

dissolve^{struct} ALGORITHM HANDBOOK

TRIBHUVANESH OREKONDY

CONTENTS

1. Introduction	1
2. Stochastic Gradient Descent (SGD)	1
3. Mini-Batch Stochastic Gradient Descent (MB-SGD)	2
4. Block-Coordinate Frank-Wolfe (BCFW)	2
5. CoCoA Block-Coordinate Frank-Wolfe (CoCoA-BCFW)	3
6. Mini-Batch Block-Coordinate Frank-Wolfe (MB-BCFW)	4
7. CoCoA ⁺ BCFW	5

1. INTRODUCTION

2. STOCHASTIC GRADIENT DESCENT (SGD)

Algorithm 1: SGD: Stochastic Gradient Descent

Input: Data $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$

Initialize: $\mathbf{w}^{(0)} \leftarrow \mathbf{0}$

1 **for** $t = 1 \dots T$

2 Choose $i \in \{1, 2, \dots, n\}$ uniformly at random

3 Solve $\hat{\mathbf{y}}_i \leftarrow \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}_i} H_i(\mathbf{y}; \mathbf{w}^{(t-1)})$ // Max Oracle

4 Let $p \leftarrow \lambda \mathbf{w}^{(t-1)} - \psi_i(\hat{\mathbf{y}}_i)$ // Compute gradient

5 Update $\mathbf{w}^{(t)} \leftarrow \mathbf{w}^{(t-1)} - \gamma_t p$

6 **end**

Output: $\mathbf{w}^{(T)}$

3. MINI-BATCH STOCHASTIC GRADIENT DESCENT (MB-SGD)

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

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

4. BLOCK-COORDINATE FRANK-WOLFE (BCFW)

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

Input: Data $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$
Initialize: $\mathbf{w}^{(0)} \leftarrow \mathbf{0}$, $\mathbf{w}_i^{(0)} \leftarrow \mathbf{0}$, $\ell^{(0)} \leftarrow 0$, $\ell_i^{(0)} \leftarrow 0$
1 **for** $t = 1 \dots T$
2 Choose $i \in \{1, 2, \dots, n\}$ uniformly at random
3 Solve $\hat{\mathbf{y}}_i \leftarrow \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}_i} H_i(\mathbf{y}; \mathbf{w}^{(t)})$
4 Let $\mathbf{w}_s \leftarrow \frac{1}{\lambda n} \psi_i(\hat{\mathbf{y}}_i)$ and $\ell_s \leftarrow \frac{1}{M} \Delta^m(\hat{\mathbf{y}}_i)$
5 Let $\gamma \leftarrow \frac{\lambda (\mathbf{w}_i^{(t-1)} - \mathbf{w}_s)^T \mathbf{w}^{(t-1)} - \ell_i^{(t-1)} + \ell_s}{\lambda \|\mathbf{w}_i^{(t-1)} - \mathbf{w}_s\|^2}$ and clip to $[0, 1]$
6 Update $\mathbf{w}_i^{(t)} \leftarrow (1 - \gamma) \mathbf{w}_i^{(t-1)} + \gamma \mathbf{w}_s$
7 and $\ell_i^{(t)} \leftarrow (1 - \gamma) \ell_i^{(t-1)} + \gamma \ell_s$
8 Update $\mathbf{w}^{(t)} \leftarrow \mathbf{w}^{(t-1)} + \mathbf{w}_i^{(t)} - \mathbf{w}_i^{(t-1)}$
9 and $\ell^{(t)} \leftarrow \ell^{(t-1)} + \ell_i^{(t)} - \ell_i^{(t-1)}$
10 **end**
Output: $\mathbf{w}^{(T)}$ and $\ell^{(T)}$

5. CoCoA BLOCK-COORDINATE FRANK-WOLFE (CoCoA-BCFW)

Procedure A: LOCALBCFW: BCFW iterations on machine k

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

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

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

```

1 for  $r = 1, 2, \dots, R$ 
2   choose  $i \in [k]$  uniformly at random
3   Solve  $\hat{\mathbf{y}} \leftarrow \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}_i} H_i(\mathbf{y}; \mathbf{w}^{(r-1)})$ 
4    $\mathbf{w}_s \leftarrow \frac{1}{\lambda n} \psi_i(\hat{\mathbf{y}}_i)$  and  $\ell_s \leftarrow \frac{1}{n} L_i(\hat{\mathbf{y}}_i)$ 
5    $\gamma \leftarrow \frac{\lambda (\mathbf{w}_i^{(r-1)} - \mathbf{w}_s)^T \mathbf{w}_i^{(r-1)} - \ell_i^{(r-1)} + \ell_s}{\lambda \|\mathbf{w}_i^{(r-1)} - \mathbf{w}_s\|^2}$  and clip to  $[0, 1]$ 
6   Update  $\mathbf{w}_i^{(r)} \leftarrow (1 - \gamma) \mathbf{w}_i^{(r-1)} + \gamma \mathbf{w}_s$ 
7   and  $\ell_i^{(r)} \leftarrow (1 - \gamma) \ell_i^{(r-1)} + \gamma \ell_s$ 
8   Update  $\mathbf{w}^{(r)} \leftarrow \mathbf{w}^{(r-1)} + \mathbf{w}_i^{(r)} - \mathbf{w}_i^{(r-1)}$ 
9   and  $\ell^{(r)} \leftarrow \ell^{(r-1)} + \ell_i^{(r)} - \ell_i^{(r-1)}$ 
10 end
11  $\Delta \mathbf{w}_{[k]} \leftarrow \mathbf{w}_{[k]}^{(R)} - \mathbf{w}_{[k]}$  and  $\Delta \ell_{[k]} \leftarrow \ell_{[k]}^{(R)} - \ell_{[k]}$ 
12  $\Delta \mathbf{w}_k \leftarrow \mathbf{w}^{(R)} - \mathbf{w}$  and  $\Delta \ell_k \leftarrow \ell^{(R)} - \ell$ 
   Output:  $\Delta \mathbf{w}_{[k]}, \Delta \mathbf{w}, \Delta \ell_{[k]}, \Delta \ell$ 

```

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

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

```

1 for  $t = 1, 2, \dots, T$ 
2   for all machines  $k = 1, 2, \dots, K$  in parallel
3      $(\Delta \mathbf{w}_{[k]}, \Delta \mathbf{w}_k) \leftarrow \text{LOCALBCFW}(\mathbf{w}_{[k]}^{(t-1)}, \mathbf{w}^{(t-1)})$ 
4      $\mathbf{w}_{[k]}^{(t)} \leftarrow \mathbf{w}_{[k]}^{(t-1)} + \frac{\beta_K}{K} \Delta \mathbf{w}_{[k]}$ 
5      $\ell_{[k]}^{(t)} \leftarrow \ell_{[k]}^{(t-1)} + \frac{\beta_K}{K} \Delta \ell_{[k]}$ 
6   end
7   reduce  $\mathbf{w}^{(t)} \leftarrow \mathbf{w}^{(t-1)} + \frac{\beta_K}{K} \sum_{k=1}^K \Delta \mathbf{w}^k$ 
8   and  $\ell^{(t)} \leftarrow \ell^{(t-1)} + \frac{\beta_K}{K} \sum_{k=1}^K \Delta \ell^k$ 
9 end

```

6. MINI-BATCH BLOCK-COORDINATE FRANK-WOLFE (MB-BCFW)

Procedure B: MB-LOCALBCFW: Mini-Batch BCFW iterations on machine k

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

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

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

```

1 for  $r = 1, 2, \dots, R$ 
2   choose  $i \in [k]$  uniformly at random
3   Solve  $\hat{\mathbf{y}} \leftarrow \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}_i} H_i(\mathbf{y}; \mathbf{w})$ 
4    $\mathbf{w}_s \leftarrow \frac{1}{\lambda n} \psi_i(\hat{\mathbf{y}}_i)$  and  $\ell_s \leftarrow \frac{1}{n} L_i(\hat{\mathbf{y}}_i)$ 
5    $\gamma \leftarrow \frac{\lambda (\mathbf{w}_i^{(r-1)} - \mathbf{w}_s)^T \mathbf{w} - \ell_i^{(r-1)} + \ell_s}{\lambda \|\mathbf{w}_i^{(r-1)} - \mathbf{w}_s\|^2}$  and clip to  $[0, 1]$ 
6   Update  $\mathbf{w}_i^{(r)} \leftarrow (1 - \gamma) \mathbf{w}_i^{(r-1)} + \gamma \mathbf{w}_s$ 
7   and  $\ell_i^{(r)} \leftarrow (1 - \gamma) \ell_i^{(r-1)} + \gamma \ell_s$ 
8   Update  $\Delta \mathbf{w} \leftarrow \Delta \mathbf{w} + \mathbf{w}_i^{(r)} - \mathbf{w}_i^{(r-1)}$ 
9   and  $\Delta \ell \leftarrow \Delta \ell + \ell_i^{(r)} - \ell_i^{(r-1)}$ 
10 end
11  $\Delta \mathbf{w}_{[k]} \leftarrow \mathbf{w}_{[k]}^{(R)} - \mathbf{w}_{[k]}$  and  $\Delta \ell_{[k]} \leftarrow \ell_{[k]}^{(R)} - \ell_{[k]}$ 
12  $\Delta \mathbf{w}_k \leftarrow \Delta \mathbf{w}$  and  $\Delta \ell_k \leftarrow \Delta \ell$ 
Output:  $\Delta \mathbf{w}_{[k]}$ ,  $\Delta \mathbf{w}$ ,  $\Delta \ell_{[k]}$ ,  $\Delta \ell$ 

```

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

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

```

1 for  $t = 1, 2, \dots, T$ 
2   for all machines  $k = 1, 2, \dots, K$  in parallel
3      $(\Delta \mathbf{w}_{[k]}, \Delta \mathbf{w}_k) \leftarrow \text{MB-LOCALBCFW}(\mathbf{w}_{[k]}^{(t-1)}, \mathbf{w}^{(t-1)})$ 
4      $\mathbf{w}_{[k]}^{(t)} \leftarrow \mathbf{w}_{[k]}^{(t-1)} + \frac{\beta_K}{K} \Delta \mathbf{w}_{[k]}$ 
5      $\ell_{[k]}^{(t)} \leftarrow \ell_{[k]}^{(t-1)} + \frac{\beta_K}{K} \Delta \ell_{[k]}$ 
6   end
7   reduce  $\mathbf{w}^{(t)} \leftarrow \mathbf{w}^{(t-1)} + \frac{\beta_K}{K} \sum_{k=1}^K \Delta \mathbf{w}^k$ 
8   and  $\ell^{(t)} \leftarrow \ell^{(t-1)} + \frac{\beta_K}{K} \sum_{k=1}^K \Delta \ell^k$ 
9 end

```

7. CoCoA⁺ BCFW

Procedure C: CoCoA⁺-LOCALBCFW: BCFW iterations on machine k

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

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

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

```

1 for  $r = 1, 2, \dots, R$ 
2   | choose  $i \in [k]$  uniformly at random
3   | Solve  $\hat{\mathbf{y}} \leftarrow \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}_i} H_i(\mathbf{y}; \mathbf{w}^{(r-1)})$ 
4   | TODO
5 end
6  $\Delta \mathbf{w}_{[k]} \leftarrow \mathbf{w}_{[k]}^{(R)} - \mathbf{w}_{[k]}$  and  $\Delta \ell_{[k]} \leftarrow \ell_{[k]}^{(R)} - \ell_{[k]}$ 
7  $\Delta \mathbf{w}_k \leftarrow \mathbf{w}^{(R)} - \mathbf{w}$  and  $\Delta \ell_k \leftarrow \ell^{(R)} - \ell$ 
Output:  $\Delta \mathbf{w}_{[k]}$ ,  $\Delta \mathbf{w}$ ,  $\Delta \ell_{[k]}$ ,  $\Delta \ell$ 

```

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

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

```

1 for  $t = 1, 2, \dots, T$ 
2   | for all machines  $k = 1, 2, \dots, K$  in parallel
3   |    $(\Delta \mathbf{w}_{[k]}, \Delta \mathbf{w}_k) \leftarrow \text{LOCALBCFW}(\mathbf{w}_{[k]}^{(t-1)}, \mathbf{w}^{(t-1)})$ 
4   |    $\mathbf{w}_{[k]}^{(t)} \leftarrow \mathbf{w}_{[k]}^{(t-1)} + \beta_K \Delta \mathbf{w}_{[k]}$ 
5   |    $\ell_{[k]}^{(t)} \leftarrow \ell_{[k]}^{(t-1)} + \beta_K \Delta \ell_{[k]}$ 
6   | end
7   | reduce  $\mathbf{w}^{(t)} \leftarrow \mathbf{w}^{(t-1)} + \beta_K \sum_{k=1}^K \Delta \mathbf{w}^k$ 
8   | and  $\ell^{(t)} \leftarrow \ell^{(t-1)} + \beta_K \sum_{k=1}^K \Delta \ell^k$ 
9 end

```
