Adaptive Optimization Methods for Machine Learning and Signal Processing

Part IV/IV: Adaptivity in min-max optimization

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)
Setup: min-max optimization

\[
\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \Phi(x, y)
\]

- \( \Phi(\cdot, y) \) is convex for all \( y \)
- \( \Phi(x, \cdot) \) is concave for all \( x \)
- \( \mathcal{X}, \mathcal{Y} \) are closed, convex sets
- Solution set is nonempty: \( \exists x^*, y^* : \)

\[
\Phi(x^*, y^*) = \min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \Phi(x, y)
\]
Setup: min-max optimization

- $\Phi(\cdot, y)$ is convex for all $y$
- $\Phi(x, \cdot)$ is concave for all $x$
- $\mathcal{X}, \mathcal{Y}$ are closed, convex sets
- Solution set is nonempty: $\exists x^*, y^*$:
  \[
  \Phi(x^*, y^*) = \min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \Phi(x, y)
  \]

- The solution is the saddle point
## Classic methods

<table>
<thead>
<tr>
<th>Adaptivity to</th>
<th>Lipschitz constant</th>
<th>Noise</th>
<th>Strong convexity</th>
<th>Sparsity of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Bach and Levy, 2019]</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>[Chambolle et al., 2018]</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>[Alacaoglu et al., 2020]</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Gradient Descent-ascent

\[
\min_{x \in X} \max_{y \in Y} \Phi(x, y)
\]

**Algorithm** Simultaneous GDA (Forward-Backward) [Sibony, 1970]

\[
\begin{align*}
&\text{for } t = 0 \text{ to } T - 1 \text{ do} \\
&\quad x_{t+1} = P_X(x_t - \eta \nabla_x \Phi(x_t, y_t)) \\
&\quad y_{t+1} = P_Y(y_t + \eta \nabla_y \Phi(x_t, y_t)) \\
&\text{end for}
\end{align*}
\]

**Algorithm** Alternating GDA (Arrow-Hurwicz) [Arrow et al., 1958]

\[
\begin{align*}
&\text{for } t = 0 \text{ to } T - 1 \text{ do} \\
&\quad x_{t+1} = P_X(x_t - \eta \nabla_x \Phi(x_t, y_t)) \\
&\quad y_{t+1} = P_Y(y_t + \eta \nabla_y \Phi(x_{t+1}, y_t)) \\
&\text{end for}
\end{align*}
\]

Theorem. [Gidel et al., 2018] With any \( \eta > 0 \), iterates of simGDA diverge:

\[
x^2_{t+1} + y^2_{t+1} = (1 + \eta^2)(x^2_t + y^2_t)
\]

Theorem. [Gidel et al., 2018] With any \( \eta > 0 \), iterates of altGDA do not converge:

\[
x^2_{t+1} + y^2_{t+1} = \Theta(x^2_0 + y^2_0)
\]
Gradient Descent-ascent

\[
\min_{x \in X} \max_{y \in Y} \Phi(x, y)
\]

**Algorithm** Simultaneous GDA (Forward-Backward) [Sibony, 1970]

\[
\begin{align*}
\text{for } & t = 0 \text{ to } T - 1 \text{ do} \\
& x_{t+1} = P_X(x_t - \eta \nabla_x \Phi(x_t, y_t)) \\
& y_{t+1} = P_Y(y_t + \eta \nabla_y \Phi(x_t, y_t)) \\
\text{end for}
\end{align*}
\]

- Behavior of GDA on the toy problem:

\[
\min_{x \in \mathbb{R}} \max_{y \in \mathbb{R}} xy
\]

**Theorem.** [Gidel et al., 2018] With any \( \eta > 0 \), iterates of simGDA diverge:

\[
x_{t+1}^2 + y_{t+1}^2 = (1 + \eta^2)(x_t^2 + y_t^2)
\]

**Algorithm** Alternating GDA (Arrow-Hurwicz) [Arrow et al., 1958]

\[
\begin{align*}
\text{for } & t = 0 \text{ to } T - 1 \text{ do} \\
& x_{t+1} = P_X(x_t - \eta \nabla_x \Phi(x_t, y_t)) \\
& y_{t+1} = P_Y(y_t + \eta \nabla_y \Phi(x_t+1, y_t)) \\
\text{end for}
\end{align*}
\]

**Theorem.** [Gidel et al., 2018] With any \( \eta > 0 \), iterates of altGDA do not converge:

\[
x_{t+1}^2 + y_{t+1}^2 = \Theta(x_0^2 + y_0^2)
\]
Toy problem: in practice

\[
\min_{x \in \mathbb{R}} \max_{y \in \mathbb{R}} \ xy
\]

- Simultaneous GDA
- Alternating GDA
Extragradient

\[
\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \Phi(x, y)
\]

○ Notation for convenience: \( z = (x, y), \) \( \mathcal{Z} = \mathcal{X} \times \mathcal{Y}, \) \( F(z) = (\nabla_x \Phi(x, y), -\nabla_y \Phi(x, y)) \).

Algorithm Simultaneous GDA (Forward-backward)

\begin{verbatim}
for t = 0 to T - 1 do
    z_{t+1} = P_{\mathcal{Z}}(z_t - \eta F(z_t))
end for
\end{verbatim}

○ Note the equivalence to

\[
x_{t+1} = P_{\mathcal{X}}(x_t - \eta \nabla_x \Phi(x_t, y_t)) \\
y_{t+1} = P_{\mathcal{Y}}(y_t + \eta \nabla_y \Phi(x_t, y_t))
\]
Extragradient

\[
\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \Phi(x, y)
\]

- Notation for convenience: \( z = (x, y), \ Z = \mathcal{X} \times \mathcal{Y}, \ F(z) = (\nabla_x \Phi(x, y), -\nabla_y \Phi(x, y)) \).

**Algorithm** Simultaneous GDA (Forward-backward)

```plaintext
for \( t = 0 \) to \( T - 1 \) do
  \( z_{t+1} = P_Z(z_t - \eta F(z_t)) \)
end for
```

- Note the equivalence to

\[
\begin{align*}
x_{t+1} &= P_{\mathcal{X}}(x_t - \eta \nabla_x \Phi(x_t, y_t)) \\
y_{t+1} &= P_{\mathcal{Y}}(y_t + \eta \nabla_y \Phi(x_t, y_t))
\end{align*}
\]

**Algorithm** Extragradient [Korpelevich, 1976]

```plaintext
for \( t = 0 \) to \( T - 1 \) do
  \( z_{t+1/2} = P_Z(z_t - \eta F(z_t)) \)
  \( z_{t+1} = P_Z(z_t - \eta F(z_{t+1/2})) \)
end for
```

Diagram:
- extra-gradient step
- gradient step
Toy problem: in practice

\[
\min_{x \in \mathbb{R}} \max_{y \in \mathbb{R}} x y
\]
Convergence of Extragradient

\[ \min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \Phi(x, y) \]

○ Notation for convenience: \( z = (x, y), \ Z = \mathcal{X} \times \mathcal{Y}, \ F(z) = (\nabla_x \Phi(x, y), -\nabla_y \Phi(x, y)) \).

Algorithm (Stochastic) Extragradient\(^1\)

\[
\text{for } t = 0 \text{ to } T - 1 \text{ do}
\]
\[
z_{t+1/2} = P_Z(z_t - \eta_t \tilde{F}(z_t))
\]
\[
z_{t+1} = P_Z(z_t - \eta_t \tilde{F}(z_{t+1/2}))
\]
\[\text{end for}\]

Assumptions.

○ \( \Phi(x, y) \) is convex-concave

○ A solution \( z^* \) exists

○ Lipschitzness: \( \|F(u) - F(v)\| \leq L\|u - v\| \)

○ Unbiasedness: \( \mathbb{E}[\tilde{F}(z)] = F(z) \)

○ Variance bound: \( \mathbb{E}\|\tilde{F}(z) - F(z)\|^2 \leq R^2 \)

\(^1\)Same ideas also apply when we use Bregman distances in the update rule. This version is known as Mirror-Prox.

\(^2\)Merit function for the rate is primal-dual gap: \( \text{err}(\bar{z}) = \max_{x, y} \Phi(\bar{x}, y) - \Phi(x, \bar{y}) \).
Convergence of Extragradient

\[
\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \Phi(x, y)
\]

- Notation for convenience: \( z = (x, y), \ Z = \mathcal{X} \times \mathcal{Y}, \ F(z) = (\nabla_x \Phi(x, y), -\nabla_y \Phi(x, y)) \).

Algorithm (Stochastic) Extragradient\(^1\)

\[
\begin{align*}
&\text{for } t = 0 \text{ to } T - 1 \text{ do} \\
&\quad z_{t+1/2} = P_Z(z_t - \eta_t \tilde{F}(z_t)) \\
&\quad z_{t+1} = P_Z(z_t - \eta_t \tilde{F}(z_{t+1/2})) \\
&\end{align*}
\]

end for


\( \text{EG w/ } \tilde{F} = F, \eta_t = \eta < 1/L \):

- Convergence: \( z_{t+1} \rightarrow z^* \)
- Rate\(^2\): \( \text{err} \left( \frac{1}{T} \sum_{t=1}^{T} z_{t+1/2} \right) \leq O \left( \frac{1}{T} \right) \)

Assumptions.

- \( \Phi(x, y) \) is convex-concave
- A solution \( z^* \) exists
- Lipschitzness: \( \| F(u) - F(v) \| \leq L \| u - v \| \)
- Unbiasedness: \( \mathbb{E}[\tilde{F}(z)] = F(z) \)
- Variance bound: \( \mathbb{E} \| \tilde{F}(z) - F(z) \|^2 \leq R^2 \)

\( ^1 \)Same ideas also apply when we use Bregman distances in the update rule. This version is known as Mirror-Prox.

\( ^2 \)Merit function for the rate is primal-dual gap: \( \text{err}(\bar{z}) = \max_{x,y} \Phi(\bar{x}, y) - \Phi(x, \bar{y}) \).
Convergence of Extragradient

\[
\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \Phi(x, y)
\]

- Notation for convenience: \( z = (x, y) \), \( \mathcal{Z} = \mathcal{X} \times \mathcal{Y} \), \( F(z) = (\nabla_x \Phi(x, y), -\nabla_y \Phi(x, y)) \).

**Algorithm (Stochastic) Extragradient**

\[
\begin{align*}
& \text{for } t = 0 \text{ to } T - 1 \text{ do} \\
& \quad z_{t+1/2} = P_{Z}(z_t - \eta_t \tilde{F}(z_t)) \\
& \quad z_{t+1} = P_{Z}(z_t - \eta_t \tilde{F}(z_{t+1/2})) \\
& \text{end for}
\end{align*}
\]

**Theorem (Deterministic).** [Korpelevich, 1976, Nemirovski, 2004]

EG w/ \( \tilde{F} = F \), \( \eta_t = \eta < 1/L \):

- Convergence: \( z_{t+1} \to z^* \)
- Rate\(^2\): \( \text{err} \left( \frac{1}{T} \sum_{t=1}^{T} z_{t+1/2} \right) \leq O \left( \frac{1}{T} \right) \)

**Theorem (Stochastic).** [Juditsky et al., 2011]

EG w/ noisy oracles \( \mathbb{E}[\tilde{F}(z)] = F(z) \), \( \eta_t = \frac{\eta_0}{\sqrt{t}} \):

- Rate: \( \mathbb{E} \text{ err} \left( \frac{1}{T} \sum_{t=1}^{T} z_{t+1/2} \right) \leq O \left( \frac{1}{\sqrt{T}} \right) \)

Assumptions.

- \( \Phi(x, y) \) is convex-concave
- A solution \( z^* \) exists
- Lipschitzness: \( \|F(u) - F(v)\| \leq L\|u - v\| \)
- Unbiasedness: \( \mathbb{E}[\tilde{F}(z)] = F(z) \)
- Variance bound: \( \mathbb{E}\|\tilde{F}(z) - F(z)\|^2 \leq R^2 \)

\(^1\)Same ideas also apply when we use Bregman distances in the update rule. This version is known as Mirror-Prox.

\(^2\)Merit function for the rate is primal-dual gap: \( \text{err}(\bar{z}) = \max_{x, y} \Phi(\bar{x}, y) - \Phi(x, \bar{y}) \).
Takeaway

**Algorithm** (Stochastic) Extragradient

\[ \text{for } t = 0 \text{ to } T - 1 \text{ do} \]
\[ z_{t+1/2} = P_Z(z_t - \eta_t \tilde{F}(z_t)) \]
\[ z_{t+1} = P_Z(z_t - \eta_t \tilde{F}(z_{t+1/2})) \]
\[ \text{end for} \]

→ Different step sizes for deterministic & stochastic: \(1/L\) vs \(\eta_0/\sqrt{t}\)

→ Need to know \(L\) to set step size
<table>
<thead>
<tr>
<th>Adativity to</th>
<th>Lipschitz constant</th>
<th>noise</th>
<th>strong convexity</th>
<th>sparsity of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Bach and Levy, 2019]</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>[Chambolle et al., 2018]</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>[Alacaoglu et al., 2020]</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Adaptivity to smoothness and noise

○ \( F(z) = (\nabla_x \Phi(x, y), -\nabla_y \Phi(x, y)) = \mathbb{E}[\tilde{F}(z)] \)

Algorithm (Stochastic) Adaptive extragradient

\[
\text{for } t = 0 \text{ to } T - 1 \text{ do}
\begin{align*}
  z_{t+1/2} &= P_Z(z_t - \eta_t \tilde{F}(z_t)) \\
  z_{t+1} &= P_Z(z_t - \eta_t \tilde{F}(z_{t+1/2}))
\end{align*}
\text{end for}
\]

○ Run EG w/ adaptive step size

[Bach and Levy, 2019]

\[(\max_{x,y} \|x - y\| \leq D, G_0 > 0) , \]

\[
\eta_t = \frac{D}{\sqrt{G_0^2 + \sum_{i=0}^{t-1} Z_i^2}} , \quad (1)
\]

where, \( Z_i^2 = \frac{\|z_{i+1} - z_{i+1/2}\|^2 + \|z_{i+1/2} - z_i\|^2}{5\eta_i^2} \).
Adaptivity to smoothness and noise

\( F(z) = (\nabla_x \Phi(x,y), -\nabla_y \Phi(x,y)) = \mathbb{E}[\tilde{F}(z)] \)

**Algorithm** (Stochastic) Adaptive extragradient

```
for t = 0 to T - 1 do
    z_{t+1/2} = P_Z(z_t - \eta_t \tilde{F}(z_t))
    z_{t+1} = P_Z(z_t - \eta_t \tilde{F}(z_{t+1/2}))
end for
```

- Run EG w/ adaptive step size
  [Bach and Levy, 2019]

\( \max_{x,y} \|x - y\| \leq D, G_0 > 0 \),

\[
\eta_t = \frac{D}{\sqrt{G_0^2 + \sum_{i=0}^{t-1} Z_i^2}},
\]

where, \( Z_i^2 = \frac{\|z_{i+1} - z_{i+1/2}\|^2 + \|z_{i+1/2} - z_i\|^2}{5\eta_i^2} \).

Intuition. Recall AdaGrad step size for \( \min_x f(x) \):

\[
\eta_t = \frac{D}{\sqrt{\sum_{i=1}^{t} \|\nabla f(x_i)\|^2}}.
\]

- \( Z_i^2 \sim \|\tilde{F}(z)\|^2 \), since when \( Z = \mathbb{R}^{d+n} \),

\[
\|z_{i+1/2} - z_i\| = \|\eta_i \tilde{F}(z_i)\|.
\]
Adaptivity to smoothness and noise

- $F(z) = (\nabla_x \Phi(x,y), -\nabla_y \Phi(x,y)) = \mathbb{E}[\tilde{F}(z)]$

**Algorithm (Stochastic) Adaptive extragradient**

```plaintext
for $t = 0$ to $T - 1$ do
    $z_{t+1/2} = P_Z(z_t - \eta_t \tilde{F}(z_t))$
    $z_{t+1} = P_Z(z_t - \eta_t \tilde{F}(z_{t+1/2}))$
end for
```

- Run EG w/ adaptive step size
  [Bach and Levy, 2019]
  $(\max_{x,y} \|x - y\| \leq D, G_0 > 0)$,
  
  $$\eta_t = \frac{D}{\sqrt{G_0^2 + \sum_{i=0}^{t-1} Z_i^2}}$$
  \[ (1) \]

  where, $Z_i^2 = \frac{\|z_{i+1} - z_{i+1/2}\|^2 + \|z_{i+1/2} - z_i\|^2}{5\eta_i^2}$.

**Theorem (Deterministic).** [Bach and Levy, 2019]

EG w/ perfect oracle $\tilde{F} = F$ & $\eta_t$ in (1):
- Rate: $\text{err} \left( \frac{1}{T} \sum_{t=1}^T z_{t+1/2} \right) \leq O \left( \frac{1}{T} \right)$

**Theorem (Stochastic).** [Bach and Levy, 2019]

EG w/ noisy oracles $\mathbb{E}[\tilde{F}(z)] = F(z)$ & $\eta_t$ in (1):
- Rate: $\mathbb{E} \text{err} \left( \frac{1}{T} \sum_{t=1}^T z_{t+1/2} \right) \leq O \left( \frac{1}{\sqrt{T}} \right)$
Takeaway

Algorithm (Stochastic) Adaptive extragradient
[Bach and Levy, 2019]

for $t = 0$ to $T - 1$ do

$z_{t+1/2} = P_Z(z_t - \eta_t \tilde{F}(z_t))$

$z_{t+1} = P_Z(z_t - \eta_t \tilde{F}(z_{t+1/2}))$

end for

→ Same step size for deterministic & stochastic
→ No need to know $L$
→ Optimal rate interpolation between for deterministic & stochastic

EG+AdaGrad step size

$$\eta_t = \frac{D}{\sqrt{G_0^2 + \sum_{i=0}^{t-1} Z_i^2}},$$

where,

$$Z_i^2 = \frac{\|z_{i+1} - z_{i+1/2}\|^2 + \|z_{i+1/2} - z_i\|^2}{5\eta_i^2}.$$
ExtraAdam

Algorithm ExtraAdam [Gidel et al., 2018]

\[
\begin{align*}
\text{for } t = 0 \text{ to } T - 1 \text{ do} \\
g_t &= \tilde{F}(z_t) \\
m_{t-1/2} &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\
v_{t-1/2} &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \\
z_{t+1/2} &= z_t - \frac{\eta}{\sqrt{v_{t-1/2}}} m_{t-1} \\
g_{t+1/2} &= \tilde{F}(z_{t+1/2}) \\
m_t &= \beta_1 m_{t-1/2} + (1 - \beta_1) g_{t+1/2} \\
v_t &= \beta_2 v_{t-1/2} + (1 - \beta_2) g_{t+1/2}^2 \\
z_{t+1} &= z_t - \frac{\eta}{\sqrt{v_t}} m_t 
\end{align*}
\]

Extragradient step

Main update step

\[ \rightarrow \text{Extragradient + Adam} \]
\[ \rightarrow \text{Limited theoretical understanding} \]
\[ \rightarrow \text{Compelling practical performance for training GANs} \]

\[ ^1 \text{Bias correction steps are omitted from ExtraAdam for simplicity.} \]
Real LSUN Dataset: Extra-Adam, $4 \times 10^4, 8 \times 10^4, 10^5$ iterations [Hsieh et al., 2019]
### Classic methods

<table>
<thead>
<tr>
<th>Adaptivity to</th>
<th>Lipschitz constant</th>
<th>noise</th>
<th>strong convexity</th>
<th>sparsity of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Bach and Levy, 2019]</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>[Chambolle et al., 2018]</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>[Alacaoglu et al., 2020]</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Bilinear setting

\[
\min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^n} \langle Ax, y \rangle + f(x) - h(y)
\]

- \( f, h \) are closed, convex functions
- Solution set is nonempty
- Linearly constrained problems, TV regularization, empirical risk minimization...

**Algorithm** Alternating GDA (Arrow-Hurwicz)

\[
\text{for } t = 0 \text{ to } T - 1 \text{ do} \\
\quad x_{t+1} = \text{prox}_{\tau f}(x_t - \tau A^\top y_t) \\
\quad y_{t+1} = \text{prox}_{\sigma h}(y_t + \sigma A x_{t+1}) \\
\text{end for}
\]

- \( \text{prox}_{\tau g}(u) = \arg \min_x g(x) + \frac{1}{2\tau} \|x - u\|^2 \)
Bilinear setting

\[
\min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^n} \langle Ax, y \rangle + f(x) - h(y)
\]

- \(f, h\) are closed, convex functions
- Solution set is nonempty
- Linearly constrained problems, TV regularization, empirical risk minimization...

**Algorithm** Alternating GDA (Arrow-Hurwicz)

\[
\begin{align*}
\text{for } t = 0 \text{ to } T - 1 \text{ do} \\
\quad x_{t+1} &= \text{prox}_{\tau f}(x_t - \tau A^\top y_t) \\
\quad y_{t+1} &= \text{prox}_{\sigma h}(y_t + \sigma A x_{t+1}) \\
\text{end for}
\end{align*}
\]

- \(\text{prox}_{\tau g}(u) = \arg \min_x g(x) + \frac{1}{2\tau} \|x - u\|^2\)

**Algorithm** Primal-dual hybrid gradient (PDHG) [Chambolle and Pock, 2011]

\[
\begin{align*}
\text{for } t = 0 \text{ to } T - 1 \text{ do} \\
\quad x_{t+1} &= \text{prox}_{\tau f}(x_t - \tau A^\top (y_t + y_t - y_{t-1})) \\
\quad y_{t+1} &= \text{prox}_{\sigma h}(y_t + \sigma A x_{t+1}) \\
\text{end for}
\end{align*}
\]

- PDHG w/ step sizes \(\tau \sigma \|A\|^2 < 1\) converges to \((x^*, y^*)\) [Chambolle and Pock, 2011].
- Rate: \(\text{err} \left( \frac{1}{T} \sum_{t=1}^T z_t \right) \leq O \left( \frac{1}{T} \right)\)
Separable bilinear setting

\[
\min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^n} f(x) + \sum_{i=1}^{n} \langle A_i x, y_i \rangle - h_i(y_i)
\]

- Linearly constrained problems, TV regularization, empirical risk minimization...

**Algorithm**  Stochastic Primal-dual hybrid gradient (SPDHG) [Chambolle et al., 2018]

\[
\begin{align*}
\text{for } t = 0 \text{ to } T - 1 \text{ do} \\
\quad x_{t+1} &= \text{prox}_{\tau f}(x_t - \tau A^\top \bar{y}_t) \\
\quad \text{Pick } i_t &\in [n] \text{ randomly} \\
\quad y_{t+1,i_t} &= \text{prox}_{\sigma_{i_t} h_{i_t}}(y_{t,i_t} + \sigma A_{i_t} x_{t+1}) \\
\quad y_{t+1,i} &= y_{t,i}, \text{ for } i \neq i_t \\
\quad \bar{y}_{t+1} &= y_{t+1} + n(y_{t+1} - y_t)
\end{align*}
\]
Separable bilinear setting

$$\min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^n} f(x) + \sum_{i=1}^{n} \langle A_i x, y_i \rangle - h_i(y_i)$$

○ Linearly constrained problems, TV regularization, empirical risk minimization...

**Algorithm** Stochastic Primal-dual hybrid gradient (SPDHG) [Chambolle et al., 2018]

for $t = 0$ to $T - 1$ do
  $x_{t+1} = \text{prox}_{\tau f}(x_t - \tau A^T \bar{y}_t)$
  Pick $i_t \in [n]$ randomly
  $y_{t+1,i_t} = \text{prox}_{\sigma_i h_{i_t}}(y_{t,i_t} + \sigma A_{i_t} x_{t+1})$
  $y_{t+1,i} = y_t,i$, for $i \neq i_t$
  $\bar{y}_{t+1} = y_{t+1} + n(y_{t+1} - y_t)$
end for

**Theorem.** [Alacaoglu et al., 2019] PDHG w/ step sizes $n \tau \sigma_i \|A_i\|^2 < 1$:

○ Convergence: $(x_t, y_t) \to (x^*, y^*)$

○ Rate:
  $$\mathbb{E} \text{err} \left( \frac{1}{T} \sum_{t=1}^{T} x_t, \frac{1}{T} \sum_{t=1}^{T} y_t \right) \leq \mathcal{O} \left( \frac{1}{T} \right)$$
Separable bilinear setting

\[
\begin{align*}
\min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^n} f(x) + \sum_{i=1}^n \langle A_i x, y_i \rangle - h_i(y_i)
\end{align*}
\]

- Linearly constrained problems, TV regularization, empirical risk minimization...

**Algorithm** Stochastic Primal-dual hybrid gradient (SPDHG) [Chambolle et al., 2018]

```
for t = 0 to T - 1 do
    x_{t+1} = \text{prox}_{\tau f}(x_t - \tau A^\top \bar{y}_t)
    \text{Pick } i_t \in [n] \text{ randomly}
    y_{t+1,i_t} = \text{prox}_{\sigma_{i_t} h_{i_t}}(y_{t,i_t} + \sigma A_{i_t} x_{t+1})
    y_{t+1,i} = y_{t,i}, \text{ for } i \neq i_t
    \bar{y}_{t+1} = y_{t+1} + n(y_{t+1} - y_t)
end for
```

**Theorem.** [Alacaoglu et al., 2019] PDHG w/ step sizes \( n\tau\sigma_i \| A_i \|^2 < 1 \):
- Convergence: \((x_t, y_t) \to (x^*, y^*)\)
- Rate:
  \[
  \mathbb{E} \text{ err} \left( \frac{1}{T} \sum_{t=1}^T x_t, \frac{1}{T} \sum_{t=1}^T y_t \right) \leq \mathcal{O} \left( \frac{1}{T} \right)
  \]

**Theorem.** [Chambolle et al., 2018] If \( f \) and \( h \) are \((\mu_f, \mu_i)\) strongly convex and \( \tau, \sigma_i \) are chosen depending on \((\mu_f, \mu_i)\):

\[
(x_t, y_t) \text{ converge linearly to } (x^*, y^*).
\]
Takeaway

\[
\min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^n} f(x) + \sum_{i=1}^{n} \langle A_i x, y_i \rangle - h_i(y_i)
\]

Algorithm  Stochastic Primal-dual hybrid gradient (SPDHG)

\[
\text{for } t = 0 \text{ to } T - 1 \text{ do}
\]
\[
x_{t+1} = \text{prox}_{\tau f}(x_t - \tau A^\top \bar{y}_t)
\]
\[
\text{Pick } i_t \in [n] \text{ randomly}
\]
\[
y_{t+1,i_t} = \text{prox}_{\sigma_{i_t} h_{i_t}}(y_{t,i_t} + \sigma A_{i_t} x_{t+1})
\]
\[
y_{t+1,i} = y_{t,i}, \text{ for } i \neq i_t
\]
\[
\bar{y}_{t+1} = y_{t+1} + n(y_{t+1} - y_t)
\]
\[
\text{end for}
\]

→ Different step sizes for sublinear & linear rate
→ Need knowledge of \(\mu_f, \mu_i\) to set step sizes for linear rate
Convergence of SPDHG under metric subregularity

\[
\min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^n} f(x) + \sum_{i=1}^{n} \langle A_i x, y_i \rangle - h_i(y_i)
\]

**Algorithm** Stochastic Primal-dual hybrid gradient (SPDHG)

```
for t = 0 to T - 1 do
    x_{t+1} = \text{prox}_{\tau f}(x_t - \tau A^T \bar{y}_t)
    \text{Pick } i_t \in [n] \text{ randomly}
    y_{t+1,i_t} = \text{prox}_{\sigma_{i_t} h_{i_t}}(y_{t,i_t} + \sigma A_{i_t} x_{t+1})
    y_{t+1,i} = y_{t,i}, \text{ for } i \neq i_t
    \bar{y}_{t+1} = y_{t+1} + n(y_{t+1} - y_t)
end for
```

- **Metric subregularity:** Generalization of strong convexity.
- Satisfied when \( f, h \) are strongly convex, or when \( f, h \) are piecewise linear-quadratic (PLQ): indicator of polyhedral sets, polyhedral norms, hinge loss, Huber loss etc.

**Theorem.** [Alacaoglu et al., 2019] Assume metric subregularity and pick \( \tau, \sigma \) as \( n \tau \sigma_i \| A_i \|^2 < 1 \), then

\((x_t, y_t)\) converge linearly to \((x^*, y^*)\).
Performance

\[
\min_{x \in \mathbb{R}^d} \|x\|_1 : \quad Ax = b
\]

- Synthetic setup: \(A\) has normal distribution with \(\Sigma_{i,j} = 0.5|i-j|\)
- \(n, d = 500, 1000\)
Takeaway

Algorithm  Stochastic Primal-dual hybrid gradient (SPDHG) [Chambolle et al., 2018]

\[
\text{for } t = 0 \text{ to } T - 1 \text{ do}
\]
\[
x_{t+1} = \text{prox}_{\tau f}(x_t - \tau A^\top \bar{y}_t)
\]
Pick \(i_t \in [n]\) randomly
\[
y_{t+1,i_t} = \text{prox}_{\sigma_{i_t} h_{i_t}}(y_{t,i_t} + \sigma A_{i_t} x_{t+1})
\]
\[
y_{t+1,i} = y_{t,i}, \text{ for } i \neq i_t
\]
\[
\bar{y}_{t+1} = y_{t+1} + n(y_{t+1} - y_t)
\]
\text{end for}

\rightarrow \text{Randomization speeds up PDHG} \\
\rightarrow \text{Same step sizes with and without strong convexity} \\
\rightarrow \text{No need to know } \mu_f, \mu_i \text{ to obtain linear convergence}

\circ \text{ Our main reference for theoretical results: [Alacaoglu et al., 2019]}
### Classic methods

<table>
<thead>
<tr>
<th>Adaptivity to</th>
<th>Lipschitz constant</th>
<th>noise</th>
<th>strong convexity</th>
<th>sparsity of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Bach and Levy, 2019]</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>[Chambolle et al., 2018]</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>[Alacaoglu et al., 2020]</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Per iteration cost of SPDHG

\[
\min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^n} f(x) + \sum_{i=1}^{n} \langle A_i x, y_i \rangle - h_i(y_i)
\]

Algorithm Stochastic Primal-dual hybrid gradient (SPDHG)

\begin{verbatim}
for t = 0 to T - 1 do
    x_{t+1} = \text{prox}_{\tau f}(x_t - \tau A^\top \bar{y}_t)
    Pick \ i_t \in [n] \ randomly
    y_{t+1,i_t} = \text{prox}_{\sigma_{i_t} h_{i_t}}(y_{t,i_t} + \sigma A_{i_t} x_{t+1})
    y_{t+1,i} = y_{t,i}, \ for \ i \neq i_t
    \bar{y}_{t+1} = y_{t+1} + n(y_{t+1} - y_t)
end for
\end{verbatim}
Per iteration cost of SPDHG

\[
\min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^n} f(x) + \sum_{i=1}^{n} \langle A_i x, y_i \rangle - h_i(y_i)
\]

<table>
<thead>
<tr>
<th>Algorithm (SPDHG)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stochastic Primal-dual hybrid gradient</strong></td>
</tr>
</tbody>
</table>

\[\text{for } t = 0 \text{ to } T - 1 \text{ do}\]
- \[x_{t+1} = \text{prox}_{\tau f}(x_t - \tau A^\top \bar{y}_t)\]
- Pick \(i_t \in [n]\) randomly
- \[y_{t+1,i_t} = \text{prox}_{\sigma_i h_{i_t}}(y_{t,i_t} + \sigma A_{i_t} x_{t+1})\]
- \[y_{t+1,i} = y_{t,i}, \text{ for } i \neq i_t\]
- \[\bar{y}_{t+1} = y_{t+1} + n(y_{t+1} - y_t)\]
\[\text{end for}\]

Analysis of per iteration cost:

- Recall \(A = \begin{bmatrix} A_1 \\
                         \vdots \\
                         A_n \end{bmatrix}, \text{ with } A_i^\top \in \mathbb{R}^d\).
- Compute \(x_{t+1} \rightarrow \text{cost } d\).
- \([Ax_{k+1}]_{i_t} = \langle A_{i_t}^\top, x_{k+1} \rangle \rightarrow \text{cost } \text{nnz}(A_{i_t})\).
- We maintain \(A^\top y_t\) and compute:
  \[A^\top \bar{y}_{t+1} = A^\top y_t + (n + 1)A_{i_t}^\top (y_{t+1} - y_{t,i_t})\]
  \[\rightarrow \text{cost } \text{nnz}(A_{i_t}).\]
Algorithm PURE-CD [Alacaoglu et al., 2020]

for $t = 0$ to $T - 1$ do

$\bar{x}_{t+1} = \text{prox}_{\tau f}(x_t - \tau A^\top y_t)$

$\bar{y}_{t+1} = \text{prox}_{\sigma h}(y_t + \sigma A \bar{x}_{t+1})$

Pick $i_t \in [n]$ randomly

$y_{t+1, i_t} = \bar{y}_{t+1, i_t}$

$y_{t+1, j} = y_{t, j}$, $\forall j \neq i_t$

$x_{t+1, j} = \bar{x}_{t+1, j} - \tau_j \theta_j [A^\top(y_{t+1} - y_t)]_j$, $\forall j \in J(i_t)$

$x_{t+1, j} = x_{t, j}$, $\forall j \notin J(i_t)$

end for

Parameters:

- $I(j) = \{i \in [n] : A_{i, j} \neq 0\}$
- $J(i) = \{j \in [d] : A_{i, j} \neq 0\}$
- $(\tau_i)_{i=1}^d$ and $(\sigma_i)_{i=1}^n$ are chosen using nonzero entries of $A$ due to $I(j)$ and $J(i)$.

\[
\text{step size w. dense } A \quad \text{iter. cost} \quad \frac{n\tau \sigma_i \|A_i\|^2}{\|A_i\|_2^2} < 1 \quad \text{nnz}(A_i)
\]

... compared to SPDHG that had

\[
\text{step size w. dense } A \quad \text{iter. cost} \quad \frac{n\tau \sigma_i \|A_i\|^2}{\|A_i\|_2^2} < 1 \quad d
\]

$\rightarrow$ Significant improvement when $d$ is big & $A$ is sparse

$\rightarrow$ Similar theoretical rates as SPDHG
PURE-CD

- Lasso with different levels of sparsity of data.

Figure: Lasso: Left: rcv1, \( n = 20,242, m = 47,236 \), density = 0.16\%, \( \lambda = 10 \); Middle: w8a, \( n = 49,749, m = 300 \), density = 3.9\%, \( \lambda = 10^{-1} \); Right: covtype, \( n = 581,012, m = 54 \), density = 22.1\%, \( \lambda = 10 \).
Takeaway

**Algorithm**  PURE-CD [Alacaoglu et al., 2020]

\[
\begin{align*}
&\text{for } t = 0 \text{ to } T - 1 \text{ do} \\
&\quad \bar{x}_{t+1} = \text{prox}_{\tau f}(x_t - \tau A^T y_t) \\
&\quad \bar{y}_{t+1} = \text{prox}_{\sigma h}(y_t + \sigma A \bar{x}_{t+1}) \\
&\quad \text{Pick } i_t \in [n] \text{ randomly} \\
&\quad y_{t+1,i_t} = \bar{y}_{t+1,i_t} \\
&\quad y_{t+1,j} = y_{t,j}, \forall j \neq i_t \\
&\quad x_{t+1,j} = \bar{x}_{t+1,j} - \tau_j \theta_j [A^T (y_{t+1} - y_t)]_j, \forall j \in J(i_t) \\
&\quad x_{t+1,j} = x_{t,j}, \forall j \notin J(i_t) \\
&\end{align*}
\]

→ Randomization speeds up PDHG
→ No need to know $\mu_f, \mu_i$ to obtain linear convergence
→ Per-iteration cost and step sizes adapt to sparsity.
## Summary

\[
\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \Phi(x, y)
\]

<table>
<thead>
<tr>
<th>Adaptivity to</th>
<th>Lipschitz constant</th>
<th>noise</th>
<th>strong convexity</th>
<th>sparsity of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Bach and Levy, 2019]</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>[Chambolle et al., 2018]</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>[Alacaoglu et al., 2020]</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
References I

On the convergence of stochastic primal-dual hybrid gradient.  

Random extrapolation for primal-dual coordinate descent.  
In International conference on machine learning, pages 191–201. PMLR.

Studies in linear and non-linear programming, volume 2.  
Stanford University Press.

A universal algorithm for variational inequalities adaptive to smoothness and noise.  

Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications.  


