Decomposition and Stochastic Subgradient Algorithms for Support Vector Machines

Sangkyun Lee and Stephen J. Wright

University of Wisconsin-Madison

ISMP 2009
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,
  
  \[
  \begin{cases}
  \text{Number of variables} = \text{length of an input vector.} \\
  \text{“Primal”} \\
  \text{Obj. consists of a quadratic term and a piecewise linear function.} \\
  \text{Costly obj. function evaluation with many input points.}
  \end{cases}
  \]

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

  \[
  \begin{align*}
  \text{“Primal”} & \quad \text{Number of variables} = \text{length of an input vector.} \\
  & \quad \text{Obj. consists of a quadratic term and a piecewise linear function.} \\
  & \quad \text{Costly obj. function evaluation with many input points.}
  \\
  \text{“Dual”} & \quad \text{Number of variables} = \text{number of input points.} \\
  & \quad \text{QP with dense and ill-conditioned Hessian.} \\
  & \quad \text{A single equality constraint and bound constraints.}
  \end{align*}
  \]
SVMs for Classification (SVC)

\[ h(x) = +1 \]

\[ h(x) = -1 \]

\[ h(x) = 0 \]

\[ \|w\|_2 \]

\[ \mathcal{H} \]

\[ \{(x_i, y_i)\}_{i=1}^M \text{ i.i.d. } \sim P(X, Y), \]

\[ x_i \in \mathbb{R}^N. \]

\[ y_i \in \{-1, +1\}. \]
SVMs for Classification (SVC)

\[ h(x) = +1 \]
\[ h(x) = -1 \]
\[ h(x) = 0 \]
\[ \|w\|_2 \]
\[ H \]

\[ \{(x_i, y_i)\}_{i=1}^M \text{ i.i.d. } \sim \mathcal{P}(X, Y), \]

\[ x_i \in \mathbb{R}^N. \]

\[ y_i \in \{-1, +1\}. \]

\[ \phi : \mathbb{R}^N \longrightarrow H. \]
SVMs for Classification (SVC)

\[
\begin{align*}
    h(x) &= +1 \\
    h(x) &= 0 \\
    h(x) &= -1
\end{align*}
\]

\[
\|w\|_2^2
\]

\[
\mathcal{H}
\]

\[
\text{Maximizing the “margin” } \frac{2}{\|w\|_2}.
\]

\[
\{(x_i, y_i)\}_{i=1}^M \text{ i.i.d. } \sim P(X, Y),
\]

\[
x_i \in \mathbb{R}^N.
\]

\[
y_i \in \{-1, +1\}.
\]

\[
\phi : \mathbb{R}^N \longrightarrow \mathcal{H}.
\]

Find a classifier

\[
h(x) = \langle w, \phi(x) \rangle + b,
\]

\[
h(x_i) \geq +1 \text{ for } y_i = +1,
\]

\[
h(x_i) \leq -1 \text{ for } y_i = -1,
\]

\[
\phi : \mathbb{R}^N \longrightarrow \mathcal{H}.\]
SVMs for Classification (SVC)

\[ h(x) = +1 \]
\[ h(x) = 0 \]
\[ h(x) = -1 \]

\[ 2 / \|w\|_2 \]

\[ \|w\|_2^2 \]

\[ M \]

\[ i.i.d. \sim P(X, Y) \]

\[ x_i \in \mathbb{R}^N \]

\[ y_i \in \{-1, +1\} \]

\[ \phi : \mathbb{R}^N \rightarrow \mathcal{H} \]

Find a classifier
\[ h(x) = \langle w, \phi(x) \rangle + b, \]

\[ h(x_i) \geq +1 \text{ for } y_i = +1, \]

\[ h(x_i) \leq -1 \text{ for } y_i = -1, \]

Maximizing the “margin”
\[ 2 / \|w\|_2. \]

\[ \min_{w, b} \frac{1}{2} \|w\|_2^2 + \frac{C}{M} \sum_{i=1}^{M} \ell_H(h; x_i, y_i), \]

Hinge loss:
\[ \ell_H(h; x_i, y_i) := \max\{1 - y_i h(x_i), 0\}. \]
SVMs for Regression (SVR)

\[ h(x) = \langle w, \phi(x) \rangle + b \]

\( \{ (x_i, y_i) \}_{i=1}^{M} \) i.i.d. \( \sim P(X, Y) \),

- \( x_i \in \mathbb{R}^N \).
- \( y_i \in \mathbb{R} \).

\( \ell_\epsilon(h; x_i, y_i) := \max\{\ |y_i - h(x_i)| - \epsilon, 0 \} \).
SVMs for Regression (SVR)

\[
\begin{align*}
\mathbf{h}(\mathbf{x}) &= \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b \\
\end{align*}
\]

\[
\begin{align*}
&\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^M \text{ i.i.d } \sim P(\mathbf{X}, \mathbf{Y}) , \\
&\mathbf{x}_i \in \mathbb{R}^N. \\
&\mathbf{y}_i \in \mathbb{R}.
\end{align*}
\]

Find a regression function,
\[
\mathbf{h}(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b
\]

- Minimizing prediction error.
- Capture data points in an \(\epsilon\)-radius hyper-tube surrounding \(\mathbf{h}(\mathbf{x})\).
SVMs for Regression (SVR)

\[ h(x) = \langle w, \phi(x) \rangle + b \]

\[ \min_{w,b} \frac{1}{2} \|w\|_2^2 + \frac{C}{M} \sum_{i=1}^{M} \ell_\epsilon(h; x_i, y_i), \]

\[ \ell_\epsilon(h; x_i, y_i) := \max\{|y_i - h(x_i)| - \epsilon, 0\}. \]

\[ \{(x_i, y_i)\}_{i=1}^{M} \text{ i.i.d } \sim P(X, Y), \]

\[ x_i \in \mathbb{R}^N. \]

\[ y_i \in \mathbb{R}. \]

Find a regression function,

\[ h(x) = \langle w, \phi(x) \rangle + b \]

- Minimizing prediction error.
- Capture data points in an \( \epsilon \)-radius hyper-tube surrounding \( h(x) \).
SVM Formulations of Interest

**Primal**

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

where

\[
R_{\text{emp}} = \begin{cases} 
\frac{1}{M} \sum_{i=1}^{M} \ell_H(h; x_i, y_i), & \text{(SVC)} \\
\frac{1}{M} \sum_{i=1}^{M} \ell_\epsilon(h; x_i, y_i), & \text{(SVR)}
\end{cases}
\]

and \( \lambda = 1/C \). The objective function is convex but non-smooth.
SVM Formulations of Interest

**Primal**

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

where

\[
R_{\text{emp}} = \begin{cases} 
\frac{1}{M} \sum_{i=1}^{M} \ell_H(h; x_i, y_i), \ (\text{SVC}) \\
\frac{1}{M} \sum_{i=1}^{M} \ell_\epsilon(h; x_i, y_i), \ (\text{SVR}) 
\end{cases}
\]

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

**Dual**

\[
\min_z \quad \frac{1}{2} z^T Q z + p^T z \\
\text{s.t.} \quad c^T z = d \\
\ell \leq z \leq u,
\]  

(1)

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

- Standard (nonparametric) SVR: use a linear model

\[ h(x) = \langle w, \phi(x) \rangle + b, \]

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

\[ \tilde{h}(x) = \langle w, \phi(x) \rangle + K_{X} \sum_{j=1}^{J} \beta_{j} \psi_{j}(x), \]

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).
Semiparametric SVM

- Standard (nonparametric) SVR: use a linear model

\[ h(x) = \langle w, \phi(x) \rangle + b, \]

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

\[ \tilde{h}(x) = \langle w, \phi(x) \rangle + \sum_{j=1}^{K} \beta_j \psi_j(x), \]

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).
Semiparametric SVM

- Standard (nonparametric) SVR: use a linear model

\[ h(x) = \langle w, \phi(x) \rangle + b , \]

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

\[ \tilde{h}(x) = \langle w, \phi(x) \rangle + \sum_{j=1}^{K} \beta_j \psi_j(x) , \]

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).
Primal Formulation

The “primal” SVR formulation is,

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

Introducing slack variables $\xi_i$ and $\xi_i^*$ to represent the deviations from the $\varepsilon$-tube, we obtain

$$\min_{w,\beta,\xi,\xi^*} \frac{1}{2} w^T w + \frac{C}{M} \sum_{i=1}^{M} (\xi_i + \xi_i^*)$$

(2a)

s.t.  

$$y_i - \langle w, \phi(x_i) \rangle - \sum_{j=1}^{K} \beta_j \psi_j(x_i) \leq \varepsilon + \xi_i \quad \text{for } i = 1, \ldots, M$$

(2b)

$$- \left[ y_i - \langle w, \phi(x_i) \rangle - \sum_{j=1}^{K} \beta_j \psi_j(x_i) \right] \leq \varepsilon + \xi_i^* \quad \text{for } i = 1, \ldots, M$$

(2c)

$$\xi \geq 0, \quad \xi^* \geq 0.$$  

(2d)
Dual Formulation

\[
\min_z F(z) := \frac{1}{2} z^T Q z + p^T z \quad \text{s.t.} \quad A z = 0, \quad 0 \leq z \leq \frac{C}{M} 1, \quad (3)
\]

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

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

\[ p = [\epsilon - y_1, \ldots, \epsilon - y_M, \epsilon + y_1, \ldots, \epsilon + y_M]^T \in \mathbb{R}^{2M}, \]

\[ Q_{ij} = \begin{cases} y_i y_j \kappa(x_i, x_j) & \text{if } 1 \leq i, j \leq M, \text{ or } M + 1 \leq i, j \leq 2M \\ -y_i y_j \kappa(x_i, x_j) & \text{otherwise} \end{cases}, \]

\[ A = \begin{bmatrix} \psi_1(x_1) & \cdots & \psi_1(x_M) & -\psi_1(x_1) & \cdots & -\psi_1(x_M) \\ \psi_2(x_1) & \cdots & \psi_2(x_M) & -\psi_2(x_1) & \cdots & -\psi_2(x_M) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \psi_K(x_1) & \cdots & \psi_K(x_M) & -\psi_K(x_1) & \cdots & -\psi_K(x_M) \end{bmatrix} \in \mathbb{R}^{K \times 2M}. \]

This is a generalization of the standard SVM dual problem. \( n := 2M \).
In each outer iteration, we split variables $z$ into

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

- Basic variables $z_B$, $B \subset \{1, 2, \ldots, n\}$.
- Nonbasic variables $z_N$, $N = \{1, 2, \ldots, n\} \setminus B$.
- $B$ is our working set, of which the size $n_B \ll n$. 
Decomposition Framework [LW09]

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

Given $z_k = (z_k^B, z_k^N)$, we solve the subproblem to get $z_{k+1}^B$.

Subproblem

$$
\begin{align*}
\min_{z^B} & \quad f(z^B) := \frac{1}{2} z^T B Q B z^B + (Q^B N z_k^N + p^B)^T z^B \\
\text{s.t.} & \quad A^B z^B = -A^N z_k^N + b, \\
& \quad 0 \leq z^B \leq C_M 1.
\end{align*}
$$

$z_{k+1} \leftarrow (z_{k+1}^B, z_k^N)$. 

Decomposition Framework [LW09]

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

**Subproblem**

\[
\begin{align*}
\min_{z_B} & \quad f(z_B) := \frac{1}{2} z_B^T Q_{BB} z_B + (Q_{BN} z_N^k + p_B)^T z_B \\
\text{s.t.} & \quad A_B z_B = -A_N z_N^k + b, \quad 0 \leq z_B \leq \frac{C}{M} 1.
\end{align*}
\]
Decomposition Framework [LW09]

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

**Subproblem**

$$
\begin{align*}
\min_{z_B} \quad & f(z_B) := \frac{1}{2} z_B^T Q_{BB} z_B + (Q_{BN} z_N^k + p_B)^T z_B \\
\text{s.t.} \quad & A_B z_B = -A_N z_N^k + b, \quad 0 \leq z_B \leq \frac{C}{M} 1.
\end{align*}
$$

- $z^{k+1} \leftarrow (z_B^{k+1}, z_N^k)$. 
Choosing $\beta$: Working Set Selection

- Inspired by the approach of [Joa99], later improved by [SZ05].
  - $n_\beta$: working set size.
  - $n_c$: max. number of “fresh” indices. $n_c \ll n_\beta$. 

Consider Lagrangian relaxation $L$ of the dual formulation (3),

$$L(z; \eta) = F(z) + \eta^T Az.$$  

(5)

Given $(z_k, \eta_k)$, find a solution $d$ of

$$\min d \quad \nabla_z L(z_k; \eta_k)^T d$$

s.t.

$$0 \leq d_i \leq 1 \text{ if } z_{k+1}^i = 0,$$

$$-1 \leq d_i \leq 0 \text{ if } z_{k+1}^i = C/M,$$

$$-1 \leq d_i \leq 1 \text{ if } z_{k+1}^i \in (0, C/M),$$

$\#\{d_i | d_i \neq 0\} \leq n_c$.

(6)

Solved efficiently, $O(n \log n)$. 

Convergence of decomposition + working set selection [Lin01, TY08].
Choosing $\beta$: Working Set Selection

- Inspired by the approach of [Joa99], later improved by [SZ05].
  - $n_\beta$: working set size.
  - $n_c$: max. number of “fresh” indices. $n_c \ll n_\beta$.

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

$$\mathcal{L}(z; \eta) = F(z) + \eta^T Az.$$  \hfill (5)

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

$$\min_d \left( \nabla_z \mathcal{L}(z^k; \eta^k) \right)^T d$$

s.t.

- $0 \leq d_i \leq 1$ if $z_i^{k+1} = 0$,
- $-1 \leq d_i \leq 0$ if $z_i^{k+1} = C/M$,
- $-1 \leq d_i \leq 1$ if $z_i^{k+1} \in (0, C/M)$,
- $\# \{d_i \mid d_i \neq 0 \} \leq n_c$.  \hfill (6)
Choosing $\beta$: Working Set Selection

- Inspired by the approach of [Joa99], later improved by [SZ05].
  - $n_\beta$: working set size.
  - $n_c$: max. number of “fresh” indices. $n_c \ll n_\beta$.

Consider Lagrangian relaxation $\mathcal{L}$ of the dual formulation (3),
\[
\mathcal{L}(\mathbf{z}; \eta) = F(\mathbf{z}) + \eta^T \mathbf{A} \mathbf{z}.
\] (5)

Given $(\mathbf{z}^k, \eta^k)$, find a solution $\mathbf{d}$ of
\[
\min_{\mathbf{d}} \quad \left( \nabla_{\mathbf{z}} \mathcal{L}(\mathbf{z}^k; \eta^k) \right)^T \mathbf{d}
\]
\[
0 \leq d_i \leq 1 \quad \text{if } \mathbf{z}_i^{k+1} = 0,
\]
\[-1 \leq d_i \leq 0 \quad \text{if } \mathbf{z}_i^{k+1} = C/M,
\]
\[-1 \leq d_i \leq 1 \quad \text{if } \mathbf{z}_i^{k+1} \in (0, C/M),
\]
\[
\# \{ \mathbf{d}_i | d_i \neq 0 \} \leq n_c.
\] (6)

- Solved efficiently, $O(n \log n)$.
- Convergence of decomposition + working set selection [Lin01, TY08].
Subproblem: Primal-dual Solver (PDSG)

We consider the following formulation of (4):

$$\max_{\eta} \min_{z_B \in \Omega} \tilde{L}(z_B, \eta),$$

where

$$\tilde{L}(z_B, \eta) := f(z_B) + \eta^T (A_B z_B + A_N z_N^k),$$

$$\Omega = \{z_B \in \mathbb{R}^{n_B} | 0 \leq z_B \leq \frac{C}{M}1\}.$$
Subproblem: Primal-dual Solver (PDSG)

- We consider the following formulation of (4):

\[
\max \min_{\eta \in \Omega} \tilde{\mathcal{L}}(z_B, \eta),
\]

where

\[
\tilde{\mathcal{L}}(z_B, \eta) := f(z_B) + \eta^T (A_B z_B + A_N z_N^k),
\]

\[
\Omega = \{ z_B \in \mathbb{R}^{n_B} | 0 \leq z_B \leq \frac{C}{M} 1 \}.
\]

In each “inner” iteration, update primal and dual variables by,

\[
\begin{cases}
    z_B^{l+1} \leftarrow z_B^l + s(z_B^l, \eta^l) \\
    \eta^{l+1} \leftarrow \eta^l + t(z_B^{l+1}, \eta^l)
\end{cases},
\]

- 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 \( \nabla_\eta \tilde{\mathcal{C}} \), scaled by dual Hessian diagonal [KS05], on a sub-workingset of size 2.
Update

Update primal-dual iterate pair \((\mathbf{z}^{k+1}, \eta^{k+1})\).

- \(\mathbf{z}^{k+1} \leftarrow (\mathbf{z}_B^{k+1}, \mathbf{z}_N^k)\).
- \(\eta^{k+1}\) is provided by the subproblem solver.
Update

- Update primal-dual iterate pair \((z^{k+1}, \eta^{k+1})\).
  - \(z^{k+1} \leftarrow (z^{k+1}_B, z^{k+1}_N)\).
  - \(\eta^{k+1}\) is provided by the subproblem solver.

- “Full gradient” \(\nabla_z \mathcal{L}(z; \eta)\) has to be updated.
  - To check KKT conditions violation.
  - For the next working set selection.
Update

- Update primal-dual iterate pair \((\mathbf{z}^{k+1}, \eta^{k+1})\).
  - \(\mathbf{z}^{k+1} \leftarrow (\mathbf{z}_B^{k+1}, \mathbf{z}_N^k)\).
  - \(\eta^{k+1}\) is provided by the subproblem solver.

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

Update incrementally,

\[
\nabla_{\mathbf{z}} \mathcal{L}(\mathbf{z}^{k+1}, \eta^{k+1}) = \nabla F(\mathbf{z}^{k+1}) + (\eta^{k+1})^T \mathbf{A} \\
= \nabla F(\mathbf{z}^k) + \begin{bmatrix} Q_{BB} & Q_{NB} \end{bmatrix} (\mathbf{z}_B^{k+1} - \mathbf{z}_B^k) + (\eta^{k+1})^T \mathbf{A}.
\]

(9)
Experiments

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

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

- Sample \( y_i \)'s from \( \omega \) 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]:
  \[ \omega(x) = \sin(x) + \text{sinc}(2\pi(x - 5)). \]

- Sample \( y_i \)'s from \( \omega \) 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: \( \psi_1(x) = \sin(x), \psi_2(x) = \text{sinc}(2\pi(x - 5)). \)
  - Gaussian kernel \( \kappa(x, y) = \exp(-\gamma||x - y||^2) \) with \( \gamma = 0.25 \).
  - Loss function parameter \( \epsilon = 0.05 \).
  - \( n_B = 500, n_c = n_B / 5 \).
Experiments

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

- Sample \( y_i \)'s from \( \omega \) 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: \( \psi_1(x) = \sin(x), \psi_2(x) = \text{sinc}(2\pi(x - 5)) \).
  - Gaussian kernel \( \kappa(x, y) = \exp(-\gamma||x - y||^2) \) with \( \gamma = 0.25 \).
  - Loss function parameter \( \epsilon = 0.05 \).
  - \( n_B = 500, n_c = n_B/5 \).

- 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.
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 $M$ grows: the full gradient update step becomes dominant as $M$ grows.
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.
Recent ML research on solving the primal formulation\textsuperscript{1},

\[
\min_{w,b} \ f(w, D) = \frac{\lambda}{2} w^T w + \frac{1}{M} \sum_{i=1}^{M} \ell_H(w; x_i, y_i). \tag{11}
\]

A large dataset \(D := \{(x_i, y_i) : i = 1, \ldots, M\}\).
Recent ML research on solving the primal formulation\(^1\),

\[
\min_{w,b} f(w, D) = \frac{\lambda}{2} w^T w + \frac{1}{M} \sum_{i=1}^{M} \ell_H(w; x_i, y_i). \quad (11)
\]

- A large dataset \(D := \{(x_i, y_i) : i = 1, \ldots, M\}\).
- The objective function is strongly convex\(^*\).

\(^1\),\(^*\); \(w \leftarrow (w, b)\)
Recent ML research on solving the primal formulation\(^1\),

\[
\min_{w,b} f(w, D) = \frac{\lambda}{2} w^T w + \frac{1}{M} \sum_{i=1}^{M} \ell_H(w; x_i, y_i). \tag{11}
\]

- A large dataset \(D := \{(x_i, y_i) : i = 1, \ldots, M\}\).
- 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.

\[^{1,*}; w \leftarrow (w, b)\]
Recent ML research on solving the primal formulation\(^1\),

\[
\min_{w,b} f(w, D) = \frac{\lambda}{2} w^T w + \frac{1}{M} \sum_{i=1}^{M} \ell_H(w; x_i, y_i).
\]  

\(^1\),\(^*\); \(w \leftarrow (w, b)\)

A large dataset \(D := \{(x_i, y_i) : i = 1, \ldots, M\}\).

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.
Large-scale Linear SVM Training [Bot, SSSS07]

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

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

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

where $\mathcal{D}_t^+ := \{(\mathbf{x}, y) \in \mathcal{D}_t : 1 - y(\mathbf{w}^T \mathbf{x}) > 0\}$. 
Given $\mathcal{D}$, consider the subgradient of an approximate objective function $\tilde{f}(\mathbf{w}; \mathcal{D}_t)$ of $f(\mathbf{w}; \mathcal{D})$ in (11) for a sample dataset $\mathcal{D}_t \subseteq \mathcal{D}$:

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

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

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

Update the iterate $\mathbf{w}$ by

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

where

$$
\eta_t = \frac{1}{\lambda t}, \quad \mathcal{W} := \{\mathbf{w} : \|\mathbf{w}\|_2 \leq \frac{1}{\sqrt{\lambda}}\}, \quad |\mathcal{D}_t| = 1.
$$
Stochastic Approximation (SA)

Classical SA methods

- Choice of $\eta_t = O(1/t)$ 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 $\lambda$.
- Highly sensitive to the scaling of $\eta_t$ [NJLS09].
- Asymptotic convergence of $O(1/t)$ in expectation.
Stochastic Approximation (SA)

**Classical SA methods**
- Choice of $\eta_t = O(1/t)$ 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 $\lambda$.
- Highly sensitive to the scaling of $\eta_t$ [NJLS09].
- Asymptotic convergence of $O(1/t)$ in expectation.

**Robust SA methods**
- Choice of $\eta_t = O(1/\sqrt{t})$ 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 $O(1/\sqrt{t})$ in expectation.
- Similar analysis in online learning [Zin03].
Stochastic Approximation (SA)

**Classical SA methods**
- Choice of $\eta_t = O(1/t)$ 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 $\lambda$.
- Highly sensitive to the scaling of $\eta_t$ [NJLS09].
- Asymptotic convergence of $O(1/t)$ in expectation.

**Robust SA methods**
- Choice of $\eta_t = O(1/\sqrt{t})$ 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 $O(1/\sqrt{t})$ in expectation.
- Similar analysis in online learning [Zin03].

Both requires a bound on $\mathbb{E}(\|g(w; D)\|^2)$. 
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. $\sim P(X, Y)$. 

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.
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 $R(f) := \mathbb{E}(\ell(f(X), Y)) = \int \ell(f(X), Y)dP(X, Y)$.
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 $R_{\text{emp}}(f) := \frac{1}{M} \sum_{i=1}^M \ell(f(x_i), y_i)$.
- $R^* := \inf_f R(f)$. 
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 $R_{\text{emp}}(f) := \frac{1}{M} \sum_{i=1}^M \ell(f(x_i), y_i)$.
- $R^* := \inf_f R(f)$.

Error decomposition,

$$
\inf_{f \in \mathcal{F}} R_{\text{emp}}(f) - R^* = \left( \inf_{f \in \mathcal{F}} R(f) - R^* \right) + \left( \inf_{f \in \mathcal{F}} R_{\text{emp}}(f) - \inf_{f \in \mathcal{F}} R(f) \right).
$$

- Generalization error
- Approximation error
- Estimation error
[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

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

- Approx. error doesn’t change for fixed $\mathcal{F}$.
- As $M \to \infty$, (est. err) → 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 \(M \to \infty\), \((\text{est. err}) \to 0\) 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, \( \nu \)-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 \( \phi(x) \) (other than \( \phi(x) = x \)).
  - 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. $z$:

$$\mathcal{L}(z; \eta) = F(z) + \eta^T Az \ .$$  \hspace{1cm} (13)

$$\nabla_z \mathcal{L}(z; \eta) = Qz + p + A^T \eta \ .$$  \hspace{1cm} (14)

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

$$\left( Qz + p + A^T \eta \right)_i \geq 0$$ if $z_i = 0$  \hspace{1cm} (15a)

$$\left( Qz + p + A^T \eta \right)_i \leq 0$$ if $z_i = C$  \hspace{1cm} (15b)

$$\left( Qz + p + A^T \eta \right)_i = 0$$ if $z_i \in (0, C/M)$  \hspace{1cm} (15c)

$$Az = b$$  \hspace{1cm} (15d)

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

which is necessary and sufficient.
Algorithm 1 Decomposition Framework

1. **Initialization.** Choose an initial $\mathbf{z}^1$ (3) (possibly infeasible), initial guess of $\eta^1$, positive integers $n_B \geq K$ and $0 < n_c < n_B$, and $\text{tolD}$. 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 $\mathbf{z}^{k+1}_B$ and $\eta^{k+1}$. Set $\mathbf{z}^{k+1} = (\mathbf{z}^{k+1}_B, \mathbf{z}^k_N)$.

3. **Gradient Update.**

\[
\nabla F(\mathbf{z}^{k+1}) + (\eta^{k+1})^T \mathbf{A} = \nabla F(\mathbf{z}^k) + \begin{bmatrix}
Q_{BB} \\
Q_{NB}
\end{bmatrix} (\mathbf{z}^{k+1}_B - \mathbf{z}^k_B) + (\eta^{k+1})^T \mathbf{A}.
\]

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

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.
Scaling of D: PDSG w.r.t $K$

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

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


Reference II


