dissolve$^{\text{struct}}$ ALGORITHM HANDBOOK

TRIBHUVANESH OREKONDY

Contents

1. Introduction

2. Stochastic Gradient Descent (SGD)

Algorithm 1: SGD: Stochastic Gradient Descent

Input: Data $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$
Initialize: $\boldsymbol{w}^{(0)} \leftarrow \mathbf{0}$
for $t = 1 \dots T$
  Choose $i \in \{1, 2, \dots, n\}$ uniformly at random
  Solve $\hat{\boldsymbol{y}}_i \leftarrow \text{argmax}_{\boldsymbol{y} \in \mathcal{Y}_i} H_i(\boldsymbol{y};\ \boldsymbol{w}^{(t-1)})$ `// Max Oracle`
  Let $p \leftarrow \lambda \boldsymbol{w}^{(t-1)} - \psi_i(\hat{\boldsymbol{y}}_i)$ `// Compute gradient`
  Update $\boldsymbol{w}^{(t)} \leftarrow \boldsymbol{w}^{(t-1)} - \gamma_t p$
end
Output: $\boldsymbol{w}^{(T)}$

3. Mini-Batch Stochastic Gradient Descent (MB-SGD)

Algorithm 2: MB-SGD: Mini-batch Stochastic Gradient Descent

Input: Data $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$
Initialize: $\boldsymbol{w}^{(0)} \leftarrow \boldsymbol{0}$
for $t = 1 \dots T$
  for $k = 1 \dots K$, *in parallel*
    for $i \in [k]$
      Solve $\hat{\boldsymbol{y}}_i \leftarrow \text{argmax}_{\boldsymbol{y} \in \mathcal{Y}_i} H_i(\boldsymbol{y};\ \boldsymbol{w}^{(t-1)})$ // Max Oracle
    end
    Let $p \leftarrow \lambda \boldsymbol{w}^{(t-1)} - \sum_{i \in [k]} \psi_i(\hat{\boldsymbol{y}}_i)$
    Communicate $\Delta \boldsymbol{w}_k \leftarrow \gamma_t p$
  end
  Update $\boldsymbol{w}^{(t)} \leftarrow \boldsymbol{w}^{(t-1)} - \frac{\beta}{K} \sum_{k=1}^K \Delta \boldsymbol{w}_k$ // Merge updates
end
Output: $\boldsymbol{w}^{(T)}$

4. Block-Coordinate Frank-Wolfe (BCFW)

Algorithm 3: Bcfw: Block-Coordinate Frank-Wolfe algorithm for Structured SVM

Input: Data $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$
Initialize: $\boldsymbol{w}^{(0)} \leftarrow \boldsymbol{0},\ \boldsymbol{w}_i^{(0)} \leftarrow \boldsymbol{0},\ \ell^{(0)} \leftarrow 0,\ \ell_i^{(0)} \leftarrow 0$
for $t = 1 \dots T$
  Choose $i \in \{1, 2, \dots, n\}$ uniformly at random
  Solve $\hat{\boldsymbol{y}}_i \leftarrow \text{argmax}_{\boldsymbol{y} \in \mathcal{Y}_i} H_i(\boldsymbol{y};\ \boldsymbol{w}^{(t)})$
  Let $\boldsymbol{w}_{\boldsymbol{s}} \leftarrow \frac{1}{\lambda n} \psi_i(\hat{\boldsymbol{y}}_i)$ and $\ell_{\boldsymbol{s}} \leftarrow \frac{1}{M} \Delta^m(\hat{\boldsymbol{y}}_i)$
  Let $\gamma \leftarrow \frac{\lambda \left(\boldsymbol{w}_i^{(t-1)} - \boldsymbol{w}_{\boldsymbol{s}}\right)^T \boldsymbol{w}^{(t-1)} - \ell_i^{(t-1)} + \ell_{\boldsymbol{s}}}{\lambda \|\boldsymbol{w}_i^{(t-1)} - \boldsymbol{w}_{\boldsymbol{s}}\|^2}$ and clip to $[0, 1]$
  Update $\boldsymbol{w}_i^{(t)} \leftarrow (1 - \gamma) \boldsymbol{w}_i^{(t-1)} + \gamma \boldsymbol{w}_{\boldsymbol{s}}$
    and $\ell_i^{(t)} \leftarrow (1 - \gamma) \ell_i^{(t-1)} + \gamma \ell_{\boldsymbol{s}}$
  Update $\boldsymbol{w}^{(t)} \leftarrow \boldsymbol{w}^{(t-1)} + \boldsymbol{w}_i^{(t)} - \boldsymbol{w}_i^{(t-1)}$
    and $\ell^{(t)} \leftarrow \ell^{(t-1)} + \ell_i^{(t)} - \ell_i^{(t-1)}$
end
Output: $\boldsymbol{w}^{(T)}$ and $\ell^{(T)}$

5. CoCoA Block-Coordinate Frank-Wolfe (CoCoA-BCFW)

Procedure A: LocalBcfw: BCFW iterations on machine k

Input: $f \in (0,1]$, $\boldsymbol{w}_{[k]} \in \mathbb{R}^{n_k \times d}$ and $\boldsymbol{w} \in \mathbb{R}^d$ consistent with other coordinate blocks of $\boldsymbol{\alpha}$ s.t. $\boldsymbol{w} = A\boldsymbol{\alpha}$

Data: Local $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i \in n_k}$

Initialize: $\boldsymbol{w}^{(0)} \leftarrow \boldsymbol{w},\ \boldsymbol{w}_{[k]}^{(0)} \leftarrow \boldsymbol{w}_{[k]} \in \mathbb{R}^{n_k \times d}$

for $r = 1, 2, \ldots, R$
- choose $i \in [k]$ uniformly at random
- Solve $\hat{\boldsymbol{y}} \leftarrow \operatorname{argmax}_{\boldsymbol{y} \in \mathcal{Y}_i} H_i(\boldsymbol{y};\ \boldsymbol{w}^{(r-1)})$
- $\boldsymbol{w}_{\boldsymbol{s}} \leftarrow \frac{1}{\lambda n}\psi_i(\hat{\boldsymbol{y}}_i)$ and $\ell_{\boldsymbol{s}} \leftarrow \frac{1}{n} L_i(\hat{\boldsymbol{y}}_i)$
- $\gamma \leftarrow \frac{\lambda\left(\boldsymbol{w}_i^{(r-1)} - \boldsymbol{w}_{\boldsymbol{s}}\right)^T \boldsymbol{w}_i^{(r-1)} \ - \ \ell_i^{(r-1)} \ + \ \ell_{\boldsymbol{s}}}{\lambda \|\boldsymbol{w}_i^{(r-1)} - \boldsymbol{w}_{\boldsymbol{s}}\|^2}$ and clip to $[0,1]$
- Update $\boldsymbol{w}_i^{(r)} \leftarrow (1-\gamma)\boldsymbol{w}_i^{(r-1)} \ + \ \gamma \boldsymbol{w}_{\boldsymbol{s}}$
  and $\ell_i^{(r)} \leftarrow (1-\gamma)\ell_i^{(r-1)} \ + \ \gamma \ell_{\boldsymbol{s}}$
- Update $\boldsymbol{w}^{(r)} \leftarrow \boldsymbol{w}^{(r-1)} \ + \ \boldsymbol{w}_i^{(r)} \ - \ \boldsymbol{w}_i^{(r-1)}$
  and $\ell^{(r)} \leftarrow \ell^{(r-1)} \ + \ \ell_i^{(r)} \ - \ \ell_i^{(r-1)}$

end

$\Delta \boldsymbol{w}_{[k]} \leftarrow \boldsymbol{w}_{[k]}^{(R)} - \boldsymbol{w}_{[k]}$ and $\Delta \ell_{[k]} \leftarrow \ell_{[k]}^{(R)} - \ell_{[k]}$

$\Delta \boldsymbol{w}_k \leftarrow \boldsymbol{w}^{(R)} - \boldsymbol{w}$ and $\Delta \ell_k \leftarrow \ell^{(R)} - \ell$

Output: $\Delta \boldsymbol{w}_{[k]},\ \Delta \boldsymbol{w},\ \Delta \ell_{[k]},\ \Delta \ell$

Algorithm 4: Cocoa-BCFW: Communication-Efficient Distributed BCFW

Input: $T \geq 1$, scaling parameter $1 \leq \beta_K \leq K$ (default: $\beta_K := 1$).

Data: $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ distributed over K machines

Initialize: $\boldsymbol{w}_{(0)}^{[k]} \leftarrow \mathbf{0},\ \ell_{(0)}^{[k]} \leftarrow 0$ for all machines k and $\boldsymbol{w}_{(0)} \leftarrow \mathbf{0},\ \ell_{(0)} \leftarrow 0$

for $t = 1, 2, \ldots, T$
- **for** ***all machines*** $k = 1, 2, \ldots, K$ ***in parallel***
  - $(\Delta \boldsymbol{w}_{[k]},\ \Delta \boldsymbol{w}_k) \leftarrow$ LocalBcfw$(\boldsymbol{w}_{[k]}^{(t-1)},\ \boldsymbol{w}^{(t-1)})$
  - $\boldsymbol{w}_{[k]}^{(t)} \leftarrow \boldsymbol{w}_{[k]}^{(t-1)} + \frac{\beta_K}{K} \Delta \boldsymbol{w}_{[k]}$
  - $\ell_{[k]}^{(t)} \leftarrow \ell_{[k]}^{(t-1)} + \frac{\beta_K}{K} \Delta \ell_{[k]}$
- **end**
- reduce $\boldsymbol{w}^{(t)} \leftarrow \boldsymbol{w}^{(t-1)} + \frac{\beta_K}{K} \sum_{k=1}^K \Delta \boldsymbol{w}^k$
  and $\ell^{(t)} \leftarrow \ell^{(t-1)} + \frac{\beta_K}{K} \sum_{k=1}^K \Delta \ell^k$

end

6. Mini-Batch Block-Coordinate Frank-Wolfe (MB-BCFW)

Procedure B: MB-LocalBcfw: Mini-Batch Bcfw iterations on machine k

Input: $f \in (0,1]$, $\boldsymbol{w}_{[k]} \in \mathbb{R}^{n_k \times d}$ and $\boldsymbol{w} \in \mathbb{R}^d$ consistent with other coordinate blocks of $\boldsymbol{\alpha}$ s.t. $\boldsymbol{w} = A\boldsymbol{\alpha}$

Data: Local $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i \in n_k}$

Initialize: $\boldsymbol{w}^{(0)} \leftarrow \boldsymbol{w},\ \boldsymbol{w}_{[k]}^{(0)} \leftarrow \boldsymbol{w}_{[k]} \in \mathbb{R}^{n_k \times d}, \Delta\boldsymbol{w} \leftarrow \mathbf{0} \in \mathbb{R}^d, \ell \leftarrow 0$

for $r = 1, 2, \ldots, R$
- choose $i \in [k]$ uniformly at random
- Solve $\hat{\boldsymbol{y}} \leftarrow \operatorname{argmax}_{\boldsymbol{y} \in \mathcal{Y}_i} H_i(\boldsymbol{y};\ \boldsymbol{w})$
- $\boldsymbol{w_s} \leftarrow \frac{1}{\lambda n}\psi_i(\hat{\boldsymbol{y}}_i)$ and $\ell_{\boldsymbol{s}} \leftarrow \frac{1}{n}L_i(\hat{\boldsymbol{y}}_i)$
- $\gamma \leftarrow \frac{\lambda\left(\boldsymbol{w}_i^{(r-1)} - \boldsymbol{w_s}\right)^T \boldsymbol{w}\ -\ \ell_i^{(r-1)}\ +\ \ell_{\boldsymbol{s}}}{\lambda\|\boldsymbol{w}_i^{(r-1)} - \boldsymbol{w_s}\|^2}$ and clip to $[0,1]$
- Update $\boldsymbol{w}_i^{(r)} \leftarrow (1-\gamma)\boldsymbol{w}_i^{(r-1)}\ +\ \gamma\boldsymbol{w_s}$
  and $\ell_i^{(r)} \leftarrow (1-\gamma)\ell_i^{(r-1)}\ +\ \gamma\ell_{\boldsymbol{s}}$
- Update $\Delta\boldsymbol{w} \leftarrow \Delta\boldsymbol{w}\ +\ \boldsymbol{w}_i^{(r)}\ -\ \boldsymbol{w}_i^{(r-1)}$
  and $\Delta\ell \leftarrow \Delta\ell\ +\ \ell_i^{(r)}\ -\ \ell_i^{(r-1)}$

end

$\Delta\boldsymbol{w}_{[k]} \leftarrow \boldsymbol{w}_{[k]}^{(R)} - \boldsymbol{w}_{[k]}$ and $\Delta\ell_{[k]} \leftarrow \ell_{[k]}^{(R)} - \ell_{[k]}$

$\Delta\boldsymbol{w}_k \leftarrow \Delta\boldsymbol{w}$ and $\Delta\ell_k \leftarrow \Delta\ell$

Output: $\Delta\boldsymbol{w}_{[k]},\ \Delta\boldsymbol{w},\ \Delta\ell_{[k]},\ \Delta\ell$

Algorithm 5: MB-Bcfw: Mini-Batch Distributed Bcfw on Master

Input: $T \geq 1$, scaling parameter $1 \leq \beta_K \leq K$ (default: $\beta_K := 1$).

Data: $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ distributed over K machines

Initialize: $\boldsymbol{w}_{(0)}^{[k]} \leftarrow \mathbf{0},\ \ell_{(0)}^{[k]} \leftarrow 0$ for all machines k and $\boldsymbol{w}_{(0)} \leftarrow \mathbf{0},\ \ell_{(0)} \leftarrow 0$

for $t = 1, 2, \ldots, T$
- **for** ***all machines*** $k = 1, 2, \ldots, K$ ***in parallel***
  - $(\Delta\boldsymbol{w}_{[k]},\ \Delta\boldsymbol{w}_k) \leftarrow$ MB-LocalBcfw$(\boldsymbol{w}_{[k]}^{(t-1)},\ \boldsymbol{w}^{(t-1)})$
  - $\boldsymbol{w}_{[k]}^{(t)} \leftarrow \boldsymbol{w}_{[k]}^{(t-1)} + \frac{\beta_K}{K}\Delta\boldsymbol{w}_{[k]}$
  - $\ell_{[k]}^{(t)} \leftarrow \ell_{[k]}^{(t-1)} + \frac{\beta_K}{K}\Delta\ell_{[k]}$
- **end**
- reduce $\boldsymbol{w}^{(t)} \leftarrow \boldsymbol{w}^{(t-1)} + \frac{\beta_K}{K}\sum_{k=1}^K \Delta\boldsymbol{w}^k$
  and $\ell^{(t)} \leftarrow \ell^{(t-1)} + \frac{\beta_K}{K}\sum_{k=1}^K \Delta\ell^k$

end

7. CoCoA$^+$ Bcfw

Algorithm 6: CoCoA$^+$-Bcfw: Communication-Efficient Distributed BCFW

Input: $T \geq 1$, aggregation parameter $0 \leq \beta_K \leq 1$ (default: $\beta_K := 1$).
Data: $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ distributed over K machines
Initialize: $\boldsymbol{w}_{(0)}^{[k]} \leftarrow \boldsymbol{0},\ \ell_{(0)}^{[k]} \leftarrow 0$ for all machines k and $\boldsymbol{w}_{(0)} \leftarrow \boldsymbol{0},\ \ell_{(0)} \leftarrow 0$

for $t = 1, 2, \ldots, T$
  for *all machines* $k = 1, 2, \ldots, K$ ***in parallel***
    $(\Delta \boldsymbol{w}_{[k]},\ \Delta \boldsymbol{w}_k) \leftarrow \text{LocalBcfw}(\boldsymbol{w}_{[k]}^{(t-1)},\ \boldsymbol{w}^{(t-1)})$
    $\boldsymbol{w}_{[k]}^{(t)} \leftarrow \boldsymbol{w}_{[k]}^{(t-1)} + \beta_K \Delta \boldsymbol{w}_{[k]}$
    $\ell_{[k]}^{(t)} \leftarrow \ell_{[k]}^{(t-1)} + \beta_K \Delta \ell_{[k]}$
  end
  reduce $\boldsymbol{w}^{(t)} \leftarrow \boldsymbol{w}^{(t-1)} + \beta_K \sum_{k=1}^K \Delta \boldsymbol{w}^k$
    and $\ell^{(t)} \leftarrow \ell^{(t-1)} + \beta_K \sum_{k=1}^K \Delta \ell^k$
end