Lecture 12: Online solvers for Convex Risk Minimization

Vojtěch Franc

May 14, 2015

10.A: Sequential Dual Ascent

- Generic framework
- Steepest feasible ascent variable selection strategy
- Near maximal gain variable selection strategy

10.B: Stochastic Gradient Descent

XEP33SML – Structured Model Learning, Summer 2015

10.A: Optimization zoo

Batch methods like the cutting plane algorithms (previous lecture, 9.A, 9.B)

- Updates: expensive after processing all training examples.
+ Stopping condition: certificate of optimality.
+ Tunable memory requirements.

Primal on-line methods like the Stochastic Gradient Descent (this lecture, 10.B)

+ Very simple algorithm.
+ Updates: cheap after processing a single example.
+ Small memory requirements.
- Stopping condition: missing.
- Step-size: requires sensitive setting.

Sequential Dual Ascent (this lecture, 10.A)

+ Updates: cheap after processing a single example.
+ Stopping condition: certificate of optimality.
+ Step-size: optimal can computed analytically.
- Memory requirements.

10.A: SDA - formulation of the primal problem

- Consider learning of a linear classifier $h(\boldsymbol{x};\boldsymbol{w}) = \operatorname{argmax}_{\boldsymbol{y}\in\mathcal{Y}}\langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x},\boldsymbol{y})\rangle$ from a training set $\{(\boldsymbol{x}^1,\boldsymbol{y}^1),\ldots,(\boldsymbol{x}^m,\boldsymbol{y}^m)\} \in (\mathcal{X}\times\mathcal{Y})^m$ by solving the SO-SVM problem

$$\boldsymbol{w}^* = \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^n} \left[\frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m} \max_{\boldsymbol{y}\in\mathcal{Y}} \left(\ell(\boldsymbol{y}^i,\boldsymbol{y}) + \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}^i,\boldsymbol{y}) - \boldsymbol{\Psi}(\boldsymbol{x}^i,\boldsymbol{y}^i)\rangle\right)\right]$$

  where $\lambda > 0$ is a reg. constant, $\boldsymbol{\Psi}\colon \mathcal{X}\times\mathcal{Y}\to\mathbb{R}^n$ is a feature map and $\ell\colon \mathcal{Y}^2\to\mathbb{R}_+$ is a loss such that $\ell(\boldsymbol{y},\boldsymbol{y}') = 0$ iff $\boldsymbol{y} = \boldsymbol{y}'$.

- We will use a compact formulation of the same problem:

$$\boldsymbol{w}^* = \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^n} P(\boldsymbol{w}) \quad \text{where} \quad P(\boldsymbol{w}) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{m}\sum_{i\in\mathcal{I}} \max_{\boldsymbol{y}\in\mathcal{Y}} \left(\ell_i(\boldsymbol{y}) + \langle \boldsymbol{w}, \boldsymbol{\Psi}_i(\boldsymbol{y})\right)$$

  where $\mathcal{I} = \{1,\ldots,m\}$, $\boldsymbol{\Psi}_i(\boldsymbol{y}) = \boldsymbol{\Psi}(\boldsymbol{x}^i,\boldsymbol{y}) - \boldsymbol{\Psi}(\boldsymbol{x}^i,\boldsymbol{y}^i)$, $\ell_i(\boldsymbol{y}) = \ell(\boldsymbol{y}^i,\boldsymbol{y})$ and λ.

- The formulations for other proxies like the slack-rescaling or ''PosLearn'' are similar.

10.A: SDA - dual problem

- The SO-SVM dual problem reads

$$\boldsymbol{\alpha}^* = \operatorname*{argmin}_{\boldsymbol{\alpha} \in \mathbb{R}^d} D(\boldsymbol{\alpha}) \quad \text{where} \quad D(\boldsymbol{\alpha}) = \langle \boldsymbol{b}, \boldsymbol{\alpha} \rangle - \frac{1}{2} \|\mathbf{A}\boldsymbol{\alpha}\|^2$$

where

$\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1; \ldots; \boldsymbol{\alpha}_m)^d$ is a vector of $d = m|\mathcal{Y}|$ dual variables where $\boldsymbol{\alpha}_i = (\alpha_i(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathcal{Y}) \in \mathbb{R}^{|\mathcal{Y}|}$.

$\boldsymbol{b} = (\boldsymbol{b}_1; \ldots; \boldsymbol{b}_m) \in \mathbb{R}^d$ is a vector of losses where $\boldsymbol{b}_i = (\ell_i(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathcal{Y}) \in \mathbb{R}^{|\mathcal{Y}|}$.

$\mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_m) \in \mathbb{R}^{n \times d}$ is a matrix of features where $\mathbf{A}_i = (\boldsymbol{\Psi}_i(\boldsymbol{y})/\sqrt{\lambda} \mid \boldsymbol{y} \in \mathcal{Y}) \in \mathbb{R}^{n \times |\mathcal{Y}|}$.

$\mathcal{A} = \{\boldsymbol{\alpha} \in \mathbb{R}^d \mid \boldsymbol{\alpha} \geq \mathbf{0} \ \wedge \ \sum_{\boldsymbol{y} \in \mathcal{Y}} \alpha_i(\boldsymbol{y}) = \frac{1}{m}, i \in \mathcal{I}\}$ is feasibility set.

- The optimal primal variables can be computed from the dual variables by

$$\boldsymbol{w}^* = -\frac{1}{\sqrt{\lambda}} \mathbf{A}\boldsymbol{\alpha}^*$$

- Thanks to the weak duality we have $P(\boldsymbol{w}) \geq D(\boldsymbol{\alpha}), \forall \boldsymbol{\alpha} \in \mathcal{A}$, and thanks to the strong duality $P(\boldsymbol{w}^*) = D(\boldsymbol{\alpha}^*)$.

10.A: SDA - generic algorithm

- Solving the SO-SVM dual can be converted into a series of simpler (reduced) QP tasks which can be solved analytically.
- The generic SDA: starting from $\boldsymbol{\alpha}^0 \in \mathcal{A}$ it recursively solves a reduced problem

$$\boldsymbol{\alpha}^{t+1} = \operatorname*{argmax}_{\boldsymbol{\alpha} \in \mathcal{A}^t} D(\boldsymbol{\alpha})$$

where $\mathcal{A}^t \subset \mathcal{A}$ is a line between the current solution $\boldsymbol{\alpha}^t$ and a point $\boldsymbol{\beta}^t$ selected from $\mathcal{A}$, i.e.

$$\mathcal{A}^t = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^d \mid \boldsymbol{\alpha} = (1-\tau)\boldsymbol{\alpha}^t + \tau\boldsymbol{\beta}^t, \tau \in [0,1] \right\}$$

- All iterates of the SDA algorithm are feasible, i.e. $\boldsymbol{\alpha}^t \in \mathcal{A}$, $\forall t$.
- The instances of the SDA algorithm differ in the strategy used to construct the point $\boldsymbol{\beta}^t$:
  - In order to queary the oracle for the i-th example we will modify only those dual variables associated with i-th example: $\beta_j^t(y) = \alpha_j^t(y)$, $\forall j \in \mathcal{I} \setminus \{i\}$, $\forall y \in \mathcal{Y}$.
  - **Steepest feasible ascent:** maximize the derivative $D'_{\mathcal{A}^t}(0)$ where $D_{\mathcal{A}^t}(\tau) = D(\boldsymbol{\alpha}^t(1-\tau)\boldsymbol{\alpha}^t + \tau\boldsymbol{\beta}^t)$
  - **Near maximal gain:** maximize the improvement $\delta^t = D(\boldsymbol{\alpha}^{t+1}) - D(\boldsymbol{\alpha}^t)$

10.A: SDA - solving the reduced problem analytically

- The dual objective $D(\boldsymbol{\alpha}) = \langle \boldsymbol{b}, \boldsymbol{\alpha} \rangle - \frac{1}{2}\|\mathbf{A}\boldsymbol{\alpha}\|^2$ restricted to the line segment $\mathcal{A}^t$ reads

$$D_{\mathcal{A}^t}(\tau) = D((1-\tau)\boldsymbol{\alpha}^t + \tau\boldsymbol{\beta}^t) = D(\boldsymbol{\alpha}^t) + \tau\langle \boldsymbol{\beta}^t - \boldsymbol{\alpha}^t, \boldsymbol{b} - \mathbf{A}^T\mathbf{A}\boldsymbol{\alpha}^t \rangle - \frac{\tau^2}{2}\|\mathbf{A}(\boldsymbol{\beta}^t - \boldsymbol{\alpha}^t)\|^2$$

  and its derivative is

$$D'_{\mathcal{A}^t}(\tau) = \langle \boldsymbol{\beta}^t - \boldsymbol{\alpha}^t, \boldsymbol{b} - \mathbf{A}^T\mathbf{A}\boldsymbol{\alpha}^t \rangle - \tau\|\mathbf{A}(\boldsymbol{\beta}^t - \boldsymbol{\alpha}^t)\|^2$$

- The reduced problem $\boldsymbol{\alpha}^{t+1} = \operatorname{argmax}_{\boldsymbol{\alpha} \in \mathcal{A}^t} D(\boldsymbol{\alpha})$ can be solved by computing

$$\tau^t = \operatorname*{argmax}_{\tau \in [0,1]} D(\boldsymbol{\alpha}^t(1-\tau) + \boldsymbol{\beta}^t)$$

  and setting $\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t(1-\tau^t) + \boldsymbol{\beta}^t\tau^t$.

- Provided $D'_{\mathcal{A}^t}(0) > 0$, the optimal τ^t is either 1 or a solution of $D'_{\mathcal{A}^t}(\tau) = 0$, i.e.

$$\tau^t = \min\left\{1, \frac{\langle \boldsymbol{\beta}^t - \boldsymbol{\alpha}^t, \boldsymbol{b} - \mathbf{A}^T\mathbf{A}\boldsymbol{\alpha}^t \rangle}{\|\mathbf{A}(\boldsymbol{\beta}^t - \boldsymbol{\alpha}^t)\|^2}\right\}$$

10.A: SDA - The stopping condition inspired variable selection strategy

- Let $\boldsymbol{w}^t = -\frac{1}{\sqrt{\lambda}}\mathbf{A}\boldsymbol{\alpha}^t$ be a primal solution. Then the duality gap reads:

$$
\begin{aligned}
&G(\boldsymbol{w}^t, \boldsymbol{\alpha}^t) = P(\boldsymbol{w}^t) - D(\boldsymbol{\alpha}^t) \\
&= \frac{\lambda}{2}\|\boldsymbol{w}^t\|^2 + \frac{1}{m}\sum_{i\in\mathcal{I}} \max_{\boldsymbol{y}\in\mathcal{Y}} \big(\ell_i(\boldsymbol{y}) + \langle \boldsymbol{w}^t, \boldsymbol{\Psi}_i(\boldsymbol{y})\rangle\big) - \sum_{i\in\mathcal{I}}\sum_{\boldsymbol{y}\in\mathcal{Y}} \alpha_i(\boldsymbol{y})\ell_i(\boldsymbol{y}) + \frac{1}{2}\|\mathbf{A}\boldsymbol{\alpha}^t\|^2 \\
&= \frac{1}{m}\sum_{i\in\mathcal{I}} \max_{\boldsymbol{y}\in\mathcal{Y}} \big(\ell_i(\boldsymbol{y}) + \langle \boldsymbol{w}^t, \boldsymbol{\Psi}_i(\boldsymbol{y})\rangle\big) - \sum_{i\in\mathcal{I}}\sum_{\boldsymbol{y}\in\mathcal{Y}} \alpha_i^t(\boldsymbol{y})\ell_i(\boldsymbol{y}) + \lambda\|\boldsymbol{w}^t\|^2 \\
&= \frac{1}{m}\sum_{i\in\mathcal{I}} \max_{\boldsymbol{y}\in\mathcal{Y}} \big(\ell_i(\boldsymbol{y}) + \langle \boldsymbol{w}^t, \boldsymbol{\Psi}_i(\boldsymbol{y})\rangle\big) - \sum_{i\in\mathcal{I}}\sum_{\boldsymbol{y}\in\mathcal{Y}} \alpha_i^t(\boldsymbol{y})\big(\ell_i(\boldsymbol{y}) + \langle \boldsymbol{\Psi}_i(\boldsymbol{y}), \boldsymbol{w}^t\rangle\big) \\
&= \frac{1}{m}\Big[\sum_{i\in\mathcal{I}} \max_{\boldsymbol{y}\in\mathcal{Y}} s_i(\boldsymbol{y}, \boldsymbol{w}^t) - m\sum_{\boldsymbol{y}\in\mathcal{Y}} \alpha_i^t(\boldsymbol{y}) s_i(\boldsymbol{y}, \boldsymbol{w}^t)\Big] = \frac{1}{m}\sum_{i\in\mathcal{I}} G_i(\boldsymbol{w}^t, \boldsymbol{\alpha}^t)
\end{aligned}
$$

  where $s_i(\boldsymbol{y}, \boldsymbol{w}) = \ell_i(\boldsymbol{y}) + \langle \boldsymbol{w}, \boldsymbol{\Psi}_i(\boldsymbol{y})\rangle$ is the loss augmented classification score.

- Strategy to select the block of variables $\boldsymbol{\alpha}_i^t = (\alpha_i^t(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathcal{Y})$ to be updated: go sequentially over all examples $i \in \mathcal{I}$ and if

$$G_i(\boldsymbol{w}^t, \boldsymbol{\alpha}^t) > \varepsilon$$

  holds then update $\boldsymbol{\alpha}_i^t$. Otherwise, if $G_i(\boldsymbol{w}^t, \boldsymbol{\alpha}^t) \leq \varepsilon$, $\forall i \in \mathcal{I}$, then $P(\boldsymbol{w}^t) \leq P(\boldsymbol{w}^*) + \varepsilon$.

10.A: SDA - steepest feasible ascent

- Assume we are going to update the i-th block of variables so that $\boldsymbol{\beta}^t$ must be from $\mathcal{A} \cap \mathcal{B}_i$ where $\mathcal{B}_i^t = \{\boldsymbol{\beta} \in \mathbb{R}^d \mid \beta_j^t(\boldsymbol{y}) = \alpha_j^t(\boldsymbol{y}), \forall j \in \mathcal{I} \setminus \{i\}, \forall \boldsymbol{y} \in \mathcal{Y}\}$
- The point $\boldsymbol{\beta}^t$ can be constructed such that

$$\begin{aligned}\boldsymbol{\beta}^t &= \operatorname*{argmax}_{\boldsymbol{\beta} \in \mathcal{A} \cap \mathcal{B}_i^t} D'_{\mathcal{A}^t}(0) = \operatorname*{argmax}_{\boldsymbol{\beta} \in \mathcal{A} \cap \mathcal{B}_i^t} \langle \boldsymbol{\beta} - \boldsymbol{\alpha}^t, \boldsymbol{b} - \mathbf{A}^T \mathbf{A} \boldsymbol{\alpha}^t \rangle \\ &= \operatorname*{argmax}_{\substack{\boldsymbol{\beta}_i \geq 0 \\ \sum_{\boldsymbol{y} \in \mathcal{Y}} \beta_i(\boldsymbol{y}) = \frac{1}{m}}} \sum_{\boldsymbol{y} \in \mathcal{Y}} \beta_i(\boldsymbol{y}) \Big(\ell_j(\boldsymbol{y}) + \langle \boldsymbol{w}^t, \boldsymbol{\Psi}_j(\boldsymbol{y}) \rangle \Big)\end{aligned}$$

  where $\boldsymbol{w}^t = -\frac{1}{\sqrt{\lambda}} \mathbf{A} \boldsymbol{\alpha}^t$.
- The solution is the vector $\boldsymbol{\beta}^t = (\beta_j(\boldsymbol{y}) \mid j \in \mathcal{I}, \boldsymbol{y} \in \mathcal{Y})$ with components

$$\beta_j^t(\boldsymbol{y}) = \begin{cases} \frac{1}{m} & \text{if} \quad \boldsymbol{y} = \boldsymbol{u}^t \wedge i = j \\ 0 & \text{if} \quad \boldsymbol{y} \neq \boldsymbol{u}^t \wedge i = j \\ \alpha_j^t(\boldsymbol{y}) & \text{if} \quad j \neq i \end{cases}$$

  where

$$\boldsymbol{u}^t \in \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \Big(\ell_i(\boldsymbol{y}) + \langle \boldsymbol{w}^t, \boldsymbol{\Psi}_i(\boldsymbol{y}) \rangle \Big)$$

  which is the loss augmented classification problem.

10.A: Algorithm: SDA with steepest feasible ascent (the dual form)

Input: precision parameter $\varepsilon > 0$

Output: ε-optimal primal-dual pair $(\boldsymbol{w}^t, \boldsymbol{\alpha}^t)$

Initialization:

$$\alpha_i^0(\boldsymbol{y}) = \begin{cases} \frac{1}{m} & \text{if} \quad \boldsymbol{y} = \boldsymbol{y}^i \\ 0 & \text{otherwise} \end{cases}, \; i \in \mathcal{I}$$

repeat

num_updates $:= 0$

forall $i \in \mathcal{I}$ **do**

$\boldsymbol{w}^t = -\frac{1}{\sqrt{\lambda}} \mathbf{A}\boldsymbol{\alpha}^t$

$\boldsymbol{u}^t \in \operatorname{argmax}_{\boldsymbol{y} \in \mathcal{Y}} s_i(\boldsymbol{y}, \boldsymbol{w}^t)$

if $s_i(\boldsymbol{u}^t, \boldsymbol{w}^t) - m \sum_{\boldsymbol{y} \in \mathcal{Y}} \alpha_i(\boldsymbol{y}) s_i(\boldsymbol{y}, \boldsymbol{w}^t) > \varepsilon$ **then**

num_updates $:=$ num_updates $+ 1$

$$\beta_j^t(\boldsymbol{y}) = \begin{cases} \frac{1}{m} & \text{if} \quad \boldsymbol{y} = \boldsymbol{u}^t \wedge i = j \\ 0 & \text{if} \quad \boldsymbol{y} \neq \boldsymbol{u}^t \wedge i = j \\ \alpha_j^t(\boldsymbol{y}) & \text{if} \quad j \neq i \end{cases}$$

$$\tau^t = \min \left\{ 1, \frac{\langle \boldsymbol{\beta}^t - \boldsymbol{\alpha}^t, \boldsymbol{b} - \mathbf{A}^T \mathbf{A} \boldsymbol{\alpha}^t \rangle}{\| \mathbf{A}(\boldsymbol{\beta}^t - \boldsymbol{\alpha}^t) \|^2} \right\}$$

$\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t (1 - \tau^t) + \boldsymbol{\beta}^t \tau^t$

$t \leftarrow t + 1$

until num_updates $= 0$;

10.A: SDA - steepest feasible ascent; removing the dual variables

- The key quantities can be computed without explicitly maintaining the dual variables:

$$\begin{aligned}
\boldsymbol{w}^t = -\frac{1}{\sqrt{\lambda}}\mathbf{A}\boldsymbol{\alpha}^t &= \sum_{i\in\mathcal{I}}\Big(-\frac{1}{\sqrt{\lambda}}\mathbf{A}_i\boldsymbol{\alpha}_i^t\Big) = \sum_{i\in\mathcal{I}}\boldsymbol{w}_i^t \\
\langle\boldsymbol{\beta}^t - \boldsymbol{\alpha}^t, \boldsymbol{b} - \mathbf{A}^T\mathbf{A}\boldsymbol{\alpha}^t\rangle &= \frac{1}{m}s_i(\boldsymbol{u}^t, \boldsymbol{w}^t) - L_i^t + \lambda\langle\boldsymbol{w}^t, \boldsymbol{w}_i^t\rangle \quad \text{where} \quad L_i^t = \langle\boldsymbol{\alpha}_i^t, \boldsymbol{b}_i\rangle \\
\|\mathbf{A}(\boldsymbol{\beta}^t - \boldsymbol{\alpha}^t)\|^2 &= \lambda\left\|\frac{1}{m\lambda}\boldsymbol{\Psi}_i(\boldsymbol{u}^t) + \boldsymbol{w}_i^t\right\|^2
\end{aligned}$$

- Starting from $\boldsymbol{w}^0 = \boldsymbol{w}_1^0 = \cdots = \boldsymbol{w}_m^0 = \mathbf{0}$, $L_1^0 = \cdots = L_m^0 = 0$, the key quantities can be directly updated:

$$\begin{aligned}
L_i^{t+1} &= \langle\boldsymbol{b}_i, \boldsymbol{\alpha}_i^t(1-\tau^t) + \tau^t\boldsymbol{\beta}_i^t\rangle = L_i^t(1-\tau^t) + \frac{\tau^t}{m}\ell_i(\boldsymbol{u}^t) \\
\boldsymbol{w}_i^{t+1} &= -\frac{1}{\sqrt{\lambda}}\mathbf{A}\big(\boldsymbol{\alpha}_i^t(1-\tau^t) + \boldsymbol{\beta}^t\tau^t\big) = \boldsymbol{w}_i^t(1-\tau^t) - \frac{\tau^t}{\lambda m}\boldsymbol{\Psi}_i(\boldsymbol{u}^t) \\
\boldsymbol{w}^{t+1} &= \boldsymbol{w}^t + \boldsymbol{w}_i^{t+1} - \boldsymbol{w}_i^t
\end{aligned}$$

10.A: Algorithm: SDA with steepest feasible ascent (the primal form)

Input: precision parameter $\varepsilon > 0$
Output: ε-optimal primal solution $\boldsymbol{w}^t$
Initialization:
$t := 0$, $\boldsymbol{w}^0 := \mathbf{0}$, $(\boldsymbol{w}_i^0 := \mathbf{0},\ i \in \mathcal{I})$, $(L_i^0 := 0,\ i \in \mathcal{I})$
repeat
  num_updates $:= 0$
  forall $i \in \mathcal{I}$ **do**
    $\boldsymbol{u}^t \in \operatorname{argmax}_{\boldsymbol{y} \in \mathcal{Y}} s_i(\boldsymbol{y}, \boldsymbol{w}^t)$
    if $s_i(\boldsymbol{u}^t, \boldsymbol{w}^t) - mL_i^t + m\lambda\langle \boldsymbol{w}^t, \boldsymbol{w}_i^t \rangle > \varepsilon$ **then**
      num_updates $\leftarrow$ num_updates $+ 1$

$$\tau^t = \min\left\{1, \frac{\frac{1}{m}s_i(\boldsymbol{u}^t, \boldsymbol{w}^t) - L_i^t + \lambda\langle \boldsymbol{w}^t, \boldsymbol{w}_i^t \rangle}{\lambda \left\| \frac{1}{m\lambda}\boldsymbol{\Psi}_i(\boldsymbol{u}^t) + \boldsymbol{w}_i^t \right\|^2}\right\}$$

$$\boldsymbol{w}_i^t = \boldsymbol{w}_i^t(1 - \tau^t) - \frac{\tau^t}{\lambda m}\boldsymbol{\Psi}_i(\boldsymbol{u}^t)$$

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \boldsymbol{w}_i^{t+1} - \boldsymbol{w}_i^t$$

$$L_i^{t+1} = L_i^t(1 - \tau^t) + \frac{\tau^t}{m}\ell_i(\boldsymbol{u}^t)$$

      $t := t + 1$
until num_updates $= 0$;

10.A: SDA - improvement of the dual objective

- Provided $\tau^{t+1} = 1$, the improvement of the objective reads

$$\begin{aligned}
\delta(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^t) &= D(\boldsymbol{\alpha}^{t+1}) - D(\boldsymbol{\alpha}^t) \\
&= D(\boldsymbol{\beta}^t) - D(\boldsymbol{\alpha}^t) \\
&= \langle \boldsymbol{\alpha}^t - \boldsymbol{\beta}^t, \boldsymbol{b} - \mathbf{A}^T\mathbf{A}\boldsymbol{\alpha}^t \rangle - \frac{1}{2}\|\mathbf{A}(\boldsymbol{\alpha}^t - \boldsymbol{\beta}^t)\|^2
\end{aligned}$$

- Provided $\tau^{t+1} < 1$, the improvement of the objective reads

$$\begin{aligned}
\delta(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^t) &= D(\boldsymbol{\alpha}^{t+1}) - D(\boldsymbol{\alpha}^t) \\
&= D(\boldsymbol{\alpha}^t(1 - \tau^{t+1}) + \tau^{t+1}\boldsymbol{\beta}^t) - D(\boldsymbol{\alpha}^t) \\
&= \cdots \\
&= \frac{\langle \boldsymbol{\alpha}^t - \boldsymbol{\beta}^t, \boldsymbol{b} - \mathbf{A}^T\mathbf{A}\boldsymbol{\alpha}^t \rangle^2}{2\|\mathbf{A}(\boldsymbol{\alpha}^t - \boldsymbol{\beta}^t)\|^2}
\end{aligned}$$

10.A: SDA - variable selection based on near maximal gain

- Construct the $\boldsymbol{\beta}^t$ such that only two variables in the i-th block are updated:

$$\beta_j^t(\boldsymbol{y}) = \begin{cases} \alpha_i^t(\boldsymbol{u}^t) + \alpha_i^t(\boldsymbol{v}^t) & \text{if} \quad j = i \wedge \boldsymbol{y} = \boldsymbol{u}^t \\ 0 & \text{if} \quad j = i \wedge \boldsymbol{y} = \boldsymbol{v}^t \\ \alpha_j^t(\boldsymbol{y}^t) & \text{if} \quad j \neq i \end{cases}$$

- If the labels $\boldsymbol{u}^t$ and $\boldsymbol{v}^t$ are selected such that $D'_{\mathcal{A}^t}(0) > 0$, which is equivalent to

$$\alpha_i^t(\boldsymbol{v}^t)(s_i(\boldsymbol{u}^t, \boldsymbol{w}^t) - s_i(\boldsymbol{v}^t, \boldsymbol{w}^t)) > 0\,,$$

then the improvement $\delta^t(i, \boldsymbol{u}, \boldsymbol{v}) = D(\boldsymbol{\alpha}^{t+1}) - D(\boldsymbol{\alpha}^t)$ reads

$$\delta^t(i, \boldsymbol{u}, \boldsymbol{v}) = \begin{cases} \dfrac{\lambda(s_i(\boldsymbol{u}, \boldsymbol{w}^t) - s_i(\boldsymbol{v}, \boldsymbol{w}^t))}{\|\boldsymbol{\Psi}_i(\boldsymbol{u}) - \boldsymbol{\Psi}_i(\boldsymbol{v})\|^2} & \text{if} \quad \tau^{t+1} < 1 \\ \alpha_i^t(\boldsymbol{v})(s_i(\boldsymbol{u}, \boldsymbol{w}^t) - s_i(\boldsymbol{v}, \boldsymbol{w}^t)) - \dfrac{\alpha_i^t(\boldsymbol{v})^2}{2\lambda}\|\boldsymbol{\Psi}_i(\boldsymbol{u}) - \boldsymbol{\Psi}_i(\boldsymbol{v})\|^2 & \text{if} \quad \tau^{t+1} = 1 \end{cases}$$

- Instead of trying all possible $|\mathcal{Y}|^2$ options we can approximate:

$$\boldsymbol{u}^t \in \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} s_i(\boldsymbol{y}, \boldsymbol{w}^t) \qquad \text{and} \qquad \boldsymbol{v}^t \in \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}_i} \delta^t(i, \boldsymbol{u}^t, \boldsymbol{y})$$

where $\mathcal{Y}_i = \{\boldsymbol{y} \in \mathcal{Y} \mid \alpha_i^t(\boldsymbol{y}) > 0\}$ is a set of non-zero dual variables.

10.A: Algorithm: SDA with near maximal gain

Initialization:

$\boldsymbol{w} := \mathbf{0}$, $\mathcal{Y}_i := \{\boldsymbol{y}^i\}$, $\alpha_i(\boldsymbol{y}) := \begin{cases} \frac{1}{m} & \text{if} \quad \boldsymbol{y} = \boldsymbol{y}^i \\ 0 & \text{otherwise} \end{cases}$, $i \in \mathcal{I}$

repeat

  num_updates $:= 0$

  forall $i \in \mathcal{I}$ **do**

    $\boldsymbol{u}_1 := \operatorname{argmax}_{\boldsymbol{y} \in \mathcal{Y}} s_i(\boldsymbol{y}, \boldsymbol{w})$

    $\boldsymbol{u}_2 := \operatorname{argmax}_{\boldsymbol{y} \in \mathcal{Y}_i} s_i(\boldsymbol{y}, \boldsymbol{w})$

    if $s_i(\boldsymbol{u}_1, \boldsymbol{w}) > s_i(\boldsymbol{u}_2, \boldsymbol{w})$ **then**

      $\hat{\boldsymbol{u}} \leftarrow \boldsymbol{u}_1$

    else

      $\hat{\boldsymbol{u}} \leftarrow \boldsymbol{u}_2$

    if $s_i(\hat{\boldsymbol{u}}, \boldsymbol{w}) - m \sum_{\boldsymbol{y} \in \mathcal{Y}_i} \alpha_i(\boldsymbol{y}) s_i(\boldsymbol{y}, \boldsymbol{w}) > \varepsilon$ **then**

      num_updates $:=$ num_updates $+ 1$

      $\hat{\boldsymbol{v}} := \operatorname*{argmax}_{\boldsymbol{y} \in \{\boldsymbol{y}' \in \mathcal{Y}_i \mid \alpha_i(\boldsymbol{y}') > 0\}} \delta(i, \hat{\boldsymbol{u}}, \boldsymbol{y})$

      $\tau := \min \left\{ 1, \frac{\lambda(s_i(\hat{\boldsymbol{u}}, \boldsymbol{w}) - s_i(\hat{\boldsymbol{v}}, \boldsymbol{w}))}{\alpha_i(\hat{\boldsymbol{v}}) \|\boldsymbol{\psi}_i(\hat{\boldsymbol{u}}) - \boldsymbol{\psi}_i(\hat{\boldsymbol{v}})\|^2} \right\}$

      $\boldsymbol{w} := \boldsymbol{w} + (\boldsymbol{\psi}_i(\hat{\boldsymbol{v}}) - \boldsymbol{\psi}_i(\hat{\boldsymbol{u}})) \frac{\tau \alpha_i(\hat{\boldsymbol{v}})}{\lambda}$

      $\alpha_i(\hat{\boldsymbol{u}}) := \alpha_i(\hat{\boldsymbol{u}}) + \tau \alpha_i(\hat{\boldsymbol{v}})$

      $\alpha_i(\hat{\boldsymbol{v}}) := \alpha_i(\hat{\boldsymbol{v}}) - \tau \alpha_i(\hat{\boldsymbol{v}})$

      if $\hat{\boldsymbol{u}} = \boldsymbol{u}_1$ **then**

        $\mathcal{Y}_i := \mathcal{Y}_i \cup \{\boldsymbol{u}_1\}$

until num_updates $= 0$;

10.A: SDA - memory requirements

- SDA with steepest feasible ascent sel. strategy: $\mathcal{O}(m \cdot n)$ where m is the number of examples and n the number of primal parameters.
- SDA with near maximal gain sel. strategy: $\mathcal{O}(n' \cdot T)$ where n' is the number of non-zero elements of $\boldsymbol{\Psi}_i(\boldsymbol{y})$ and T is the total number of updates.

10.A: SDA - convergence guarantees

- The convergence theorem applies for both the SDA with the steepest feasible ascent strategy as well as the near maximal gain strategy.
- **Convergence theorem:** For any $\varepsilon > 0$ and $\lambda > 0$, the SDA algorithm terminates after

$$T = \frac{8\ell_{\max} D^2}{\varepsilon^2 \lambda}$$

updates at most where $\ell_{\max} = \max_{i \in \mathcal{I}, \boldsymbol{y} \in \mathcal{Y}} \ell_i(\boldsymbol{y})$ and $D = \max_{i \in \mathcal{I}, \boldsymbol{y} \in \mathcal{Y}} \|\boldsymbol{\Psi}_i(\boldsymbol{y})\|$.

- Although both variants of the SDA algorithm have the same upper bound on the number of iterations their practical performance is different.

10.A: Experimental comparison

Compared methods

- **CPA** Cutting Plane Algorithm: implements the Bundle Method for Risk Minimization (last lecture, 9.A)
- **BCFW** Block-Coordinate Frank-Wolfe: implements SDA with steepest feasible ascent selection strategy (this lecture, 10.A)
- **SDM** Sequential Dual Method : implements SDA with steepest feasible ascent and changing only two dual variables at time (not covered)
- **FASOLE** Fast Algorithm for Structured Output LEarning: implements SDA with near maximal gain selection strategy (this lecture, 10.A)

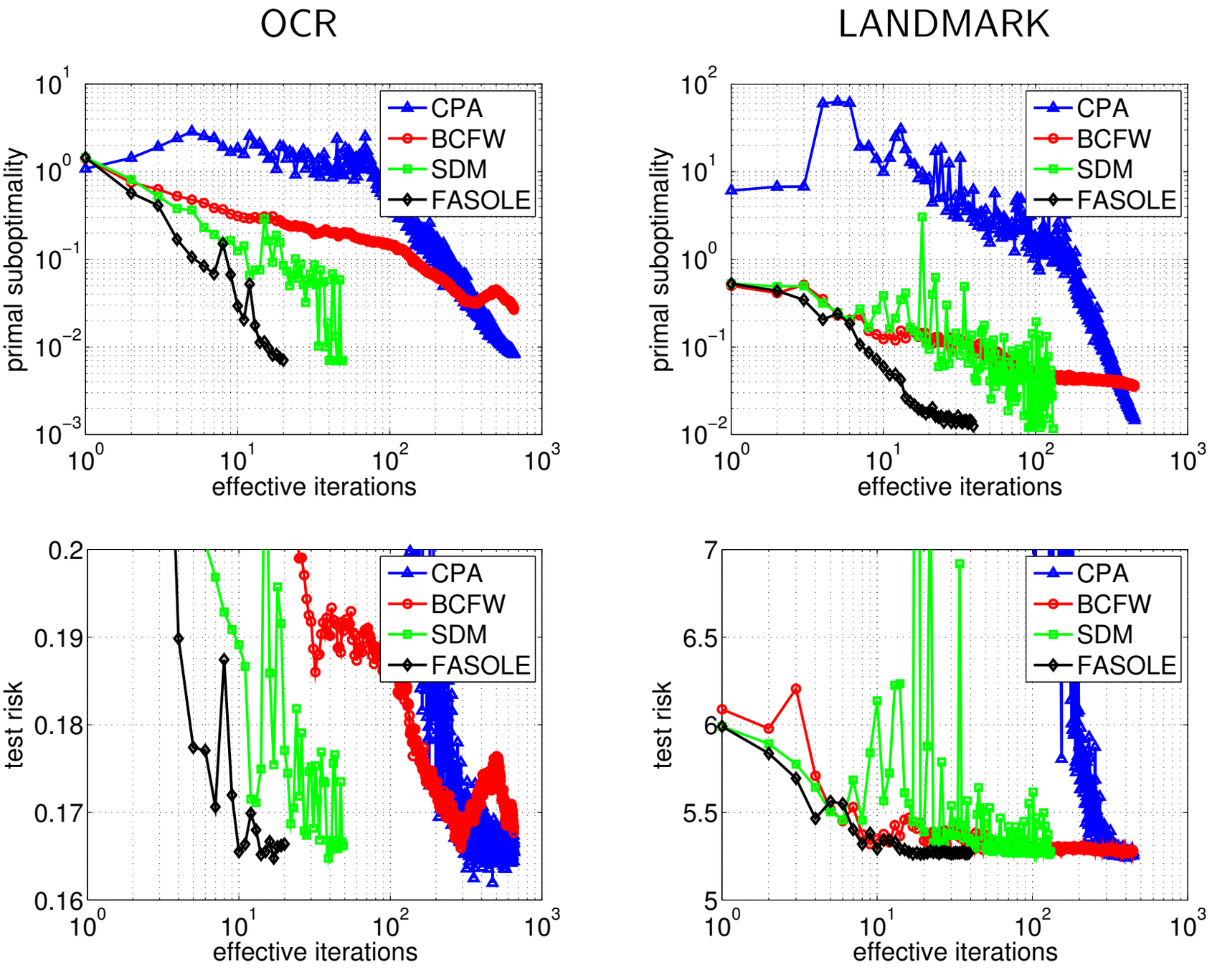
Benchmark data

dataset		#train	#test	#params	max-sum classifier
OCR	[Taskar 2003]	5,512	1,365	4,004	HMM-chain
LANDMARK	[Uricar 2012]	5,062	3,512	232,476	HMM-tree

Evaluation protocol

The solvers run to find the same ε-precise solution for a range of regularization parameters λ. We measure the convergence speed in terms of the objective value and the test risk.

10.A: Experimental comparison



10.B: Stochastic Gradient Descent - formulation

- Let us consider a convex constrained problem

$$\boldsymbol{w}^* \in \operatorname*{argmin}_{\boldsymbol{w}\in\mathcal{W}} F(\boldsymbol{w})$$

where $\mathcal{W} \subset \mathbb{R}^n$ is a closed convex set and $F\colon \mathcal{W} \to \mathbb{R}$ is a convex function.

- The stochastic gradient descent approaches the objective F via an oracle which for given $\boldsymbol{w}^t$ provides a stochastic estimate $\hat{\boldsymbol{g}}^t$ of the sub-gradient $\boldsymbol{g}^t \in \partial F(\boldsymbol{w}^t)$ such that

$$\mathbb{E}\hat{\boldsymbol{g}}^t = \boldsymbol{g}^t$$

- For example, in our setting

$$F(\boldsymbol{w}) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{m}\sum_{i\in\mathcal{I}} \ell_i(\boldsymbol{w}) = \frac{1}{m}\sum_{i\in\mathcal{I}} \left(\frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \ell_i(\boldsymbol{w})\right) = \frac{1}{m}\sum_{i\in\mathcal{I}} F_i(\boldsymbol{w})$$

the oracle can pick $i \in \mathcal{I}$ randomly with the uniform distribution over $\mathcal{I}$ and it provides a sub-gradient

$$\hat{\boldsymbol{g}}^t \in \partial F_i(\boldsymbol{w}^t)$$

10.B: Stochastic Gradient Descent

- The SGD algorithm: starting from $\boldsymbol{w}^1 = \mathbf{0}$, SGD computes new iterates recursively as follows

$$\boldsymbol{w}^{t+1} = \Pi_{\mathcal{W}}\left(\boldsymbol{w}^t - \eta^t \hat{\boldsymbol{g}}^t\right)$$

where $\Pi_{\mathcal{W}} \colon \mathbb{R}^n \to \mathcal{W}$ denotes projection on $\mathcal{W}$, i.e.

$$\Pi_{\mathcal{W}}(\boldsymbol{w}) = \operatorname*{argmin}_{\boldsymbol{w}' \in \mathcal{W}} \|\boldsymbol{w}' - \boldsymbol{w}\|$$

and η^t is a sequence of step-sizes.

- The theoretical results require a fixed step size, typically, $\sum_{t=0}^{\infty} \eta^t = \infty$ and $\lim_{t\to\infty} \eta^t = 0$.
- There is no stopping condition which would provide a certificate of optimality. In practice, SGD is stopped based on monitoring a progress of the objective or a validation error.

10.B: Stochastic Gradient Descent - convergence guarantees

- Recall that a function $F\colon \mathcal{W} \to \mathbb{R}$ is λ-strictly convex iff the function $F(\boldsymbol{w}) - \frac{\lambda}{2}\|\boldsymbol{w}\|^2$ is convex.

  For example, $F(\boldsymbol{w}) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + R(\boldsymbol{w})$ is λ-strictly convex iff $R(\boldsymbol{w})$ is convex.

- **λ-strictly convex functions:** Suppose F is λ-strictly convex, and that $\mathbb{E}[\|\hat{\boldsymbol{g}}^t\|^2] \leq G^2$, $\forall t$. Consider SGD with step sizes $\eta^t = \frac{1}{\lambda t}$. Then for any $t > 1$, it holds that

$$\mathbb{E}[F(\boldsymbol{w}^t) - F(\boldsymbol{w}^*)] \leq \frac{17G^2(1+\log(t))}{\lambda t}$$

- **Convex functions:** Assume that F is convex and that for some constants D, G it holds that $\mathbb{E}[\|\hat{\boldsymbol{g}}^t\|] \leq G$, $\forall t$, and $\sup_{\boldsymbol{w},\boldsymbol{w}' \in \mathcal{W}} \|\boldsymbol{w} - \boldsymbol{w}'\| \leq D$. Consider SGD with step size $\eta^t = \frac{c}{\sqrt{t}}$ where $c > 0$ is a constant. Then for any $t > 1$ it holds that

$$\mathbb{E}[F(\boldsymbol{w}^t) - F(\boldsymbol{w}^*)] \leq \left(\frac{D^2}{c} + cG^2\right)\frac{2+\log(t)}{\sqrt{t}}$$

10.B: Stochastic Gradient Descent with averaging

- Instead of using the last iterate of the SGD, take average from the first till the last iteration

$$\bar{\boldsymbol{w}}^t = \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{w}^i$$

which can be computed iteratively by setting $\bar{\boldsymbol{w}}^1 = \boldsymbol{w}^1$ and update recursively

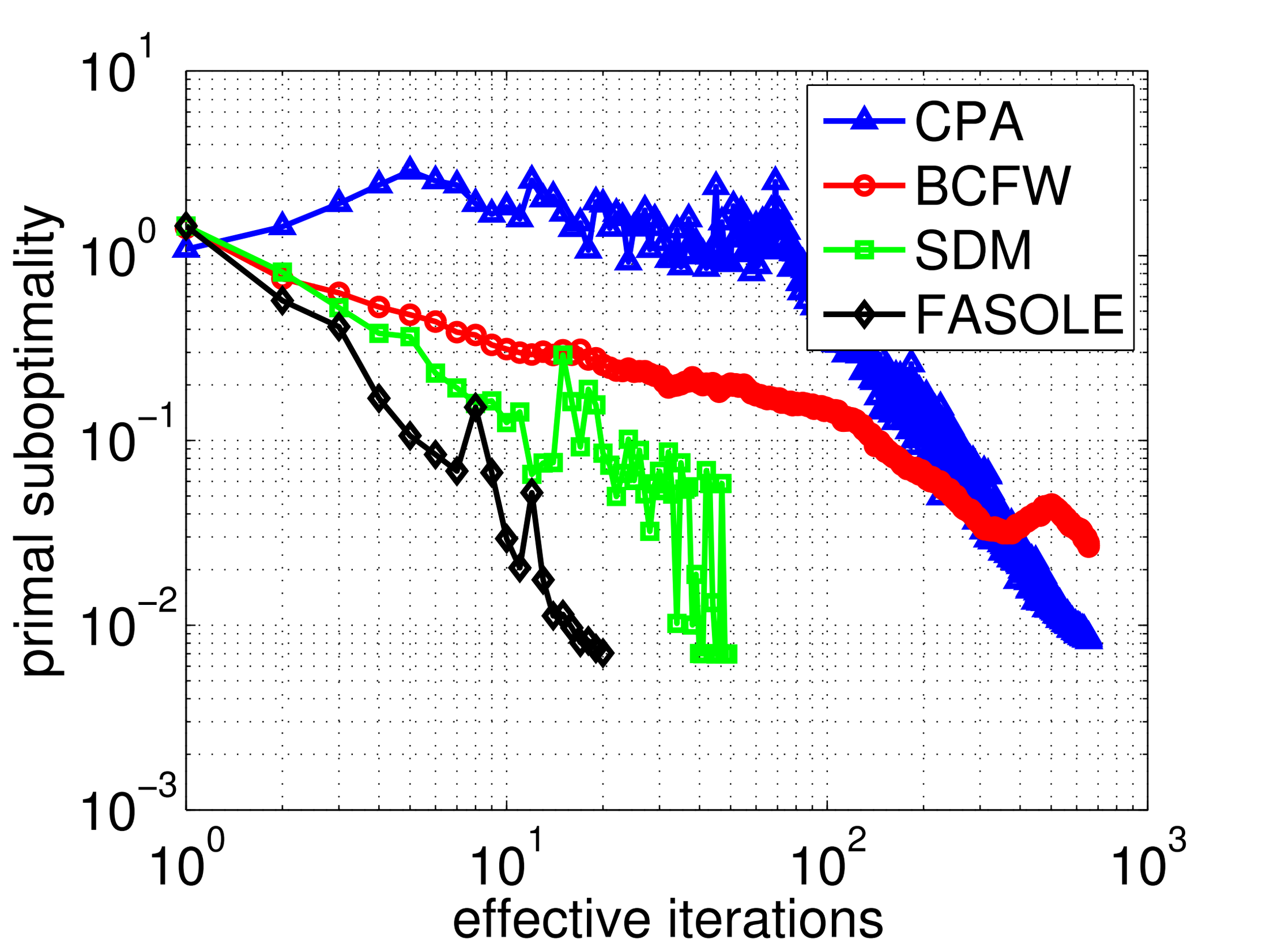
$$\bar{\boldsymbol{w}}^t = \left(1 - \frac{1}{t}\right)\bar{\boldsymbol{w}}^{t-1} + \frac{1}{t}\boldsymbol{w}^t$$

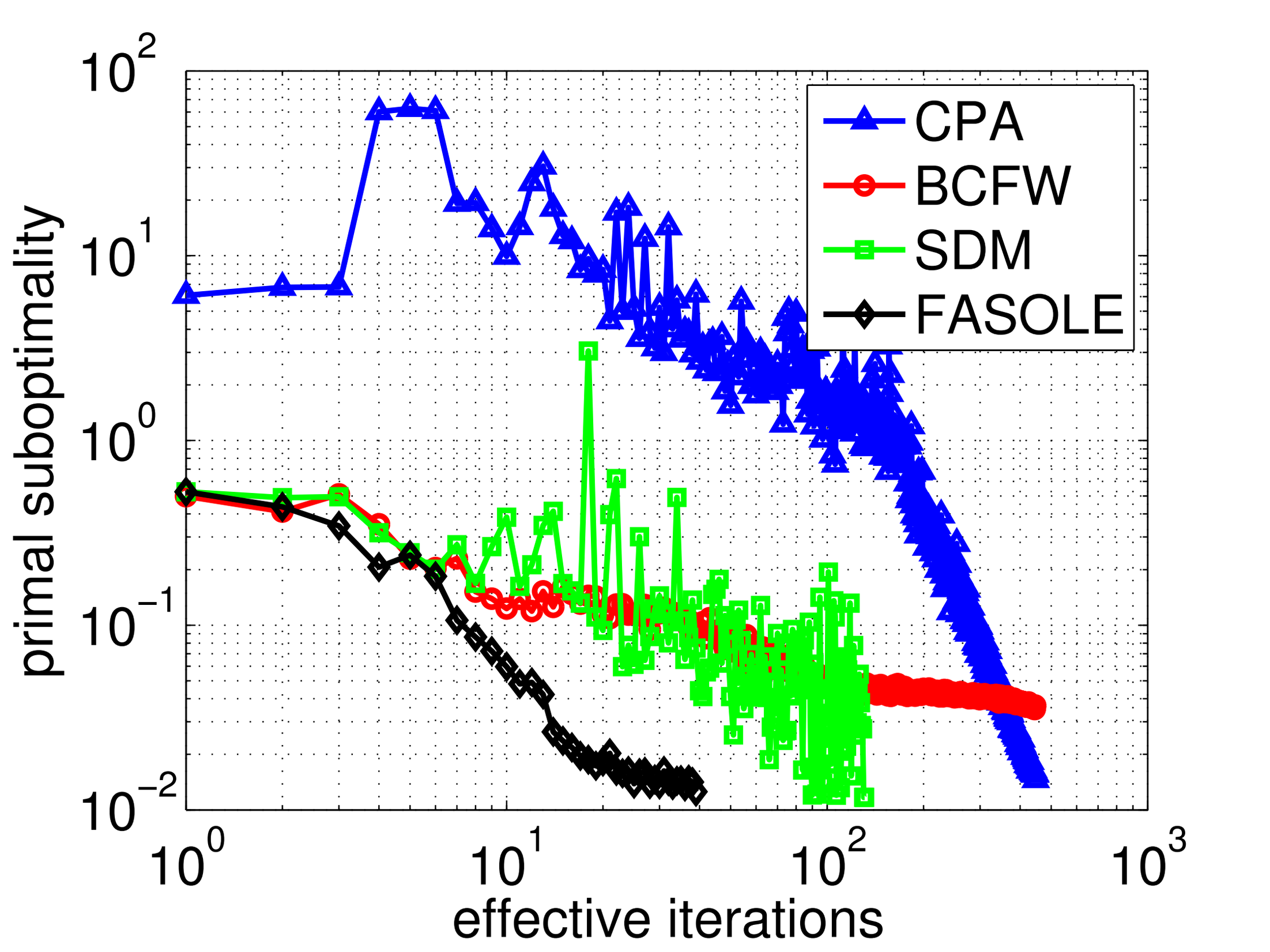
- Better results can be obtained with the polynomial decay averaging parametrized by ν:

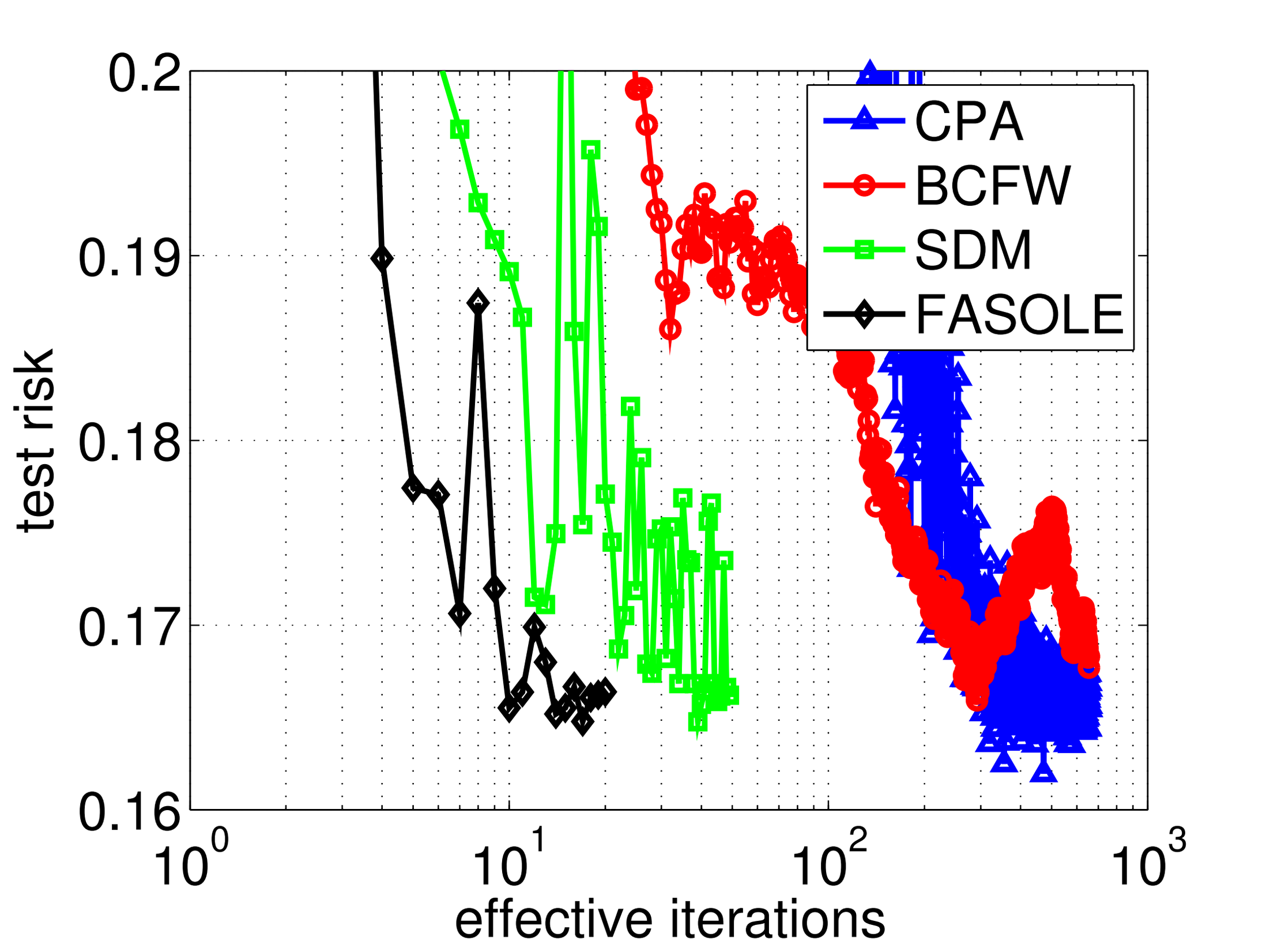
$$\bar{\boldsymbol{w}}_\nu^t = \left(1 - \frac{\nu+1}{t+\nu}\right)\bar{\boldsymbol{w}}_\nu^{t-1} + \frac{\nu+1}{t+\nu}\boldsymbol{w}^t$$

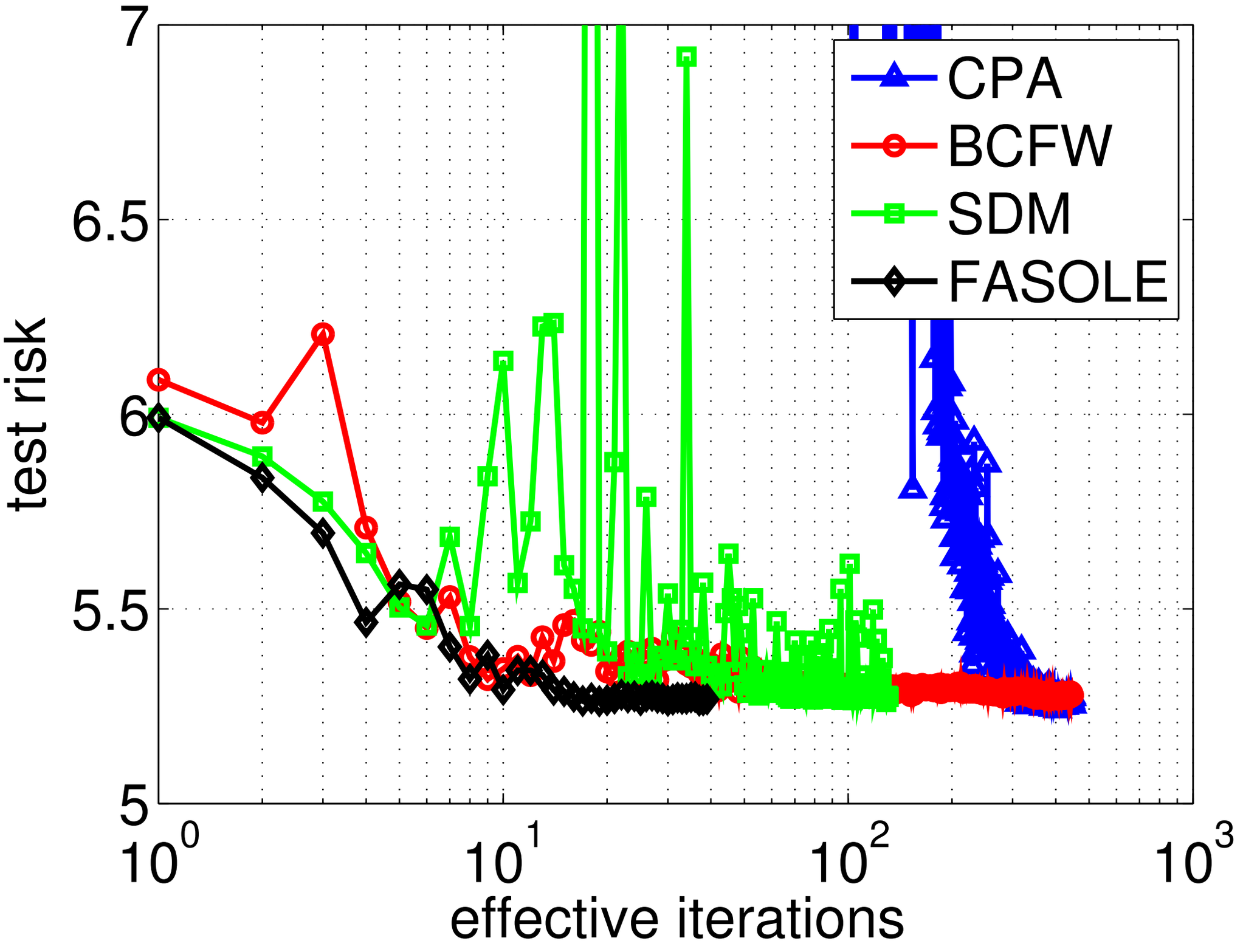
which for $\nu = 0$ reduces to the standard averaging while $\nu > 0$ increases the weight of the last iterate.

- The averaging schemes provide convergence guarantees even for a fixed step-size.


CPA
BCFW
SDM
FASOLE
primal suboptimality
effective iterations
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{0}
10^{1}
10^{2}
10^{3}


primal suboptimality
effective iterations
CPA
BCFW
SDM
FASOLE
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{0}
10^{1}
10^{2}
10^{3}


0.2
0.19
0.18
0.17
0.16
test risk
10^0
10^1
10^2
10^3
effective iterations
CPA
BCFW
SDM
FASOLE


7
6.5
6
5.5
5
test risk
10^0
10^1
10^2
10^3
effective iterations
CPA
BCFW
SDM
FASOLE