Decomposition and Stochastic Subgradient Algorithms for Support Vector Machines

Sangkyun Lee and Stephen J. Wright

University of Wisconsin-Madison

ISMP 2009

THE UNIVERSITY of WISCONSIN MADISON

Support Vector Machines

- Support Vector Machines (SVMs) are popular in many areas.

Support Vector Machines

- Support Vector Machines (SVMs) are popular in many areas.
  - Decision making, Machine learning, Statistics.
  - Bio-informatics, Neuroscience, Geophysics ...

Support Vector Machines

- Support Vector Machines (SVMs) are popular in many areas.
  - Decision making, Machine learning, Statistics.
  - Bio-informatics, Neuroscience, Geophysics ...
- For classification, regression and many other tasks.

Support Vector Machines

- Support Vector Machines (SVMs) are popular in many areas.
  - Decision making, Machine learning, Statistics.
  - Bio-informatics, Neuroscience, Geophysics ...
- For classification, regression and many other tasks.
- Result in two different types of convex programs,

“Primal”
- Number of variables = length of an input vector.
- Obj. consists of a quadratic term and a piecewise linear function.
- Costly obj. function evaluation with many input points.

“Dual”
- Number of variables = number of input points.
- QP with dense and ill-conditioned Hessian.
- A single equality constraint and bound constraints.

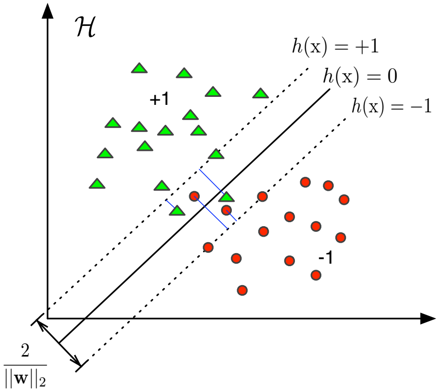
Support Vector Machines

- Support Vector Machines (SVMs) are popular in many areas.
  - Decision making, Machine learning, Statistics.
  - Bio-informatics, Neuroscience, Geophysics ...
- For classification, regression and many other tasks.
- Result in two different types of convex programs,

"Primal" $\begin{cases} \text{Number of variables = length of an input vector.} \\ \text{Obj. consists of a quadratic term and a piecewise linear function.} \\ \text{Costly obj. function evaluation with many input points.} \end{cases}$

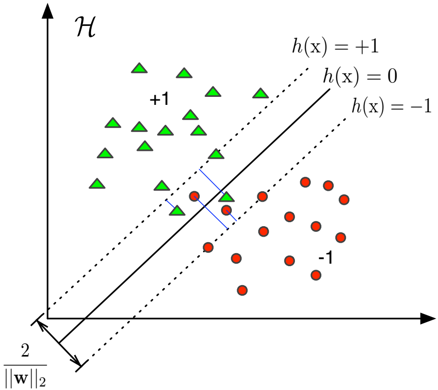
"Dual" $\begin{cases} \text{Number of variables = number of input points.} \\ \text{QP with dense and ill-conditioned Hessian.} \\ \text{A single equality constraint and bound constraints.} \end{cases}$

SVMs for Classification (SVC)



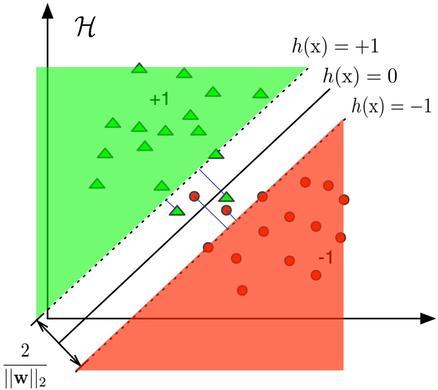
- $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{M}$ *i.i.d.* $\sim P(X, Y)$,
  - $\boldsymbol{x}_i \in \mathbb{R}^N$.
  - $\boldsymbol{y}_i \in \{-1, +1\}$.

SVMs for Classification (SVC)



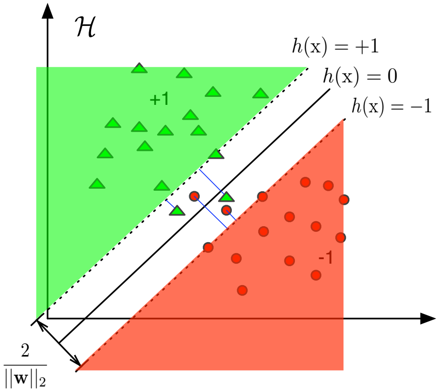
- $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{M}$ *i.i.d.* $\sim P(X, Y)$,
  - $\boldsymbol{x}_i \in \mathbb{R}^N$.
  - $\boldsymbol{y}_i \in \{-1, +1\}$.
- $\phi : \mathbb{R}^N \longrightarrow \mathcal{H}$.

SVMs for Classification (SVC)



- $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{M}$ *i.i.d.* $\sim P(X, Y)$,
  - $\boldsymbol{x}_i \in \mathbb{R}^N$.
  - $\boldsymbol{y}_i \in \{-1, +1\}$.
- $\phi : \mathbb{R}^N \longrightarrow \mathcal{H}$.
- Find a classifier $h(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle + b$,
  - $h(\boldsymbol{x}_i) \geq +1$ for $\boldsymbol{y}_i = +1$,
  - $h(\boldsymbol{x}_i) \leq -1$ for $\boldsymbol{y}_i = -1$,
  - Maximizing the “margin” $2/||\boldsymbol{w}||_2$.

SVMs for Classification (SVC)



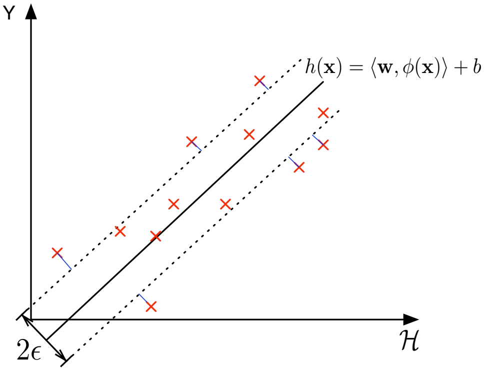
- $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{M}$ *i.i.d.* $\sim P(X, Y)$,
  - $\boldsymbol{x}_i \in \mathbb{R}^N$.
  - $\boldsymbol{y}_i \in \{-1, +1\}$.
- $\phi : \mathbb{R}^N \longrightarrow \mathcal{H}$.
- Find a classifier $h(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle + b$,
  - $h(\boldsymbol{x}_i) \geq +1$ for $\boldsymbol{y}_i = +1$,
  - $h(\boldsymbol{x}_i) \leq -1$ for $\boldsymbol{y}_i = -1$,
  - Maximizing the "margin" $2/||\boldsymbol{w}||_2$.

$$\min_{\boldsymbol{w}, b} \quad \frac{1}{2}||\boldsymbol{w}||_2^2 + \frac{C}{M} \sum_{i=1}^{M} \ell_{\mathrm{H}}(h; \boldsymbol{x}_i, \boldsymbol{y}_i) \ ,$$

Hinge loss: $\ell_{\mathrm{H}}(h; \boldsymbol{x}_i, \boldsymbol{y}_i) := \max\{1 - \boldsymbol{y}_i h(\boldsymbol{x}_i), 0\}$.

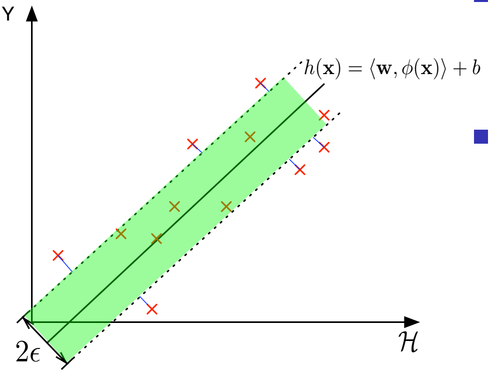
THE UNIVERSITY of WISCONSIN MADISON

SVMs for Regression (SVR)



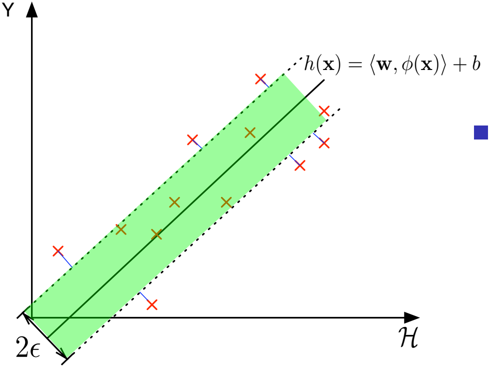
- $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{M}$ $i.i.d \sim P(X, Y)$,
  - $\boldsymbol{x}_i \in \mathbb{R}^N$.
  - $\boldsymbol{y}_i \in \mathbb{R}$.

SVMs for Regression (SVR)



- $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{M}$ $i.i.d \sim P(X, Y)$,
  - $\boldsymbol{x}_i \in \mathbb{R}^N$.
  - $\boldsymbol{y}_i \in \mathbb{R}$.
- Find a regression function, $h(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle + b$
  - Minimizing prediction error.
  - Capture data points in an ϵ-radius hyper-tube surrounding $h(\boldsymbol{x})$.

SVMs for Regression (SVR)



- $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{M}$ $i.i.d \sim P(X, Y)$,
  - $\boldsymbol{x}_i \in \mathbb{R}^N$.
  - $\boldsymbol{y}_i \in \mathbb{R}$.
- Find a regression function, $h(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle + b$
  - Minimizing prediction error.
  - Capture data points in an ϵ-radius hyper-tube surrounding $h(\boldsymbol{x})$.

$$\min_{\boldsymbol{w}, b} \quad \frac{1}{2}||\boldsymbol{w}||_2^2 + \frac{C}{M}\sum_{i=1}^{M} \ell_\epsilon(h; \boldsymbol{x}_i, \boldsymbol{y}_i),$$

ϵ-insensitive loss: $\quad \ell_\epsilon(h; \boldsymbol{x}_i, \boldsymbol{y}_i) := \max\{|\boldsymbol{y}_i - h(\boldsymbol{x}_i)| - \epsilon, 0\}.$

SVM Formulations of Interest

Primal

$$\min_{\boldsymbol{w},b} \ \frac{\lambda}{2}||\boldsymbol{w}||_2^2 + R_{\mathrm{emp}}(h; \boldsymbol{x}, \boldsymbol{y}),$$

where

$$R_{\mathrm{emp}} = \begin{cases} \dfrac{1}{M}\displaystyle\sum_{i=1}^{M} \ell_{\mathrm{H}}(h; \boldsymbol{x}_i, \boldsymbol{y}_i), & \text{(SVC)} \\ \dfrac{1}{M}\displaystyle\sum_{i=1}^{M} \ell_{\epsilon}(h; \boldsymbol{x}_i, \boldsymbol{y}_i), & \text{(SVR)} \end{cases}$$

and $\lambda = 1/C$. The objective function is convex but non-smooth.

SVM Formulations of Interest

Primal

$$\min_{\boldsymbol{w},b} \ \frac{\lambda}{2}||\boldsymbol{w}||_2^2 + R_{\text{emp}}(h; \boldsymbol{x}, \boldsymbol{y}),$$

where

$$R_{\text{emp}} = \begin{cases} \frac{1}{M}\sum_{i=1}^{M} \ell_{\text{H}}(h; \boldsymbol{x}_i, \boldsymbol{y}_i), & \text{(SVC)} \\ \frac{1}{M}\sum_{i=1}^{M} \ell_{\epsilon}(h; \boldsymbol{x}_i, \boldsymbol{y}_i), & \text{(SVR)} \end{cases}$$

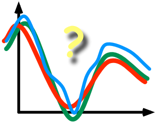
and $\lambda = 1/C$. The objective function is convex but non-smooth.

Dual

$$\begin{aligned} \min_{\boldsymbol{z}} \quad & \frac{1}{2}\boldsymbol{z}^T\boldsymbol{Q}\boldsymbol{z} + \boldsymbol{p}^T\boldsymbol{z} \\ \text{s.t.} \quad & \boldsymbol{c}^T\boldsymbol{z} = d \\ & \boldsymbol{\ell} \le \boldsymbol{z} \le \boldsymbol{u} \ , \end{aligned} \tag{1}$$

- $\boldsymbol{Q}$ is a p.s.d. $n \times n$ matrix, usually dense and ill-conditioned.
  - $n = M$ (SVC) or $n = 2M$ (SVR)
  - Determined by $\boldsymbol{y}$ and kernel function $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) := \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$.
- $\boldsymbol{z}, \boldsymbol{p}, \boldsymbol{c}, \boldsymbol{\ell}, \boldsymbol{u} \in \mathbb{R}^n$, and $d \in \mathbb{R}$.

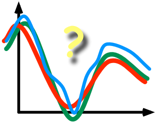
Semiparametric SVM



- Standard (nonparametric) SVR: use a linear model

$$h(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle + b \ ,$$

Semiparametric SVM



- Standard (nonparametric) SVR: use a linear model

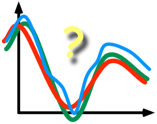
$$h(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle + b \ ,$$

- Semiparametric SVR [SFS99]: use an extended linear model

$$\tilde{h}(\boldsymbol{x}) = \underbrace{\langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle}_{\text{Nonparametric part}} + \underbrace{\sum_{j=1}^{K} \beta_j \psi_j(\boldsymbol{x})}_{\text{Parametric part}} \ ,$$

where $\psi_j(\cdot)$'s are user-defined (basis) functions.

Semiparametric SVM



- Standard (nonparametric) SVR: use a linear model

$$h(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle + b \ ,$$

- Semiparametric SVR [SFS99]: use an extended linear model

$$\tilde{h}(\boldsymbol{x}) = \underbrace{\langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle}_{\text{Nonparametric part}} + \underbrace{\sum_{j=1}^{K} \beta_j \psi_j(\boldsymbol{x})}_{\text{Parametric part}} \ ,$$

where $\psi_j(\cdot)$'s are user-defined (basis) functions.

Benefits of semiparametric models

- No explicit modeling is necessary (nonparametric).
- Embedding of prior knowledge / model interpretation (parametric).

THE UNIVERSITY of WISCONSIN MADISON

Primal Formulation

The "primal" SVR formulation is,

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{C}{M}\sum_{i=1}^{M}\ell_\epsilon(\tilde{h};\boldsymbol{x}_i,\boldsymbol{y}_i), \quad \ell_\epsilon(\tilde{h};\boldsymbol{x}_i,\boldsymbol{y}_i) := \max\{|\boldsymbol{y}_i - \tilde{h}(\boldsymbol{x}_i)| - \epsilon, 0\}.$$

Introducing slack variables $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_i^*$ to represent the deviations from the ϵ-tube, we obtain

$$\min_{\boldsymbol{w},\boldsymbol{\beta},\boldsymbol{\xi},\boldsymbol{\xi}^*} \quad \frac{1}{2}\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{w} + \frac{C}{M}\sum_{i=1}^{M}(\boldsymbol{\xi}_i + \boldsymbol{\xi}_i^*) \tag{2a}$$

$$\text{s.t.} \quad \boldsymbol{y}_i - \langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle - \sum_{j=1}^{K}\boldsymbol{\beta}_j\psi_j(\boldsymbol{x}_i) \le \epsilon + \boldsymbol{\xi}_i \quad \text{for } i = 1,\ldots,M \tag{2b}$$

$$-\left[\boldsymbol{y}_i - \langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle - \sum_{j=1}^{K}\boldsymbol{\beta}_j\psi_j(\boldsymbol{x}_i)\right] \le \epsilon + \boldsymbol{\xi}_i^* \quad \text{for } i = 1,\ldots,M \tag{2c}$$

$$\boldsymbol{\xi} \ge \boldsymbol{0}, \ \boldsymbol{\xi}^* \ge \boldsymbol{0} \ . \tag{2d}$$

Dual Formulation

$$\min_{\boldsymbol{z}} \; F(\boldsymbol{z}) := \frac{1}{2}\boldsymbol{z}^T \boldsymbol{Q}\boldsymbol{z} + \boldsymbol{p}^T\boldsymbol{z} \qquad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{z} = \boldsymbol{0}, \quad \boldsymbol{0} \le \boldsymbol{z} \le \frac{C}{M}\boldsymbol{1}, \qquad (3)$$

where $\boldsymbol{z}, \boldsymbol{p} \in \mathbb{R}^{2M}$, $\boldsymbol{Q} \in \mathbb{R}^{2M\times 2M}$ p.s.d., and $A \in \mathbb{R}^{K\times 2M}$.

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^* \end{bmatrix} \in \mathbb{R}^{2M} \text{ for the dual vectors } \boldsymbol{\alpha} \text{ and } \boldsymbol{\alpha}^* \text{ of (2b) and (2c), resp.,}$$

$$\boldsymbol{p} = [\epsilon - \boldsymbol{y}_1, \ldots, \epsilon - \boldsymbol{y}_M, \epsilon + \boldsymbol{y}_1, \ldots, \epsilon + \boldsymbol{y}_M]^T \in \mathbb{R}^{2M} \;,$$

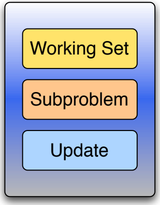
$$\boldsymbol{Q}_{ij} = \begin{cases} \boldsymbol{y}_i\boldsymbol{y}_j\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) & \text{if } 1 \le i,j \le M, \text{ or } M+1 \le i,j \le 2M \\ -\boldsymbol{y}_i\boldsymbol{y}_j\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) & \text{otherwise} \end{cases} \;,$$

$$\boldsymbol{A} = \begin{bmatrix} \psi_1(\boldsymbol{x}_1) & \cdots & \psi_1(\boldsymbol{x}_M) & -\psi_1(\boldsymbol{x}_1) & \cdots & -\psi_1(\boldsymbol{x}_M) \\ \psi_2(\boldsymbol{x}_1) & \cdots & \psi_2(\boldsymbol{x}_M) & -\psi_2(\boldsymbol{x}_1) & \cdots & -\psi_2(\boldsymbol{x}_M) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \psi_K(\boldsymbol{x}_1) & \cdots & \psi_K(\boldsymbol{x}_M) & -\psi_K(\boldsymbol{x}_1) & \cdots & -\psi_K(\boldsymbol{x}_M) \end{bmatrix} \in \mathbb{R}^{K\times 2M} \;.$$

This is a generalization of the standard SVM dual problem. $n := 2M$.

Decomposition Framework [LW09]

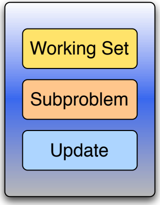
- In each outer iteration, we split variables $\boldsymbol{z}$ into
  - Basic variables $\boldsymbol{z}_{\mathcal{B}}$, $\mathcal{B} \subset \{1, 2, \ldots, n\}$.
  - Nonbasic variables $\boldsymbol{z}_{\mathcal{N}}$, $\mathcal{N} = \{1, 2, \ldots, n\} \setminus \mathcal{B}$.



THE UNIVERSITY of WISCONSIN MADISON

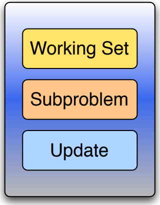
Decomposition Framework [LW09]

- In each outer iteration, we split variables $\boldsymbol{z}$ into
  - Basic variables $\boldsymbol{z}_{\mathcal{B}}$, $\mathcal{B} \subset \{1, 2, \ldots, n\}$.
  - Nonbasic variables $\boldsymbol{z}_{\mathcal{N}}$, $\mathcal{N} = \{1, 2, \ldots, n\} \setminus \mathcal{B}$.
  - $\mathcal{B}$ is our working set, of which the size $n_{\mathcal{B}} \ll n$.



Decomposition Framework [LW09]

- In each outer iteration, we split variables $\boldsymbol{z}$ into
  - Basic variables $\boldsymbol{z}_{\mathcal{B}}$, $\mathcal{B} \subset \{1, 2, \ldots, n\}$.
  - Nonbasic variables $\boldsymbol{z}_{\mathcal{N}}$, $\mathcal{N} = \{1, 2, \ldots, n\} \setminus \mathcal{B}$.
  - $\mathcal{B}$ is our working set, of which the size $n_{\mathcal{B}} \ll n$.
- Fix $\boldsymbol{z}_{\mathcal{N}}$, change $\boldsymbol{z}_{\mathcal{B}}$.



Decomposition Framework [LW09]

- In each outer iteration, we split variables $\boldsymbol{z}$ into
  - Basic variables $\boldsymbol{z}_{\mathcal{B}}$, $\mathcal{B} \subset \{1, 2, \dots, n\}$.
  - Nonbasic variables $\boldsymbol{z}_{\mathcal{N}}$, $\mathcal{N} = \{1, 2, \dots, n\} \setminus \mathcal{B}$.
  - $\mathcal{B}$ is our working set, of which the size $n_{\mathcal{B}} \ll n$.
- Fix $\boldsymbol{z}_{\mathcal{N}}$, change $\boldsymbol{z}_{\mathcal{B}}$.
- Given $\boldsymbol{z}^k = (\boldsymbol{z}^k_{\mathcal{B}}, \boldsymbol{z}^k_{\mathcal{N}})$, we solve the subproblem to get $\boldsymbol{z}^{k+1}_{\mathcal{B}}$.

Working Set

Subproblem

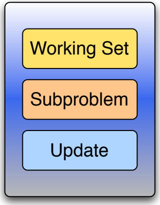
Update

Subproblem

$$\min_{\boldsymbol{z}_{\mathcal{B}}} \quad f(\boldsymbol{z}_{\mathcal{B}}) := \frac{1}{2} \boldsymbol{z}_{\mathcal{B}}^T \boldsymbol{Q}_{\mathcal{B}\mathcal{B}} \boldsymbol{z}_{\mathcal{B}} + (\boldsymbol{Q}_{\mathcal{B}\mathcal{N}} \boldsymbol{z}^k_{\mathcal{N}} + \boldsymbol{p}_{\mathcal{B}})^T \boldsymbol{z}_{\mathcal{B}} \tag{4}$$

$$\text{s.t.} \quad \boldsymbol{A}_{\mathcal{B}} \boldsymbol{z}_{\mathcal{B}} = -\boldsymbol{A}_{\mathcal{N}} \boldsymbol{z}^k_{\mathcal{N}} + \boldsymbol{b}, \qquad 0 \le \boldsymbol{z}_{\mathcal{B}} \le \frac{C}{M} \boldsymbol{1}.$$

Decomposition Framework [LW09]



- In each outer iteration, we split variables $\boldsymbol{z}$ into
  - Basic variables $\boldsymbol{z}_{\mathcal{B}}$, $\mathcal{B} \subset \{1, 2, \ldots, n\}$.
  - Nonbasic variables $\boldsymbol{z}_{\mathcal{N}}$, $\mathcal{N} = \{1, 2, \ldots, n\} \setminus \mathcal{B}$.
  - $\mathcal{B}$ is our working set, of which the size $n_{\mathcal{B}} \ll n$.
- Fix $\boldsymbol{z}_{\mathcal{N}}$, change $\boldsymbol{z}_{\mathcal{B}}$.
- Given $\boldsymbol{z}^k = (\boldsymbol{z}^k_{\mathcal{B}}, \boldsymbol{z}^k_{\mathcal{N}})$, we solve the subproblem to get $\boldsymbol{z}^{k+1}_{\mathcal{B}}$.

Subproblem

$$\begin{aligned} \min_{\boldsymbol{z}_{\mathcal{B}}} \quad & f(\boldsymbol{z}_{\mathcal{B}}) := \frac{1}{2}\boldsymbol{z}_{\mathcal{B}}^T \boldsymbol{Q}_{\mathcal{B}\mathcal{B}} \boldsymbol{z}_{\mathcal{B}} + (\boldsymbol{Q}_{\mathcal{B}\mathcal{N}} \boldsymbol{z}^k_{\mathcal{N}} + \boldsymbol{p}_{\mathcal{B}})^T \boldsymbol{z}_{\mathcal{B}} \\ \text{s.t.} \quad & \boldsymbol{A}_{\mathcal{B}} \boldsymbol{z}_{\mathcal{B}} = -\boldsymbol{A}_{\mathcal{N}} \boldsymbol{z}^k_{\mathcal{N}} + \boldsymbol{b}, \qquad 0 \le \boldsymbol{z}_{\mathcal{B}} \le \frac{C}{M}\mathbf{1}. \end{aligned} \tag{4}$$

- $\boldsymbol{z}^{k+1} \leftarrow (\boldsymbol{z}^{k+1}_{\mathcal{B}}, \boldsymbol{z}^k_{\mathcal{N}})$.

Choosing $\mathcal{B}$: Working Set Selection

Working Set
Subproblem
Update

- Inspired by the approach of [Joa99], later improved by [SZ05].
  - $n_{\mathcal{B}}$: working set size.
  - n_c: max. number of "fresh" indices. $n_c \ll n_{\mathcal{B}}$.

THE UNIVERSITY of WISCONSIN MADISON

Choosing $\mathcal{B}$: Working Set Selection

- Inspired by the approach of [Joa99], later improved by [SZ05].
  - $n_{\mathcal{B}}$: working set size.
  - n_c: max. number of "fresh" indices. $n_c \ll n_{\mathcal{B}}$.

Consider Lagrangian relaxation $\mathcal{L}$ of the dual formulation (3),

$$\mathcal{L}(\boldsymbol{z};\boldsymbol{\eta}) = F(\boldsymbol{z}) + \boldsymbol{\eta}^T \boldsymbol{A}\boldsymbol{z} \ . \tag{5}$$

Given $(\boldsymbol{z}^k, \boldsymbol{\eta}^k)$, find a solution $\boldsymbol{d}$ of

$$\begin{aligned}
\min_{\boldsymbol{d}} \quad & \left(\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}^k;\boldsymbol{\eta}^k)\right)^T \boldsymbol{d} \\
\text{s.t.} \quad & 0 \le \boldsymbol{d}_i \le 1 && \text{if } \boldsymbol{z}_i^{k+1} = 0, \\
& -1 \le \boldsymbol{d}_i \le 0 && \text{if } \boldsymbol{z}_i^{k+1} = C/M, \\
& -1 \le \boldsymbol{d}_i \le 1 && \text{if } \boldsymbol{z}_i^{k+1} \in (0, C/M), \\
& \#\{\boldsymbol{d}_i | \boldsymbol{d}_i \ne 0\} \le n_c.
\end{aligned} \tag{6}$$

Choosing $\mathcal{B}$: Working Set Selection

- Inspired by the approach of [Joa99], later improved by [SZ05].
  - $n_{\mathcal{B}}$: working set size.
  - n_c: max. number of "fresh" indices. $n_c \ll n_{\mathcal{B}}$.

Consider Lagrangian relaxation $\mathcal{L}$ of the dual formulation (3),

$$\mathcal{L}(\boldsymbol{z};\boldsymbol{\eta}) = F(\boldsymbol{z}) + \boldsymbol{\eta}^T \boldsymbol{A}\boldsymbol{z} \quad . \tag{5}$$

Given $(\boldsymbol{z}^k, \boldsymbol{\eta}^k)$, find a solution $\boldsymbol{d}$ of

$$\begin{aligned} \min_{\boldsymbol{d}} \quad & \left(\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}^k;\boldsymbol{\eta}^k)\right)^T \boldsymbol{d} \\ \text{s.t.} \quad & 0 \le \boldsymbol{d}_i \le 1 && \text{if } \boldsymbol{z}_i^{k+1} = 0, \\ & -1 \le \boldsymbol{d}_i \le 0 && \text{if } \boldsymbol{z}_i^{k+1} = C/M, \\ & -1 \le \boldsymbol{d}_i \le 1 && \text{if } \boldsymbol{z}_i^{k+1} \in (0, C/M), \\ & \#\{\boldsymbol{d}_i | \boldsymbol{d}_i \neq 0\} \le n_c. \end{aligned} \tag{6}$$

- Solved efficiently, $\mathcal{O}(n \log n)$.
- Convergence of decomposition + workingset selection [Lin01, TY08].

Subproblem: Primal-dual Solver (PDSG)

- We consider the following formulation of (4):

$$\max_{\boldsymbol{\eta}} \min_{\boldsymbol{z}_{\mathcal{B}} \in \Omega} \quad \tilde{\mathcal{L}}(\boldsymbol{z}_{\mathcal{B}}, \boldsymbol{\eta}) \ , \tag{7}$$

where

$$\tilde{\mathcal{L}}(\boldsymbol{z}_{\mathcal{B}}, \boldsymbol{\eta}) := f(\boldsymbol{z}_{\mathcal{B}}) + \boldsymbol{\eta}^T (\boldsymbol{A}_{\mathcal{B}} \boldsymbol{z}_{\mathcal{B}} + \boldsymbol{A}_{\mathcal{N}} \boldsymbol{z}_{\mathcal{N}}^k) \ ,$$

$$\Omega = \{\boldsymbol{z}_{\mathcal{B}} \in \mathbb{R}^{n_{\mathcal{B}}} | \boldsymbol{0} \leq \boldsymbol{z}_{\mathcal{B}} \leq \frac{C}{M} \boldsymbol{1}\} \ .$$

Subproblem: Primal-dual Solver (PDSG)

- We consider the following formulation of (4):

$$\max_{\boldsymbol{\eta}} \min_{\boldsymbol{z}_{\mathcal{B}} \in \Omega} \quad \tilde{\mathcal{L}}(\boldsymbol{z}_{\mathcal{B}}, \boldsymbol{\eta}) \ , \tag{7}$$

where

$$\tilde{\mathcal{L}}(\boldsymbol{z}_{\mathcal{B}}, \boldsymbol{\eta}) := f(\boldsymbol{z}_{\mathcal{B}}) + \boldsymbol{\eta}^T(\boldsymbol{A}_{\mathcal{B}}\boldsymbol{z}_{\mathcal{B}} + \boldsymbol{A}_{\mathcal{N}}\boldsymbol{z}_{\mathcal{N}}^k) \ ,$$

$$\Omega = \{\boldsymbol{z}_{\mathcal{B}} \in \mathbb{R}^{n_{\mathcal{B}}} | \boldsymbol{0} \leq \boldsymbol{z}_{\mathcal{B}} \leq \frac{C}{M}\boldsymbol{1}\} \ .$$

In each "inner" iteration, update primal and dual variables by,

$$\begin{cases} \boldsymbol{z}_{\mathcal{B}}^{\ell+1} \leftarrow \boldsymbol{z}_{\mathcal{B}}^{\ell} + s(\boldsymbol{z}_{\mathcal{B}}^{\ell}, \boldsymbol{\eta}^{\ell}) \\ \boldsymbol{\eta}^{\ell+1} \leftarrow \boldsymbol{\eta}^{\ell} + t(\boldsymbol{z}_{\mathcal{B}}^{\ell+1}, \boldsymbol{\eta}^{\ell}) \ , \end{cases} \tag{8}$$

- Primal step $s(\cdot, \cdot)$ is chosen by two-metric GP [GB84] followed by line-search, on a sub-workingset of size 2.
- Dual step $t(\cdot, \cdot)$ is a direction $\nabla_{\boldsymbol{\eta}} \tilde{\mathcal{L}}$, scaled by dual Hessian diagonal [KS05], on a sub-workingset of size 2.

Update

- Update primal-dual iterate pair $(\boldsymbol{z}^{k+1}, \boldsymbol{\eta}^{k+1})$.
  - $\boldsymbol{z}^{k+1} \leftarrow (\boldsymbol{z}_{\mathcal{B}}^{k+1}, \boldsymbol{z}_{\mathcal{N}}^{k})$.
  - $\boldsymbol{\eta}^{k+1}$ is provided by the subproblem solver.

Update

- Update primal-dual iterate pair $(\boldsymbol{z}^{k+1}, \boldsymbol{\eta}^{k+1})$.
  - $\boldsymbol{z}^{k+1} \leftarrow (\boldsymbol{z}_{\mathcal{B}}^{k+1}, \boldsymbol{z}_{\mathcal{N}}^{k})$.
  - η^{k+1} is provided by the subproblem solver.
- "Full gradient" $\nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}; \eta)$ has to be updated.
  - To check KKT conditions violation.
  - For the next working set selection.

Update

- Update primal-dual iterate pair $(\boldsymbol{z}^{k+1}, \boldsymbol{\eta}^{k+1})$.
  - $\boldsymbol{z}^{k+1} \leftarrow (\boldsymbol{z}_{\mathcal{B}}^{k+1}, \boldsymbol{z}_{\mathcal{N}}^{k})$.
  - η^{k+1} is provided by the subproblem solver.

- "Full gradient" $\nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}; \eta)$ has to be updated.
  - To check KKT conditions violation.
  - For the next working set selection.

Update incrementally,

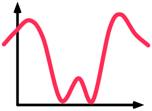
$$\nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}^{k+1}, \boldsymbol{\eta}^{k+1}) = \nabla F(\boldsymbol{z}^{k+1}) + (\boldsymbol{\eta}^{k+1})^T \boldsymbol{A} \tag{9}$$

$$= \nabla F(\boldsymbol{z}^{k}) + \begin{bmatrix} \boldsymbol{Q}_{\mathcal{B}\mathcal{B}} \\ \boldsymbol{Q}_{\mathcal{N}\mathcal{B}} \end{bmatrix} (\boldsymbol{z}_{\mathcal{B}}^{k+1} - \boldsymbol{z}_{\mathcal{B}}^{k}) + (\boldsymbol{\eta}^{k+1})^T \boldsymbol{A} \; . \tag{10}$$

Optimality Conditions
Algorithm Summary

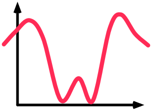
Experiments

- A toy test problem: modified Mexican hat function [SFS99, KS05]:
$$\omega(x) = \sin(x) + \operatorname{sinc}\left(2\pi(x-5)\right).$$



Experiments

- A toy test problem: modified Mexican hat function [SFS99, KS05]:

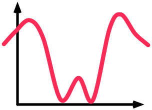


$$\omega(x) = \sin(x) + \mathrm{sinc}\left(2\pi(x-5)\right).$$

  - Sample y_i's from ω at uniform random points x_i's in $[0, 10]$ with additive noise $\zeta_i \sim \mathcal{N}(0, 0.2^2)$: $y_i = \omega(x_i) + \zeta_i$.

Experiments

- A toy test problem: modified Mexican hat function [SFS99, KS05]:



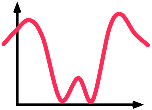
$$\omega(x) = \sin(x) + \operatorname{sinc}\left(2\pi(x-5)\right).$$

  - Sample y_i's from ω at uniform random points x_i's in $[0, 10]$ with additive noise $\zeta_i \sim \mathcal{N}(0, 0.2^2)$: $y_i = \omega(x_i) + \zeta_i$.

- Experiment settings
  - Parametric components: $\psi_1(x) = \sin(x), \psi_2(x) = \operatorname{sinc}(2\pi(x-5))$.
  - Gaussian kernel $\kappa(x, y) = \exp(-\gamma||x-y||^2)$ with $\gamma = 0.25$.
  - Loss function parameter $\epsilon = 0.05$.
  - $n_{\mathcal{B}} = 500$, $n_{\mathrm{c}} = n_{\mathcal{B}}/5$.

Experiments

- A toy test problem: modified Mexican hat function [SFS99, KS05]:



$$\omega(x) = \sin(x) + \operatorname{sinc}(2\pi(x-5)).$$

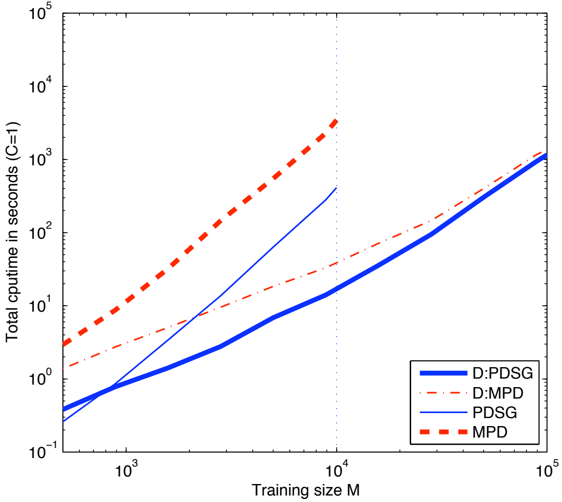
  - Sample y_i's from ω at uniform random points x_i's in $[0, 10]$ with additive noise $\zeta_i \sim \mathcal{N}(0, 0.2^2)$: $y_i = \omega(x_i) + \zeta_i$.

- Experiment settings
  - Parametric components: $\psi_1(x) = \sin(x), \psi_2(x) = \operatorname{sinc}(2\pi(x-5))$.
  - Gaussian kernel $\kappa(x, y) = \exp(-\gamma||x-y||^2)$ with $\gamma = 0.25$.
  - Loss function parameter $\epsilon = 0.05$.
  - $n_{\mathcal{B}} = 500$, $n_{\mathrm{c}} = n_{\mathcal{B}}/5$.

- Compare to the current best solver, MPD [KS05].
  - Handles the problem as a whole. Working set size is 1.
  - Primal-dual method, based on the method of multipliers.
  - Primal: gradient projection, dual: scaled gradient ascent.

THE UNIVERSITY of WISCONSIN MADISON

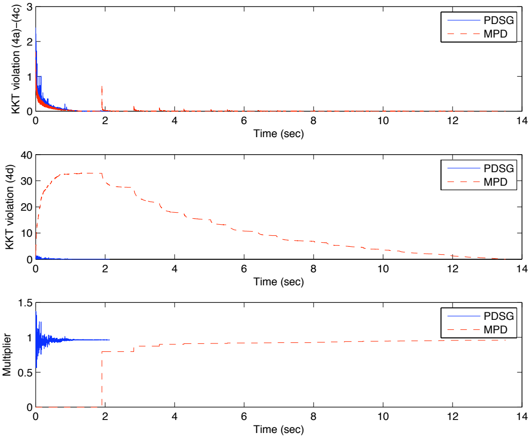
Scaling w.r.t. Training Size



- PDSG vs. MPD (stand-alone).
- D:PDSG vs. D:MPD (in decomposition).
- $C = 1$.
- $D : MPD$ catches up $D : PDSG$ when $M \uparrow$: the full gradient update step becomes dominant as M grows.

Varying K

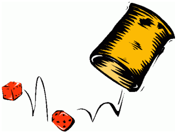
Convergence Behavior



- PDSG vs. MPD (stand-alone).
- $M = 1000$.
- PDSG: 2 sec.
- MPD: 14 sec.
- (Top) max. violation of the dual feasibility conditions.
- (Middle) max. violation of the primal equality constraints.
- (Bottom) convergence of the coefficient of the first parametric basis function.

THE UNIVERSITY of WISCONSIN MADISON

Stochastic Subgradient Methods for SVMs



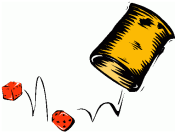
- Recent ML research on solving the primal formulation[1],

$$\min_{\boldsymbol{w},b}\ f(\boldsymbol{w},\mathcal{D}) = \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{1}{M}\sum_{i=1}^{M}\ell_{\mathrm{H}}(\boldsymbol{w};\boldsymbol{x}_i,\boldsymbol{y}_i). \tag{11}$$

  - A large dataset $\mathcal{D} := \{(\boldsymbol{x}_i,\boldsymbol{y}_i) : i = 1,\ldots,M\}$.

[1,*] ; $\boldsymbol{w} \leftarrow (\boldsymbol{w}, b)$

Stochastic Subgradient Methods for SVMs



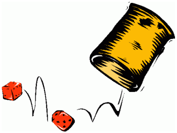
- Recent ML research on solving the primal formulation[1],

$$\min_{\boldsymbol{w},b} \; f(\boldsymbol{w}, \mathcal{D}) = \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{1}{M}\sum_{i=1}^{M} \ell_{\mathrm{H}}(\boldsymbol{w}; \boldsymbol{x}_i, \boldsymbol{y}_i). \tag{11}$$

  - A large dataset $\mathcal{D} := \{(\boldsymbol{x}_i, \boldsymbol{y}_i) : i = 1, \ldots, M\}$.
  - The objective function is strongly convex*.

[1],*; $\boldsymbol{w} \leftarrow (\boldsymbol{w}, b)$

Stochastic Subgradient Methods for SVMs



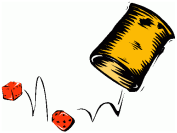
- Recent ML research on solving the primal formulation[1],

$$\min_{\boldsymbol{w},b} \; f(\boldsymbol{w},\mathcal{D}) = \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{1}{M}\sum_{i=1}^{M}\ell_{\mathrm{H}}(\boldsymbol{w};\boldsymbol{x}_i,\boldsymbol{y}_i). \tag{11}$$

  - A large dataset $\mathcal{D} := \{(\boldsymbol{x}_i,\boldsymbol{y}_i) : i = 1,\ldots,M\}$.
  - The objective function is strongly convex*.
- These has a connection to stochastic approximation methods that have developed in the past 50 years and still active.

[1,*]; $\boldsymbol{w} \leftarrow (\boldsymbol{w}, b)$

Stochastic Subgradient Methods for SVMs



- Recent ML research on solving the primal formulation[1],

$$\min_{\boldsymbol{w},b}\; f(\boldsymbol{w},\mathcal{D}) = \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{1}{M}\sum_{i=1}^{M}\ell_{\mathrm{H}}(\boldsymbol{w};\boldsymbol{x}_i,\boldsymbol{y}_i). \tag{11}$$

  - A large dataset $\mathcal{D} := \{(\boldsymbol{x}_i,\boldsymbol{y}_i) : i = 1,\ldots,M\}$.
  - The objective function is strongly convex*.
- These has a connection to stochastic approximation methods that have developed in the past 50 years and still active.
- New issues arise when applied to machine learning problems.

[1],*; $\boldsymbol{w} \leftarrow (\boldsymbol{w}, b)$

THE UNIVERSITY of WISCONSIN MADISON

Large-scale Linear SVM Training [Bot, SSSS07]

Given $\mathcal{D}$, consider the subgradient of an approximate objective function $\tilde{f}(\boldsymbol{w}; \mathcal{D}_t)$ of $f(\boldsymbol{w}; \mathcal{D})$ in (11) for a sample dataset $\mathcal{D}_t \subseteq \mathcal{D}$:

$$\tilde{f}(\boldsymbol{w}; \mathcal{D}_t) := \frac{\lambda}{2} \boldsymbol{w}^T \boldsymbol{w} + \frac{1}{|\mathcal{D}_t|} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}_t} \ell_{\mathrm{H}}(\boldsymbol{w}; (\boldsymbol{x}, y))$$

$$g(\boldsymbol{w}_t; \mathcal{D}_t) := \lambda \boldsymbol{w}_t - \frac{1}{|\mathcal{D}_t|} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}_t^+} y \boldsymbol{x} \quad \in \partial \tilde{f}(\boldsymbol{w}; \mathcal{D}_t) \ ,$$

where $\mathcal{D}_t^+ := \{(\boldsymbol{x}, y) \in \mathcal{D}_t : 1 - y(\boldsymbol{w}^T \boldsymbol{x}) > 0\}$.

THE UNIVERSITY of WISCONSIN MADISON

Large-scale Linear SVM Training [Bot, SSSS07]

Given $\mathcal{D}$, consider the subgradient of an approximate objective function $\tilde{f}(\boldsymbol{w}; \mathcal{D}_t)$ of $f(\boldsymbol{w}; \mathcal{D})$ in (11) for a sample dataset $\mathcal{D}_t \subseteq \mathcal{D}$:

$$\tilde{f}(\boldsymbol{w}; \mathcal{D}_t) := \frac{\lambda}{2} \boldsymbol{w}^T \boldsymbol{w} + \frac{1}{|\mathcal{D}_t|} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}_t} \ell_{\mathrm{H}}(\boldsymbol{w}; (\boldsymbol{x}, y))$$

$$g(\boldsymbol{w}_t; \mathcal{D}_t) := \lambda \boldsymbol{w}_t - \frac{1}{|\mathcal{D}_t|} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}_t^+} y \boldsymbol{x} \quad \in \partial \tilde{f}(\boldsymbol{w}; \mathcal{D}_t) \ ,$$

where $\mathcal{D}_t^+ := \{(\boldsymbol{x}, y) \in \mathcal{D}_t : 1 - y(\boldsymbol{w}^T \boldsymbol{x}) > 0\}$.

Update the iterate $\boldsymbol{w}$ by

$$\boldsymbol{w}_{t+1} = \mathbb{P}_{\mathcal{W}} \left(\boldsymbol{w}_t - \eta_t g(\boldsymbol{w}_t; \mathcal{D}_t) \right) \ . \qquad (12)$$

where

$$\eta_t = \frac{1}{\lambda t}, \quad \mathcal{W} := \{\boldsymbol{w} : ||\boldsymbol{w}||_2 \leq \frac{1}{\sqrt{\lambda}}\}, \quad |\mathcal{D}_t| = 1.$$

THE UNIVERSITY of WISCONSIN MADISON

Stochastic Approximation (SA)

Classical **SA methods**

- Choice of $\eta_t = \mathcal{O}(1/t)$ has a history back to [RM51, KW52, Chu54, Sac58].
- Require the objective function to be strongly convex.
  - SVM objective function $f(\cdot)$ is strongly convex with modulus λ.
- Highly sensitive to the scaling of η_t [NJLS09].
- Asymptotic convergence of $\mathcal{O}(1/t)$ in expectation.

Stochastic Approximation (SA)

Classical SA methods

- Choice of $\eta_t = \mathcal{O}(1/t)$ has a history back to [RM51, KW52, Chu54, Sac58].
- Require the objective function to be strongly convex.
  - SVM objective function $f(\cdot)$ is strongly convex with modulus λ.
- Highly sensitive to the scaling of η_t [NJLS09].
- Asymptotic convergence of $\mathcal{O}(1/t)$ in expectation.

Robust SA methods

- Choice of $\eta_t = \mathcal{O}(1/\sqrt{t})$ suggested in [NY83].
- Useful when the objective is convex but not strongly convex, or the curvature is not known.
  - $\lambda \approx 0$ for some choices of C ($\lambda = 1/C$).
- Asymptotic convergence of $\mathcal{O}(1/\sqrt{t})$ in expectation.
- Similar analysis in online learning [Zin03].

THE UNIVERSITY of WISCONSIN MADISON

Stochastic Approximation (SA)

Classical SA methods

- Choice of $\eta_t = \mathcal{O}(1/t)$ has a history back to [RM51, KW52, Chu54, Sac58].
- Require the objective function to be strongly convex.
  - SVM objective function $f(\cdot)$ is strongly convex with modulus λ.
- Highly sensitive to the scaling of η_t [NJLS09].
- Asymptotic convergence of $\mathcal{O}(1/t)$ in expectation.

Robust SA methods

- Choice of $\eta_t = \mathcal{O}(1/\sqrt{t})$ suggested in [NY83].
- Useful when the objective is convex but not strongly convex, or the curvature is not known.
  - $\lambda \approx 0$ for some choices of C ($\lambda = 1/C$).
- Asymptotic convergence of $\mathcal{O}(1/\sqrt{t})$ in expectation.
- Similar analysis in online learning [Zin03].

Both requires a bound on $\mathbb{E}(||g(\boldsymbol{w}; \mathcal{D})||^2)$.

A Stopping Criterion?

- SA algorithms require the number of iterations T to run.
- An efficient stopping criterion is important,
  - Slow convergence of SA methods.
  - Data sets are large.

A Stopping Criterion?

- SA algorithms require the number of iterations T to run.
- An efficient stopping criterion is important,
  - Slow convergence of SA methods.
  - Data sets are large.

Elements of statistical learning theory,

- Unknown $P(X, Y)$, and a dataset $D = \{x_i, y_i\}_{i=1}^{M}$ $i.i.d. \sim P(X, Y)$.

THE UNIVERSITY *of* WISCONSIN MADISON

A Stopping Criterion?

- SA algorithms require the number of iterations T to run.
- An efficient stopping criterion is important,
  - Slow convergence of SA methods.
  - Data sets are large.

Elements of statistical learning theory,

- Unknown $P(X, Y)$, and a dataset $D = \{x_i, y_i\}_{i=1}^{M}$ *i.i.d.* $\sim P(X, Y)$.
- Hypothesis $f \in \mathcal{F}$, $\mathcal{F}$ is a chosen family of hypotheses.

THE UNIVERSITY of WISCONSIN MADISON

A Stopping Criterion?

- SA algorithms require the number of iterations T to run.
- An efficient stopping criterion is important,
  - Slow convergence of SA methods.
  - Data sets are large.

Elements of statistical learning theory,

- Unknown $P(X, Y)$, and a dataset $D = \{x_i, y_i\}_{i=1}^{M}$ *i.i.d.* $\sim P(X, Y)$.
- Hypothesis $f \in \mathcal{F}$, $\mathcal{F}$ is a chosen family of hypotheses.
- Loss function $\ell(f(X), Y)$.

A Stopping Criterion?

- SA algorithms require the number of iterations T to run.
- An efficient stopping criterion is important,
  - Slow convergence of SA methods.
  - Data sets are large.

Elements of statistical learning theory,

- Unknown $P(X, Y)$, and a dataset $D = \{x_i, y_i\}_{i=1}^{M}$ *i.i.d.* $\sim P(X, Y)$.
- Hypothesis $f \in \mathcal{F}$, $\mathcal{F}$ is a chosen family of hypotheses.
- Loss function $\ell(f(X), Y)$.
- Risk $R(f) := \mathbb{E}(\ell(f(X), Y)) = \int \ell(f(X), Y) dP(X, Y)$.

THE UNIVERSITY of WISCONSIN MADISON

A Stopping Criterion?

- SA algorithms require the number of iterations T to run.
- An efficient stopping criterion is important,
  - Slow convergence of SA methods.
  - Data sets are large.

Elements of statistical learning theory,

- Unknown $P(X, Y)$, and a dataset $D = \{x_i, y_i\}_{i=1}^{M}$ $i.i.d. \sim P(X, Y)$.
- Hypothesis $f \in \mathcal{F}$, $\mathcal{F}$ is a chosen family of hypotheses.
- Loss function $\ell(f(X), Y)$.
- Risk $R(f) := \mathbb{E}(\ell(f(X), Y)) = \int \ell(f(X), Y) dP(X, Y)$.
- Empirical Risk $R_{emp}(f) := \frac{1}{M} \sum_{i=1}^{M} \ell(f(x_i), y_i)$.

A Stopping Criterion?

- SA algorithms require the number of iterations T to run.
- An efficient stopping criterion is important,
  - Slow convergence of SA methods.
  - Data sets are large.

Elements of statistical learning theory,

- Unknown $P(X,Y)$, and a dataset $D = \{x_i, y_i\}_{i=1}^{M}$ $i.i.d. \sim P(X,Y)$.
- Hypothesis $f \in \mathcal{F}$, $\mathcal{F}$ is a chosen family of hypotheses.
- Loss function $\ell(f(X), Y)$.
- Risk $R(f) := \mathbb{E}(\ell(f(X), Y)) = \int \ell(f(X), Y) dP(X, Y)$.
- Empirical Risk $R_{emp}(f) := \frac{1}{M} \sum_{i=1}^{M} \ell(f(x_i), y_i)$.
- $R^* := \inf_f R(f)$.

A Stopping Criterion?

- SA algorithms require the number of iterations T to run.
- An efficient stopping criterion is important,
  - Slow convergence of SA methods.
  - Data sets are large.

Elements of statistical learning theory,

- Unknown $P(X, Y)$, and a dataset $D = \{x_i, y_i\}_{i=1}^{M}$ $i.i.d. \sim P(X, Y)$.
- Hypothesis $f \in \mathcal{F}$, $\mathcal{F}$ is a chosen family of hypotheses.
- Loss function $\ell(f(X), Y)$.
- Risk $R(f) := \mathbb{E}(\ell(f(X), Y)) = \int \ell(f(X), Y) dP(X, Y)$.
- Empirical Risk $R_{emp}(f) := \frac{1}{M} \sum_{i=1}^{M} \ell(f(x_i), y_i)$.
- $R^* := \inf_f R(f)$.
- Error decomposition,

$$\underbrace{\inf_{f \in \mathcal{F}} R_{emp}(f) - R^*}_{\text{generalization error}} = \underbrace{\left(\inf_{f \in \mathcal{F}} R(f) - R^* \right)}_{\text{approximation error}} + \underbrace{\left(\inf_{f \in \mathcal{F}} R_{emp}(f) - \inf_{f \in \mathcal{F}} R(f) \right)}_{\text{estimation error}}.$$

THE UNIVERSITY of WISCONSIN MADISON

A Stopping Criterion?

[SSS08] suggested a new error decomposition

$$(\text{gen. err}) = (\text{approx. err}) + (\text{est. err}) + (\text{optimization err}) \; .$$

A Stopping Criterion?

[SSS08] suggested a new error decomposition

$$\text{(gen. err)} = \text{(approx. err)} + \text{(est. err)} + \text{(optimization err)} \quad .$$

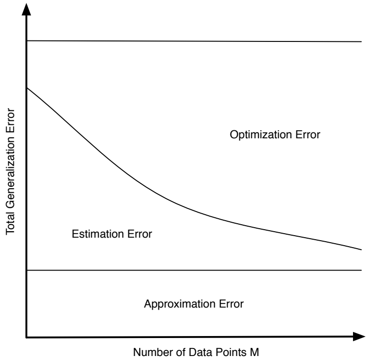
- Approx. error doesn't change for fixed $\mathcal{F}$.
- As $M \to \infty$, (est. err) $\to 0$ if f is consistent.

A Stopping Criterion?

[SSS08] suggested a new error decomposition

$$(\text{gen. err}) = (\text{approx. err}) + (\text{est. err}) + (\text{optimization err}) \ .$$

- Approx. error doesn't change for fixed $\mathcal{F}$.
- As $M \to \infty$, (est. err) $\to 0$ if f is consistent.
- Allow larger opt. err to achieve the same level of gen. err with large M.



[Due to N. Srebro at MLSS09]

Conclusions

Decomposition Algorithm

- Can solve other SVMs, ν-SVM, semiparametric SlapSVM, etc.
- Proofs are on the way.

SA Algorithms

- More work is needed.
- SA methods are inherently serial, each iterate is an instantiation.
  - Reduce the variation of the final iterate distribution, possibly by running several SA algorithms in parallel.
- Nonlinear $\phi(\boldsymbol{x})$ (other than $\phi(\boldsymbol{x}) = \boldsymbol{x}$).
  - Initial work by [JY09].
- Explicit consideration of the intercept b.

Thank you.

Optimality Condition of the Dual Formulation

Lagrangian function $\mathcal{L}$ of (3) and its gradient w.r.t. $\boldsymbol{z}$:

$$\mathcal{L}(\boldsymbol{z};\eta) = F(\boldsymbol{z}) + \eta^T \boldsymbol{A}\boldsymbol{z} \ . \tag{13}$$

$$\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z};\eta) = \boldsymbol{Q}\boldsymbol{z} + \boldsymbol{p} + \boldsymbol{A}^T\eta \ . \tag{14}$$

From Karush-Kuhn-Tucker (KKT) first-order optimality conditions,

$$\left(\boldsymbol{Q}\boldsymbol{z} + \boldsymbol{p} + \boldsymbol{A}^T\eta\right)_i \geq 0 \qquad \text{if } \boldsymbol{z}_i = 0 \tag{15a}$$

$$\left(\boldsymbol{Q}\boldsymbol{z} + \boldsymbol{p} + \boldsymbol{A}^T\eta\right)_i \leq 0 \qquad \text{if } \boldsymbol{z}_i = C \tag{15b}$$

$$\left(\boldsymbol{Q}\boldsymbol{z} + \boldsymbol{p} + \boldsymbol{A}^T\eta\right)_i = 0 \qquad \text{if } \boldsymbol{z}_i \in (0, C/M) \tag{15c}$$

$$\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b} \tag{15d}$$

$$\boldsymbol{0} \leq \boldsymbol{z} \leq (C/M)\boldsymbol{1} \ . \tag{15e}$$

which is necessary and sufficient. Return

Decomposition Framework

Algorithm 1 Decomposition Framework

1. **Initialization.** Choose an initial $\boldsymbol{z}^1$ (3) (possibly infeasible), initial guess of $\boldsymbol{\eta}^1$, positive integers $n_{\mathcal{B}} \geq K$ and $0 < n_c < n_{\mathcal{B}}$, and `tolD`. Choose an initial working set $\mathcal{B}$. $k \leftarrow 1$.

2. **Subproblem.** Solve the subproblem (4) for the current working set $\mathcal{B}$, to obtain $\boldsymbol{z}_{\mathcal{B}}^{k+1}$ and $\boldsymbol{\eta}^{k+1}$. Set $\boldsymbol{z}^{k+1} = (\boldsymbol{z}_{\mathcal{B}}^{k+1}, \boldsymbol{z}_{\mathcal{N}}^{k})$.

3. **Gradient Update.**

$$\nabla F(\boldsymbol{z}^{k+1}) + (\boldsymbol{\eta}^{k+1})^T \boldsymbol{A} = \nabla F(\boldsymbol{z}^k) + \begin{bmatrix} \boldsymbol{Q}_{\mathcal{B}\mathcal{B}} \\ \boldsymbol{Q}_{\mathcal{N}\mathcal{B}} \end{bmatrix} (\boldsymbol{z}_{\mathcal{B}}^{k+1} - \boldsymbol{z}_{\mathcal{B}}^{k}) + (\boldsymbol{\eta}^{k+1})^T \boldsymbol{A} \ .$$

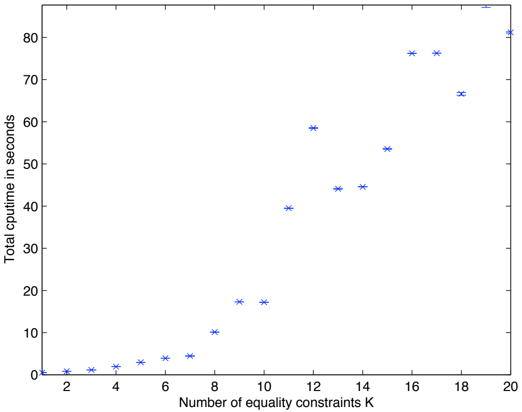
4. **Convergence Check.** If the maximal violation of the KKT conditions falls below `tolD`, terminate with the primal-dual solution $(\boldsymbol{z}^{k+1}, \boldsymbol{\eta}^{k+1})$.

5. **Working Set Update.** Find a new working set $\mathcal{B}$ by solving (6).

6. Set $k \leftarrow k + 1$ and go to step 2.

Return

Scaling of D:PDSG w.r.t K



Return

- Total solution time of D:PDSG with increasing number of parametric components K.
- $M = 1000$.
- Time complexity of D:PDSG is $\mathcal{O}(uKn_{\mathcal{B}})$, u is the number of outer iterations.
- Solver time appears to increase linearly with K for $K \geq 6$.

$$\psi_j(x) = \begin{cases} \cos(j\pi x) & j = 0, 2, 4, \ldots \\ \sin(j\pi x) & j = 1, 3, 5, \ldots \end{cases}$$

THE UNIVERSITY of WISCONSIN MADISON

Reference I

[Bot] Léon Bottou, *http://leon.bottou.org/projects/sgd*.

[Chu54] K.L Chung, *On a stochastic approximation method*, The Annals of Mathematical Statistics **25** (1954), no. 3, 463–483.

[GB84] E. M. Gafni and D. P. Bertsekas, *Two-metric projection methods for constrained optimization*, SIAM Journal on Control and Optimization **22** (1984), 936–964.

[Joa99] Thorsten Joachims, *Making large-scale support vector machine learning practical*, Advances in Kernel Methods - Support Vector Learning (B. Schölkopf, C. Burges, and A. Smola, eds.), MIT Press, Cambridge, MA, 1999, pp. 169–184.

[JY09] T. Joachims and Chun-Nam John Yu, *Sparse kernel svms via cutting-plane training*, Machine Learning (2009), European Conference on Machine Learning (ECML).

[KS05] Wolf Kienzle and Bernhard Schölkopf, *Training support vector machines with multiple equality constraints*, Machine Learning: ECML 2005, vol. 16, October 2005.

[KW52] J. Kiefer and J. Wolfowitz, *Stochastic estimation of the maximum of a regression function*, Annals of Mathematical Statistics **23** (1952), no. 3, 462–466.

[Lin01] C. J. Lin, *Linear convergence of a decomposition method for support vector machines*, Tech. report, Department of Computer Science and Information Engineering, National Taiwan University, 2001.

[LW09] Sangkyun Lee and Stephen J. Wright, *Decomposition algorithms for training large-scale semiparametric support vector machines*, Lecture Notes in Artificial Intelligence: ECML, 2009.

[NJLS09] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, *Robust stochastic approximation approach to stochastic programming*, SIAM Journal on Optimization **19** (2009), no. 4, 1574–1609.

[NY83] A. Nemirovski and D. Yudin, *Problem complexity and method efficiency in optimization*, Wiley-Intersci. Ser. Discrete Math., 1983.

[RM51] Herbert Robbins and Sutton Monro, *A stochastic approximation method*, Annals of Mathematical Statistics **22** (1951), no. 3, 400–407.

Reference II

[Sac58] Jerome Sacks, *Asymptotic distribution of stochastic approximation procedures*, The Annals of Mathematical Statistics **29** (1958), no. 2, 373–405.

[SFS99] Alex J. Smola, Thilo T. Frieß, and Bernhard Schölkopf, *Semiparametric support vector and linear programming machines*, Advances in Neural Information Processing Systems 11 (Cambridge, MA, USA), MIT Press, 1999, pp. 585–591.

[SSS08] S. Shalev-Shwartz and N. Srebro, *Svm optimization: inverse dependence on training set size*, ICML '08: Proceedings of the 25th Annual International Conference on Machine Learning (2008), 928–935.

[SSSS07] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro, *Pegasos: Primal estimated sub-gradient solver for svm*, ICML '07: Proceedings of the 24th international conference on Machine learning (New York, NY, USA), ACM, 2007, pp. 807–814.

[SZ05] T. Serafini and L. Zanni, *On the working set selection in gradient projection-based decomposition techniques for support vector machines*, Optimization Methods and Software **20** (2005), 583–596.

[TY08] P. Tseng and S. Yun, *A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training*, Published online in Computational Optimization and Applications, October 2008.

[Zin03] Martin Zinkevich, *Online convex programming and generalized infinitesimal gradient ascent*, ICML '03: Proceedings of the twentieth international conference on Machine learning, 2003, pp. 928–936.