f(x) + \frac{\mathbf{E}[|\hat{S}|]}{n} \left(\langle \nabla f(x), h \rangle + \frac{\beta}{2} \|h\|_w^2 \right)$$

ESO for SVM Dual

ESO can also be written as

$$\mathbf{E}[f(x+h_{[\hat{S}]})] \leq \left(1-\frac{\mathbf{E}[|\hat{S}|]}{n}\right) f(x) + \frac{\mathbf{E}[|\hat{S}|]}{n} \underbrace{\left(f(x) + \langle \nabla f(x), h\rangle + \frac{\beta}{2}\|h\|_w^2\right)}_{\stackrel{\text{def}}{=}\mathcal{H}_\beta(h;x)}$$

SVM setting: $f(x) = -\mathcal{D}(\boldsymbol{\alpha})$, $h = \boldsymbol{\delta}$, $\hat{S} = A_t$, $|\hat{S}| = b$, $w = (1,\ldots,1)^T$

ESO for SVM Dual

ESO can also be written as

$$\mathbf{E}[f(x+h_{[\hat{S}]})] \leq \left(1-\frac{\mathbf{E}[|\hat{S}|]}{n}\right) f(x) + \frac{\mathbf{E}[|\hat{S}|]}{n} \underbrace{\left(f(x) + \langle \nabla f(x), h\rangle + \frac{\beta}{2}\|h\|_w^2\right)}_{\overset{\text{def}}{=}\mathcal{H}_\beta(h;x)}$$

SVM setting: $f(x) = -\mathcal{D}(\boldsymbol{\alpha})$, $h = \boldsymbol{\delta}$, $\hat{S} = A_t$, $|\hat{S}| = b$, $w = (1,\dots,1)^T$

Lemma 3

For the SVM dual loss we have for all $\boldsymbol{\alpha}, \boldsymbol{\delta} \in \mathbb{R}^n$ the following ESO:

$$\mathbf{E}[\mathcal{D}(\boldsymbol{\alpha}+\boldsymbol{\delta})] \geq (1-\tfrac{b}{n})\mathcal{D}(\boldsymbol{\alpha}) + \tfrac{b}{n}\mathcal{H}_{\beta_b}(\boldsymbol{\delta};\boldsymbol{\alpha}),$$

where

$$\mathcal{H}_{\beta_b}(\boldsymbol{\delta},\boldsymbol{\alpha}) = -\frac{(\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\delta} + \beta_b\|\boldsymbol{\delta}\|^2)}{2\lambda n^2} + \sum_{i=1}^n \frac{\boldsymbol{\alpha}_i + \boldsymbol{\delta}_i}{n},$$

$$\beta_b \overset{\text{def}}{=} 1 + \frac{(b-1)(n\sigma^2-1)}{n-1}.$$

Primal Suboptimality for SDCA with Safe Mini-Batching

Theorem 2

Let us run SDCA with safe mini-batching and let $\boldsymbol{\alpha}_0 = 0 \in \mathbb{R}^n$ and $\epsilon > 0$. If we let

$$
\begin{aligned}
t_0 &\geq \max\{0, \lceil \tfrac{n}{b} \log(\tfrac{2\lambda n}{\beta_b}) \rceil\}, \\
T_0 &\geq t_0 + \tfrac{\beta_b}{b} \left[\tfrac{4}{\lambda\epsilon} - 2\tfrac{n}{\beta_b} \right]_+, \\
T &\geq T_0 + \max\{\lceil \tfrac{n}{b} \rceil, \tfrac{\beta_b}{b} \tfrac{1}{\lambda\epsilon}\}, \\
\bar{\boldsymbol{\alpha}} &\stackrel{\text{def}}{=} \frac{1}{T-T_0} \sum_{t=T_0}^{T-1} \boldsymbol{\alpha}_t,
\end{aligned}
$$

then

$$\mathbf{w}(\bar{\boldsymbol{\alpha}}) \stackrel{\text{def}}{=} \frac{1}{\lambda n} \sum_{i=1}^{n} \bar{\boldsymbol{\alpha}}_i y_i \mathbf{x}_i$$

is an ϵ-approximate solution to the PRIMAL problem, i.e.,

$$\mathbf{E}[\mathcal{P}(\mathbf{w}(\bar{\boldsymbol{\alpha}}))] - \mathcal{P}(\mathbf{w}^*) \leq \mathbf{E}[\underbrace{\mathcal{P}(\mathbf{w}(\bar{\boldsymbol{\alpha}})) - \mathcal{D}(\bar{\boldsymbol{\alpha}})}_{\text{duality gap}}] \leq \epsilon.$$

Primal Suboptimality: Simple Expression

$$\frac{\beta_b}{b} \cdot \frac{5}{\lambda\epsilon} + \frac{n}{b}\left(1 + \log\left(\frac{2\lambda n}{\beta_b}\right)\right)$$

PART III:

SGD vs SDCA: Theory and Numerics


SDCA
SGD
ROCKY
IVAN

SGD vs. SDCA: Theory

Stochastic Gradient Descent (SGD)

SGD needs

$$T = \frac{\beta_b}{b} \cdot \frac{30}{\lambda \epsilon}$$

Stochastic Dual Coordinate Ascent (SDCA)

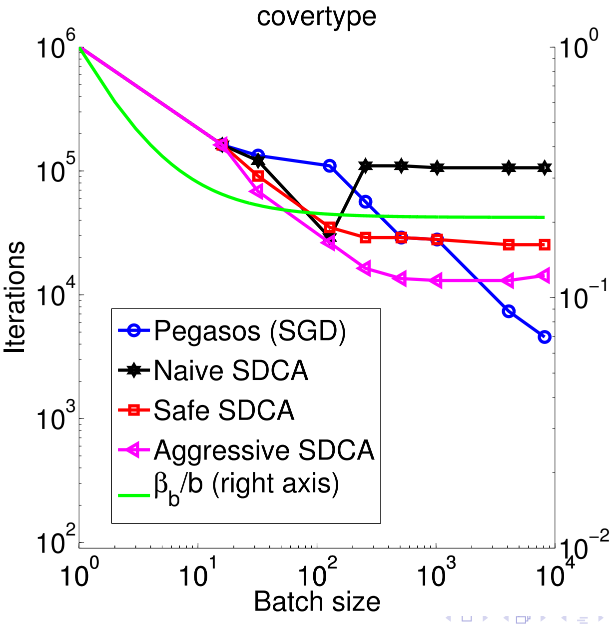
SDCA (with safe mini-batching) needs

$$T = \frac{\beta_b}{b} \cdot \frac{5}{\lambda \epsilon} + \frac{n}{b} \left(1 + \log \left(\frac{2\lambda n}{\beta_b} \right) \right)$$

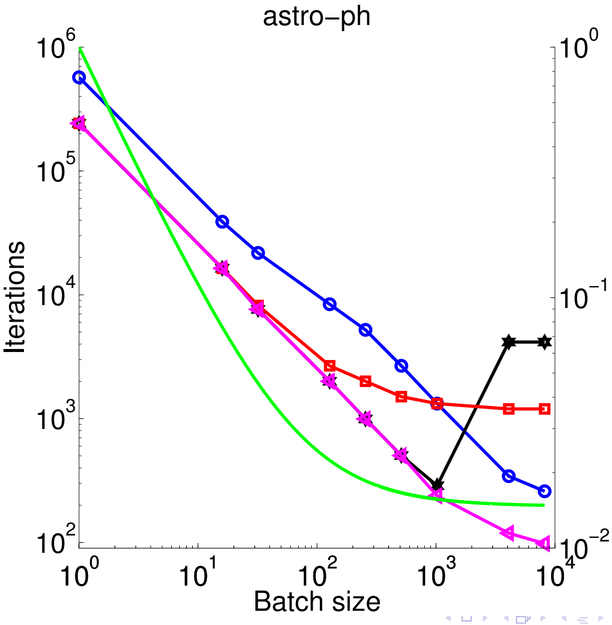
Numerical Experiments: Datasets

Data	# train	# test	# features (n)	Sparsity %	λ
cov	522,911	58,101	54	22	0.000010
rcv1	20,242	677,399	47,236	0.16	0.000100
astro-ph	29,882	32,487	99,757	0.08	0.000050
news20	15,020	4,976	1,355,191	0.04	0.000125

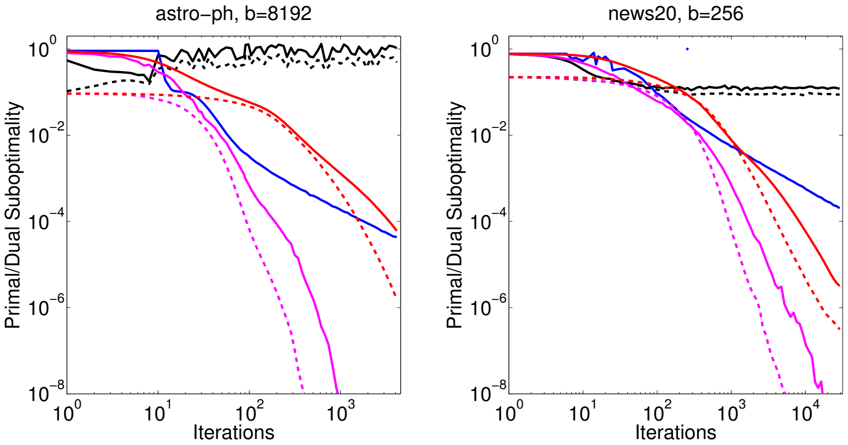
Batch Size vs Iterations $\epsilon = 0.001$



Batch Size vs Iterations $\epsilon = 0.001$

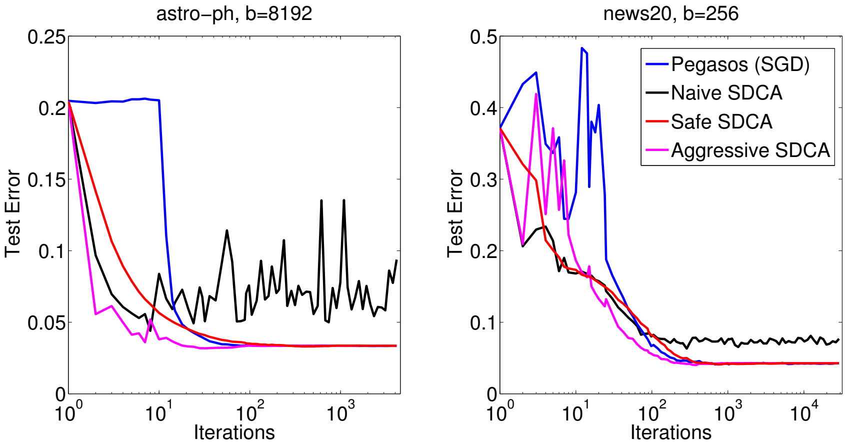


Train Error



astro−ph, b=8192
news20, b=256
Primal/Dual Suboptimality
Iterations
10^0
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^0
10^1
10^2
10^3
10^4

Test Error



astro−ph, b=8192
Test Error
0.25
0.2
0.15
0.1
0.05
0
10^0
10^1
10^2
10^3
Iterations
news20, b=256
Test Error
0.5
0.4
0.3
0.2
0.1
0
10^0
10^1
10^2
10^3
10^4
Iterations
Pegasos (SGD)
Naive SDCA
Safe SDCA
Aggressive SDCA

Summary 1

Shai Shalev-Shwartz, Yoram Singer and Nathan Srebro

Pegasos: Primal Estimated sub-GrAdient SOlver for SVM, ICML 2007

+ Analysis of SGD for $b = 1$

- Weak analysis for $b > 1$ (no speedup)

P. R. and Martin Takáč

Parallel coordinate descent methods for big data optimization, 2012

+ General analysis of SDCA for $b > 1$ (even variable b)

+ ESO: Expected Separable Overapproximation

- Dual suboptimality only

Shai Shalev-Shwartz and Tong Zhang

Stochastic dual coordinate ascent methods for regularized loss minimization, JMLR 2013

+ Primal sub-optimality for SDCA with $b = 1$

- No analysis for $b > 1$

Summary 2

Martin Takáč, Avleen Bijral, P. R. and Nathan Srebro

Mini-batch primal and dual methods for SVMs, ICML 2013

- First analysis of mini-batched SGD for SVM primal which works
- New mini-batch SDCA method for SVM dual
  - with safe mini-batching
  - with aggressive mini-batching
- Both SGD and SDCA
  - have guarantees in terms of primal suboptimality
  - spectral norm of the data controls parallelization speedup
  - have essentially identical iterations

Further Pointers

Olivier Fercoq's talk: Smooth minimization of nonsmooth functions by parallel coordinate descent