Supplemental Material for TKDE-2015-05-0359

Kijung Shin, Lee Sael, U Kang

1 Proposed Methods

1.1 Proofs of Update Rules

In this section, we present the proofs of the update rules in Section 3.5 of the main paper. Specifically, we prove the CDTF update rule for L_1 regularization (Theorem 7), the CDTF update rule for the non-negativity constraint (Theorem 8), the SALS update rule for coupled tensor factorization (Theorem 9), and the update rule for bias terms commonly used by CDTF and SALS for the bias model (Theorem 10).

Lemma 1 (Partial Derivative in CDTF*):* For a parameter $a^{(n)}_{i_n k}$, let $\hat{r}_{i_1 \dots i_N} = x_{i_1 \dots i_N} - \sum_{s \neq k} \prod_{l=1}^{N} a^{(l)}_{i_l s}$, $g = -2 \sum_{(i_1,\dots,i_N) \in \Omega^{(n)}_{i_n}} \left(\hat{r}_{i_1 \dots i_N} \prod_{l \neq n} a^{(l)}_{i_l k} \right)$, and $d = 2 \sum_{(i_1,\dots,i_N) \in \Omega^{(n)}_{i_n}} \prod_{l \neq n} (a^{(l)}_{i_l k})^2$, as in the main paper. Then,

$$\frac{\partial \left(\sum_{(i_1,\dots,i_N) \in \Omega} \left(x_{i_1 \dots i_N} - \sum_{s=1}^{K} \prod_{l=1}^{N} a^{(l)}_{i_l s} \right)^2 \right)}{\partial a^{(n)}_{i_n k}} = g + a^{(n)}_{i_n k} d.$$

Proof:

$$\begin{aligned}
&\frac{\partial \left(\sum_{(i_1,\dots,i_N) \in \Omega} \left(x_{i_1 \dots i_N} - \sum_{s=1}^{K} \prod_{l=1}^{N} a^{(l)}_{i_l s} \right)^2 \right)}{\partial a^{(n)}_{i_n k}} \\
&= \sum_{(i_1,\dots,i_N) \in \Omega^{(n)}_{i_n}} 2 \left(x_{i_1 \dots i_N} - \sum_{s=1}^{K} \prod_{l=1}^{N} a^{(l)}_{i_l s} \right) \\
&\quad \times \frac{\partial \left(x_{i_1 \dots i_N} - \sum_{s=1}^{K} \prod_{l=1}^{N} a^{(l)}_{i_l s} \right)}{\partial a^{(n)}_{i_n k}} \\
&= \sum_{(i_1,\dots,i_N) \in \Omega^{(n)}_{i_n}} 2 \left(\hat{r}_{i_1 \dots i_N} - \prod_{l=1}^{N} a^{(l)}_{i_l k} \right) \left(- \prod_{l \neq n} a^{(l)}_{i_l k} \right) \\
&= g + a^{(n)}_{i_n k} d.
\end{aligned}$$

□

Theorem 7: (Correctness of CDTF *with L_1-regularization)* The update rule (12) in the main paper minimizes the loss function (11) with respect to the updated parameter. For an updated parameter $a^{(n)}_{i_n k}$, let $\hat{r}_{i_1 \dots i_N} = x_{i_1 \dots i_N} - \sum_{s \neq k} \prod_{l=1}^{N} a^{(l)}_{i_l s}$, $g = -2 \sum_{(i_1,\dots,i_N) \in \Omega^{(n)}_{i_n}} \left(\hat{r}_{i_1 \dots i_N} \prod_{l \neq n} a^{(l)}_{i_l k} \right)$, and $d = 2 \sum_{(i_1,\dots,i_N) \in \Omega^{(n)}_{i_n}} \prod_{l \neq n} (a^{(l)}_{i_l k})^2$, as in the main paper. Then,

$$\arg\min_{a^{(n)}_{i_n k}} L_{Lasso}(\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)}) = \begin{cases} (\lambda - g)/d & \text{if g > } \lambda \\ -(\lambda + g)/d & \text{if g < -}\lambda \\ 0 & \text{otherwise.} \end{cases}$$

Proof: By Lemma 1,

$$\begin{aligned}
&\frac{\partial L_{Lasso}(\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)})}{\partial a^{(n)}_{i_n k}} \\
&= \frac{\partial \left(\sum_{(i_1,\dots,i_N) \in \Omega} \left(x_{i_1 \dots i_N} - \sum_{s=1}^{K} \prod_{l=1}^{N} a^{(l)}_{i_l s} \right)^2 + \lambda \sum_{l=1}^{N} \|\mathbf{A}^{(l)}\|_1 \right)}{\partial a^{(n)}_{i_n k}} \\
&= g + a^{(n)}_{i_n k} d + \lambda \frac{\partial |a^{(n)}_{i_n k}|}{\partial a^{(n)}_{i_n k}} = \begin{cases} g + a^{(n)}_{i_n k} d + \lambda & \text{if } a^{(n)}_{i_n k} > 0 \\ g + a^{(n)}_{i_n k} d - \lambda & \text{if } a^{(n)}_{i_n k} < 0. \end{cases}
\end{aligned}$$

Case 1: If $g > \lambda$ (> 0), $\frac{\partial L_{Lasso}}{\partial a^{(n)}_{i_n k}} > a^{(n)}_{i_n k} d$. Since $d \geq 0$, $a^{(n)}_{i_n k}$ should be negative for $\frac{\partial L_{Lasso}}{\partial a^{(n)}_{i_n k}}$ to be zero. If $a^{(n)}_{i_n k} < 0$, $\frac{\partial L_{Lasso}}{\partial a^{(n)}_{i_n k}} = g + a^{(n)}_{i_n k} d - \lambda$, and $a^{(n)}_{i_n k} = (\lambda - g)/d$ (< 0) makes $\frac{\partial L_{Lasso}}{\partial a^{(n)}_{i_n k}} = 0$. Since $\frac{\partial^2 L_{Lasso}}{\partial \left(a^{(n)}_{i_n k} \right)^2} = d \geq 0$, $a^{(n)}_{i_n k} = (\lambda - g)/d$ minimizes $L_{Lasso}(\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)})$ with respect to $a^{(n)}_{i_n k}$.

Case 2: Likewise, if $g < -\lambda$ (< 0), $\frac{\partial L_{Lasso}}{\partial a^{(n)}_{i_n k}} < a^{(n)}_{i_n k} d$. Since $d \geq 0$, $a^{(n)}_{i_n k}$ should be positive for $\frac{\partial L_{Lasso}}{\partial a^{(n)}_{i_n k}}$ to be zero. If $a^{(n)}_{i_n k} > 0$, $\frac{\partial L_{Lasso}}{\partial a^{(n)}_{i_n k}} = g + a^{(n)}_{i_n k} d + \lambda$, and $a^{(n)}_{i_n k} = -(\lambda + g)/d$ (> 0) makes $\frac{\partial L_{Lasso}}{\partial a^{(n)}_{i_n k}} = 0$. Since $\frac{\partial^2 L_{Lasso}}{\partial \left(a^{(n)}_{i_n k} \right)^2} = d \geq 0$, $a^{(n)}_{i_n k} = -(\lambda + g)/d$ minimizes $L_{Lasso}(\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)})$ with respect to $a^{(n)}_{i_n k}$.

Case 3: On the other hand, if $-\lambda \leq g \leq \lambda$,

$$\frac{\partial L_{Lasso}}{\partial a^{(n)}_{i_n k}} = \begin{cases} g + a^{(n)}_{i_n k} d + \lambda \geq a^{(n)}_{i_n k} d \geq 0 & \text{if } a^{(n)}_{i_n k} > 0 \\ g + a^{(n)}_{i_n k} d - \lambda \leq a^{(n)}_{i_n k} d \leq 0 & \text{if } a^{(n)}_{i_n k} < 0 \end{cases}$$

That is, $L_{Lasso}(\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)})$ decreases if $a^{(n)}_{i_n k} < 0$ and increases if $a^{(n)}_{i_n k} > 0$, given other parameters. Thus, $L_{Lasso}(\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)})$, which is a continuous

function, is minimized with respect to $a^{(n)}_{i_n k}$ if $a^{(n)}_{i_n k} = 0$. Hence,

$$\arg\min_{a^{(n)}_{i_n k}} L_{Lasso}(\mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)}) = \begin{cases} (\lambda - g)/d & \text{if } g > \lambda \\ -(\lambda + g)/d & \text{if } g < -\lambda \\ 0 & \text{otherwise.} \end{cases}$$

□

Theorem 8: (Correctness of CDTF *with the nonnegativity constraint)* The update rule (14) in the main paper minimizes the loss function (1) with respect to the updated parameter under the nonnegativity constraint. For an updated parameter $a^{(n)}_{i_n k}$, let $\hat{r}_{i_1 ... i_N} = x_{i_1 ... i_N} - \sum_{s \neq k} \prod_{l=1}^{N} a^{(l)}_{i_l s}$, $g = -2 \sum_{(i_1,...,i_N) \in \Omega^{(n)}_{i_n}} \left(\hat{r}_{i_1 ... i_N} \prod_{l \neq n} a^{(l)}_{i_l k} \right)$, and $d = 2 \sum_{(i_1,...,i_N) \in \Omega^{(n)}_{i_n}} \prod_{l \neq n} (a^{(l)}_{i_l k})^2$, as in the main paper. Then,

$$\arg\min_{a^{(n)}_{i_n k} \geq 0} L(\mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)}) = \max\left(\frac{-g}{d + 2\lambda}, 0 \right).$$

Proof: By Lemma 1,

$$\frac{\partial L(\mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)})}{\partial a^{(n)}_{i_n k}} = \frac{\partial \left(\sum_{(i_1,...,i_N) \in \Omega} \left(x_{i_1 ... i_N} - \sum_{s=1}^{K} \prod_{l=1}^{N} a^{(l)}_{i_l s} \right)^2 + \lambda \sum_{l=1}^{N} \|\mathbf{A}^{(l)}\|_F^2 \right)}{\partial a^{(n)}_{i_n k}}$$

$$= g + (d + 2\lambda) a^{(n)}_{i_n k}.$$

Thus,

$$\frac{\partial L}{\partial a^{(n)}_{i_n k}} \begin{cases} > 0 & \text{if } a^{(n)}_{i_n k} > -g/(d + 2\lambda) \\ = 0 & \text{if } a^{(n)}_{i_n k} = -g/(d + 2\lambda) \\ < 0 & \text{otherwise.} \end{cases}$$

Case 1: If $-g/(d+2\lambda) \geq 0$, since $\frac{\partial^2 L}{\partial \left(a^{(n)}_{i_n k} \right)^2} = d + 2\lambda \geq 0$, $L(\mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)})$ is minimized if $a^{(n)}_{i_n k} = -g/(d+2\lambda)$ with respect to $a^{(n)}_{i_n k}$.

Case 2: On the other hand, if $-g/(d + 2\lambda) < 0$, under the constraint that $a^{(n)}_{i_n k} \geq 0$, $L(\mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)})$ is minimized with respect to $a^{(n)}_{i_n k}$ if $a^{(n)}_{i_n k} = 0$. This is because $L(\mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)})$, which is a continuous function, monotonically increases (i.e., $\frac{\partial L}{\partial a^{(n)}_{i_n k}} > 0$) if $a^{(n)}_{i_n k} \geq 0$ $(> -g/(d + 2\lambda))$.

Hence,

$$\arg\min_{a^{(n)}_{i_n k} \geq 0} L(\mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)}) = \max\left(\frac{-g}{d + 2\lambda}, 0 \right).$$

□

Theorem 9: (Correctness of SALS *in Coupled Tensor Factorization)* The update rule (16) in the main paper minimizes (15) with respect to the updated parameters. Let ${}_x\mathcal{R}$ and ${}_y\mathcal{R}$ be the residual tensors for $\mathcal{X}$ and $\mathcal{Y}$, respectively. That is, for C updated parameters $a^{(1)}_{i_1 k_1}, ..., a^{(1)}_{i_1 k_C}$, ${}_x\hat{r}_{i_1 ... i_{N_x}} = x_{i_1 ... i_{N_x}} - \sum_{k=1}^{K} \prod_{n=1}^{N_x} {}_x a^{(n)}_{i_n k} + \sum_{c=1}^{C} \prod_{n=1}^{N_x} {}_x a^{(n)}_{i_n k_c}$ and ${}_y\hat{r}_{i_1 ... i_{N_y}} = y_{i_1 ... i_{N_y}} - \sum_{k=1}^{K} \prod_{n=1}^{N_y} {}_y a^{(n)}_{i_n k} + \sum_{c=1}^{C} \prod_{n=1}^{N_y} {}_y a^{(n)}_{i_n k_c}$. Likewise, let ${}_x\Omega$ and ${}_y\Omega$ be the sets of indices of the observable entries in $\mathcal{X}$ and $\mathcal{Y}$, respectively. Then,

$$\arg\min_{[a^{(1)}_{i_1 k_1}, ..., a^{(1)}_{i_1 k_C}]^T} L_{Coupled}({}_x\mathbf{A}^{(1)}, ..., {}_x\mathbf{A}^{(N_x)}, {}_y\mathbf{A}^{(1)}, ..., {}_y\mathbf{A}^{(N_y)})$$
$$= ({}_x\mathbf{B}^{(1)}_{i_1} + {}_y\mathbf{B}^{(1)}_{i_1} + \lambda \mathbf{I}_C)^{-1} ({}_x\mathbf{c}^{(1)}_{i_1} + {}_y\mathbf{c}^{(1)}_{i_1}),$$

where ${}_x\mathbf{B}^{(1)}_{i_1}$ and ${}_y\mathbf{B}^{(1)}_{i_1}$ are C by C matrices whose entries are

$$({}_x\mathbf{B}^{(1)}_{i_1})_{c_1 c_2} = \sum_{(i_1,...,i_{N_x}) \in {}_x\Omega^{(1)}_{i_1}} \left(\prod_{n \neq 1} {}_x a^{(n)}_{i_n k_{c_1}} \prod_{n \neq 1} {}_x a^{(n)}_{i_n k_{c_2}} \right),$$

$$({}_y\mathbf{B}^{(1)}_{i_1})_{c_1 c_2} = \sum_{(i_1,...,i_{N_y}) \in {}_y\Omega^{(1)}_{i_1}} \left(\prod_{n \neq 1} {}_y a^{(n)}_{i_n k_{c_1}} \prod_{n \neq 1} {}_y a^{(n)}_{i_n k_{c_2}} \right), \forall c_1, c_2,$$

${}_x\mathbf{c}^{(1)}_{i_1}$ and ${}_y\mathbf{c}^{(1)}_{i_1}$ are length C vectors whose entries are

$$({}_x\mathbf{c}^{(1)}_{i_1})_c = \sum_{(i_1,...,i_{N_x}) \in {}_x\Omega^{(1)}_{i_1}} \left({}_x\hat{r}_{i_1 ... i_N} \prod_{n \neq 1} {}_x a^{(n)}_{i_n k_c} \right),$$

$$({}_y\mathbf{c}^{(1)}_{i_1})_c = \sum_{(i_1,...,i_{N_y}) \in {}_y\Omega^{(1)}_{i_1}} \left({}_y\hat{r}_{i_1 ... i_N} \prod_{n \neq 1} {}_y a^{(n)}_{i_n k_c} \right), \forall c$$

and $\mathbf{I}_C$ is the C by C identity matrix.

Proof:

$$\frac{\partial L({}_x\mathbf{A}^{(1)}, ..., {}_x\mathbf{A}^{(N_x)})}{\partial a^{(1)}_{i_1 k_c}} = \frac{\partial L({}_x\mathbf{A}^{(1)}, ..., {}_x\mathbf{A}^{(N_x)})}{\partial ({}_x a^{(1)}_{i_1 k_c})}$$
$$= \frac{\partial \left(\sum_{(i_1,...,i_{N_x}) \in {}_x\Omega} \left(x_{i_1 ... i_{N_x}} - \sum_{s=1}^{K} \prod_{n=1}^{N_x} {}_x a^{(n)}_{i_n s} \right)^2 \right)}{\partial ({}_x a^{(1)}_{i_1 k_c})} + \frac{\partial \lambda \sum_{n=1}^{N_x} \|{}_x\mathbf{A}^{(n)}\|_F^2}{\partial ({}_x a^{(1)}_{i_1 k_c})}$$
$$= \sum_{(i_1,...,i_{N_x}) \in {}_x\Omega^{(1)}_{i_1}} 2 \left(x_{i_1 ... i_{N_x}} - \sum_{s=1}^{K} \prod_{n=1}^{N_x} {}_x a^{(n)}_{i_n s} \right) \times \frac{\partial \left(x_{i_1 ... i_{N_x}} - \sum_{s=1}^{K} \prod_{n=1}^{N_x} {}_x a^{(n)}_{i_n s} \right)}{\partial ({}_x a^{(1)}_{i_1 k_c})} + 2\lambda ({}_x a^{(1)}_{i_1 k_c})$$
$$= \sum_{(i_1,...,i_{N_x}) \in {}_x\Omega^{(1)}_{i_1}} 2 \left(\sum_{s=1}^{C} \prod_{n=1}^{N_x} {}_x a^{(n)}_{i_n k_s} - {}_x\hat{r}_{i_1 ... i_{N_x}} \right) \prod_{n \neq 1} {}_x a^{(n)}_{i_n k_c} + 2\lambda a^{(1)}_{i_1 k_c}.$$

Likewise,

$$\frac{\partial L({}_y\mathbf{A}^{(1)}, ..., {}_y\mathbf{A}^{(N_y)})}{\partial a^{(1)}_{i_1 k_c}} = \sum_{(i_1,...,i_{N_y}) \in {}_y\Omega^{(1)}_{i_1}} 2 \left(\sum_{s=1}^{C} \prod_{n=1}^{N_y} {}_y a^{(n)}_{i_n k_s} - {}_y\hat{r}_{i_1 ... i_{N_y}} \right) \prod_{n \neq 1} {}_y a^{(n)}_{i_n k_c} + 2\lambda a^{(1)}_{i_1 k_c}.$$

From these,

$$\frac{\partial L_{Coupled}}{\partial a^{(1)}_{i_1k_c}} = 0, \forall c, 1 \le c \le C$$

$$\Leftrightarrow \sum_{(i_1,...,i_{N_x})\in {}_x\Omega^{(1)}_{i_1}} \left(\sum_{s=1}^{C} \prod_{n=1}^{N_x} {}_x a^{(n)}_{i_n k_s} - {}_x\hat{r}_{i_1...i_{N_x}} \right) \prod_{n\neq 1} {}_x a^{(n)}_{i_n k_c}$$
$$+ \sum_{(i_1,...,i_{N_y})\in {}_y\Omega^{(1)}_{i_1}} \left(\sum_{s=1}^{C} \prod_{n=1}^{N_y} {}_y a^{(n)}_{i_n k_s} - {}_y\hat{r}_{i_1...i_{N_y}} \right) \prod_{n\neq 1} {}_y a^{(n)}_{i_n k_c}$$
$$+ \lambda a^{(1)}_{i_1k_c} = 0, \forall c$$

$$\Leftrightarrow \sum_{(i_1,...,i_{N_x})\in {}_x\Omega^{(1)}_{i_1}} \left(\sum_{s=1}^{C} \left({}_x a^{(1)}_{i_1k_s} \prod_{n\neq 1} {}_x a^{(n)}_{i_nk_s} \right) \prod_{n\neq 1} {}_x a^{(n)}_{i_nk_c} \right)$$
$$+ \sum_{(i_1,...,i_{N_y})\in {}_y\Omega^{(1)}_{i_1}} \left(\sum_{s=1}^{C} \left({}_y a^{(1)}_{i_1k_s} \prod_{n\neq 1} {}_y a^{(n)}_{i_nk_s} \right) \prod_{n\neq 1} {}_y a^{(n)}_{i_nk_c} \right)$$
$$+ \lambda a^{(1)}_{i_1k_c}$$
$$= \sum_{(i_1,...,i_{N_x})\in {}_x\Omega^{(1)}_{i_1}} \left({}_x\hat{r}_{i_1...i_{N_x}} \prod_{n\neq 1} {}_x a^{(n)}_{i_nk_c} \right)$$
$$+ \sum_{(i_1,...,i_{N_y})\in {}_y\Omega^{(1)}_{i_1}} \left({}_y\hat{r}_{i_1...i_{N_y}} \prod_{n\neq 1} {}_y a^{(n)}_{i_nk_c} \right), \forall c$$

$$\Leftrightarrow ({}_x\mathbf{B}^{(1)}_{i_1} + {}_y\mathbf{B}^{(1)}_{i_1} + \lambda \mathbf{I}_C)[a^{(1)}_{i_1k_1}, ..., a^{(1)}_{i_1k_C}]^T = ({}_x\mathbf{c}^{(1)}_{i_1} + {}_y\mathbf{c}^{(1)}_{i_1}).$$

Hence,

$$\underset{[a^{(1)}_{i_1k_1},...,a^{(1)}_{i_1k_C}]^T}{\arg\min} L_{Coupled}({}_x\mathbf{A}^{(1)}, ..., {}_x\mathbf{A}^{(N_x)}, {}_y\mathbf{A}^{(1)}, ..., {}_y\mathbf{A}^{(N_y)})$$
$$= ({}_x\mathbf{B}^{(1)}_{i_1} + {}_y\mathbf{B}^{(1)}_{i_1} + \lambda \mathbf{I}_C)^{-1}({}_x\mathbf{c}^{(1)}_{i_1} + {}_y\mathbf{c}^{(1)}_{i_1}).$$

□

Theorem 10: (Correctness of the Update Rule for Bias Terms) The update rule (18) in the main paper minimizes (17) with respect to the updated parameter. For an updated parameter $b^{(n)}_{i_n}$, let $\bar{r}_{i_1...i_N} = x_{i_1...i_N} - \sum_{k=1}^{K}\prod_{l=1}^{N} a^{(l)}_{i_lk} - \sum_{l\neq n} b^{(l)}_{i_l} - \mu$, as in the main paper. Then,

$$\underset{b^{(n)}_{i_n}}{\arg\min} L_{Bias}(\mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)}, \mathbf{b}^{(1)}, ..., \mathbf{b}^{(N)})$$
$$= \sum_{(i_1,...,i_N)\in\Omega^{(n)}_{i_n}} \bar{r}_{i_1...i_N}/(\lambda_{\mathbf{b}} + |\Omega^{(n)}_{i_n}|).$$

Proof:

$$\frac{\partial L_{Bias}(\mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)}, \mathbf{b}^{(1)}, ..., \mathbf{b}^{(N)})}{\partial b^{(n)}_{i_n}}$$
$$= \frac{\partial \left(\sum_{(i_1,...,i_N)\in\Omega} \left(x_{i_1...i_N} - \sum_{s=1}^{K}\prod_{l=1}^{N} a^{(l)}_{i_ls} - \sum_{l=1}^{N} b^{(l)}_{i_l} - \mu \right)^2 \right)}{\partial b^{(n)}_{i_n}}$$
$$+ \frac{\partial \left(\lambda_{\mathbf{A}} \sum_{l=1}^{N} \|\mathbf{A}^{(l)}\|^2_F \right)}{\partial b^{(n)}_{i_n}} + \frac{\partial \left(\lambda_{\mathbf{b}} \sum_{l=1}^{N} \|\mathbf{b}^{(l)}\|^2_F \right)}{\partial b^{(n)}_{i_n}}$$
$$= \sum_{(i_1,...,i_N)\in\Omega^{(n)}_{i_n}} 2\left(x_{i_1...i_N} - \sum_{s=1}^{K}\prod_{l=1}^{N} a^{(l)}_{i_ls} - \sum_{l=1}^{N} b^{(l)}_{i_l} - \mu \right)$$
$$\times \frac{\partial \left(x_{i_1...i_N} - \sum_{s=1}^{K}\prod_{l=1}^{N} a^{(l)}_{i_ls} - \sum_{l=1}^{N} b^{(l)}_{i_l} - \mu \right)}{\partial b^{(n)}_{i_n}} + \frac{\lambda_{\mathbf{b}} \sum_{l=1}^{N} \|\mathbf{b}^{(l)}\|^2_F}{\partial b^{(n)}_{i_n}}$$
$$= \sum_{(i_1,...,i_N)\in\Omega^{(n)}_{i_n}} -2(\bar{r}_{i_1...i_N} - b^{(n)}_{i_n}) + 2\lambda_{\mathbf{b}} b^{(n)}_{i_n}$$
$$= 2(\lambda_{\mathbf{b}} + |\Omega^{(n)}_{i_n}|)b^{(n)}_{i_n} - \sum_{(i_1,...,i_N)\in\Omega^{(n)}_{i_n}} 2\bar{r}_{i_1...i_N}.$$

Since $\frac{\partial^2 L_{Bias}}{\partial \left(b^{(n)}_{i_n}\right)^2} = 2(\lambda_{\mathbf{b}} + |\Omega^{(n)}_{i_n}|) \ge 0$, $L_{Bias}(\mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)}, \mathbf{b}^{(1)}, ..., \mathbf{b}^{(N)})$ is minimized with respect to $b^{(n)}_{i_n}$ if $b^{(n)}_{i_n} = \sum_{(i_1,...,i_N)\in\Omega^{(n)}_{i_n}} \bar{r}_{i_1...i_N}/(\lambda_{\mathbf{b}} + |\Omega^{(n)}_{i_n}|)$, which entails $\frac{\partial L_{Bias}}{\partial b^{(n)}_{i_n}} = 0$. Hence,

$$\underset{b^{(n)}_{i_n}}{\arg\min} L_{Bias}(\mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)}, \mathbf{b}^{(1)}, ..., \mathbf{b}^{(N)})$$
$$= \sum_{(i_1,...,i_N)\in\Omega^{(n)}_{i_n}} \bar{r}_{i_1...i_N}/(\lambda_{\mathbf{b}} + |\Omega^{(n)}_{i_n}|).$$

□

1.2 Pseudocodes

We present the pseudocodes of the SALS variants described in Section 3.5 of the main paper.

1.2.1 SALS *for Coupled Tensor Factorization*

Algorithm 6 describes SALS for coupled tensor factorization, where two tensors, denoted by $\mathcal{X}$ and $\mathcal{Y}$, share their first mode without loss of generality. We denote the residual tensors for $\mathcal{X}$ and $\mathcal{Y}$ by ${}_x\mathcal{R}$ and ${}_y\mathcal{R}$, respectively. The lengths of the n-th modes of $\mathcal{X}$ and $\mathcal{Y}$ are denoted by ${}_xI_n$ and ${}_yI_n$, respectively.

Algorithm 6: SALS for Coupled Tensor Factorization

Input : $\mathcal{X}$, $\mathcal{Y}$, K, λ
Output: $\mathbf{A}^{(1)}$, ${}_x\mathbf{A}^{(n)}$ for $n = 2...N_x$, ${}_y\mathbf{A}^{(n)}$ for $n = 2...N_y$

initialize ${}_x\mathcal{R}$, ${}_y\mathcal{R}$, $\mathbf{A}^{(1)}$, ${}_x\mathbf{A}^{(n)}$ and ${}_y\mathbf{A}^{(n)}$ for all $n \ge 2$
for *outer iter = 1..T_{out}* **do**
  for *split iter = 1..$\lceil \frac{K}{C} \rceil$* **do**
    choose $k_1, ..., k_C$ (from columns not updated yet)
    compute ${}_x\hat{\mathcal{R}}$ and ${}_y\hat{\mathcal{R}}$ using (8)
    for *inner iter = 1..T_{in}* **do**
      for $i_1 = 1..I_1 (= {}_xI_1 = {}_yI_1)$ **do**
        update $a^{(1)}_{i_1k_1}, ..., a^{(1)}_{i_1k_C}$ using (16)
      for $n = 2..N_x$ **do**
        for $i_n = 1..{}_xI_n$ **do**
          update ${}_x a^{(n)}_{i_nk_1}, ..., {}_x a^{(n)}_{i_nk_C}$ using (9)
      for $n = 2..N_y$ **do**
        for $i_n = 1..{}_yI_n$ **do**
          update ${}_y a^{(n)}_{i_nk_1}, ..., {}_y a^{(n)}_{i_nk_C}$ using (9)
    update ${}_x\mathcal{R}$ and ${}_y\mathcal{R}$ using (10)

Algorithm 7: SALS for Bias Model

Input : $\mathcal{X}$, K, λ_A, $\lambda_{\mathbf{b}}$
Output: $\mathbf{A}^{(n)}$ for all n, $\mathbf{b}^{(n)}$ for all n, μ

compute μ (the mean of the observable entries of $\mathcal{X}$)
initialize $\mathcal{R}$, $\mathbf{A}^{(n)}$ for all n, and $\mathbf{b}^{(n)}$ for all n
for *outer iter = 1..T_{out}* **do**
  for *split iter = 1..$\lceil \frac{K}{C} \rceil$* **do**
    choose $k_1, ..., k_C$ (from columns not updated yet)
    compute $\hat{\mathcal{R}}$ using (8)
    for *inner iter = 1..T_{in}* **do**
      for *n = 1..N* **do**
        for i_n *= 1..I_n* **do**
          update $a^{(n)}_{i_n k_1}, ..., a^{(n)}_{i_n k_C}$ using (9)
    update $\mathcal{R}$ using (10)
  for *n = 1..N* **do**
    for i_n *= 1..I_n* **do**
      update $b^{(n)}_{i_n}$ using (18)
      update $\mathcal{R}$ using (19)

1.2.2 SALS *for Bias Model*

SALS for the bias model is described in Algorithm 7, where each $(i_1, ..., i_N)$th entry of $\mathcal{R}$ is $r_{i_1 \dots i_n} = x_{i_1 \dots i_N} - \mu - \sum_{n=1}^{N} b^{(n)}_{i_n} - \sum_{k=1}^{K} \prod_{n=1}^{N} a^{(n)}_{i_n k}$, as explained in Section 3.5.5 of the main paper.

2 Optimization on MapReduce

In this section, we present the details of the optimization techniques described in Section 4 of the main paper.

2.1 Local Disk Caching

As explained in Section 4.1 of the main paper, in our MapReduce implementation of CDTF and SALS with local disk caching, $\mathcal{X}$ entries are distributed across machines and cached in their local disk during the map and reduce stages. Algorithm 8 gives the details of the map and reduce stages. The rest of CDTF and SALS runs in the close stage (cleanup stage in Hadoop) using the cached data.

2.2 Direct Communication

In the main paper, we introduce direct communication between reducers using distributed file system to overcome the rigidity of MapReduce model. Algorithm 9 describes the implementation of ${}_m\mathbf{a}^{(n)}_{*k}$ broadcast in CDTF (line 10 of Algorithm 3 in the main paper) based on this communication method.

2.3 Greedy Row Assignment

Our MapReduce implementation of SALS and CDTF uses the greedy row assignment, explained in Section 3.4.3 of the main paper. In this section, we explain our MapReduce implementation of the greedy row assignment. We assume that $\mathcal{X}$ is stored

Algorithm 8: Data distribution in CDTF and SALS with local disk caching

Input : $\mathcal{X}$, ${}_mS_n$ for all m and n
Output: ${}_m\Omega^{(n)}$ entries of $\mathcal{R}(=\mathcal{X})$ for all m and n

```
1  Map(Key k, Value v)
2  begin
3      ((i_1,...,i_N), x_{i_1...i_N}) ← v
4      for n = 1,...,N do
5          find m where i_n ∈ _mS_n
6          emit < (m,n), ((i_1,...,i_N), x_{i_1...i_N}) >
7  end
8  Partitioner(Key k, Value v)
9  begin
10     (m,n) ← k
11     assign < k,v > to machine m
12 end
13 Reduce(Key k, Value v[1..|v|])
14 begin
15     (m,n) ← k
16     create a file on the local disk to cache _mΩ^(n) entries of R
17     foreach ((i_1,...,i_N), x_{i_1...i_N}) ∈ v do
18         write ((i_1,...,i_N), x_{i_1...i_N}) to the file
19 end
```

Algorithm 9: ${}_m\mathbf{a}^{(n)}_{*k}$ broadcast in CDTF

Input : ${}_m\mathbf{a}^{(n)}_{*k}$ (parameters to broadcast)
Output: $\mathbf{a}^{(n)}_{*k}$ (parameters received from others)

begin
  create a data file ${}_mA$ on the distributed file system (DFS)
  write ${}_m\mathbf{a}^{(n)}_{*k}$ on the datafile
  create a dummy file ${}_mD$ on DFS
  while *not all data files are read* **do**
    get the list of dummy files from DFS
    foreach ${}_{m'}D$ in the list **do**
      if ${}_{m'}A$ *are not read* **then**
        read ${}_{m'}\mathbf{a}^{(n)}_{*k}$ from ${}_{m'}A$
end

on the distributed file system. At the first stage, $|\Omega^{(n)}_{i_n}|$ for all n and i_n is computed. Specifically, mappers output $< (n, i_n), 1 >$ for all n for each entry $x_{i_1 \dots i_N}$, and reducers output $< (n, i_n), |\Omega^{(n)}_{i_n}| >$ for all n and i_n by counting the number of values for each key. At the second stage, the outputs are aggregated to a single reducer which runs the rest of Algorithm 5 in the main paper.

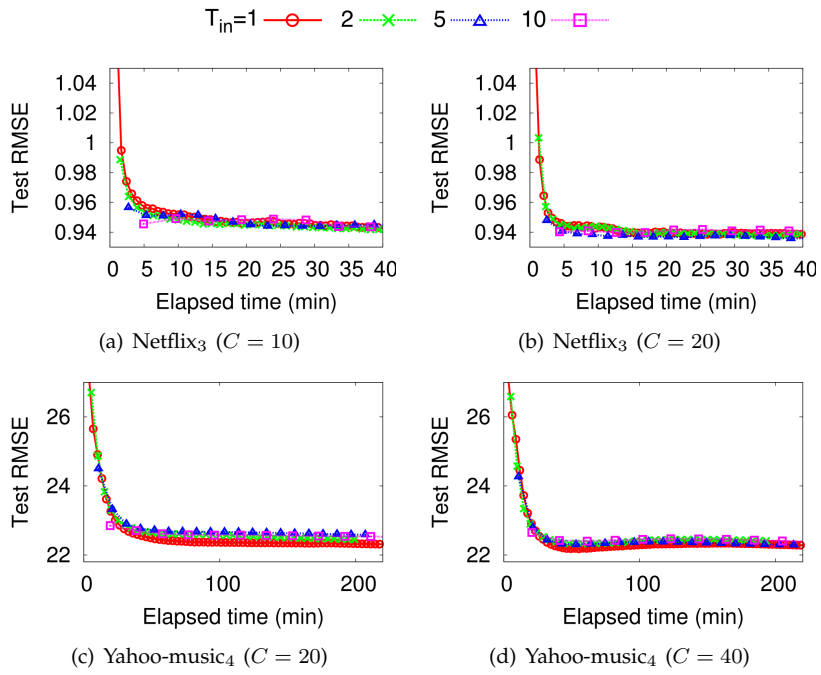
3 Experiments

In this section, we design and conduct additional experiments to answer the following questions:

- How do different numbers of inner iterations (T_{in}) affect the convergence of SALS?
- How do different numbers of columns updated at a time (C) affect the running time of SALS?

3.1 Experimental Settings

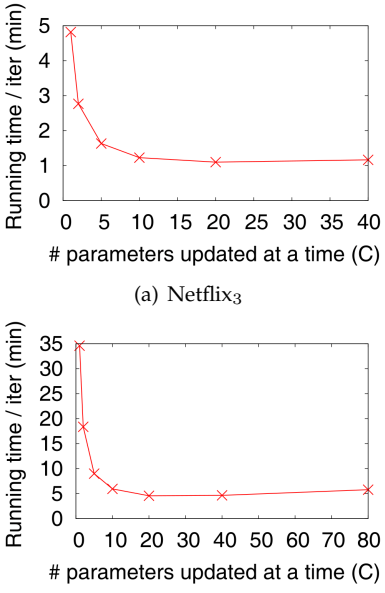
We ran experiments on a 20-node Hadoop cluster. Each node had an Intel Xeon E3-1230 3.3GHz CPU.



(a) Netflix$_3$ ($C = 10$) (b) Netflix$_3$ ($C = 20$)

(c) Yahoo-music$_4$ ($C = 20$) (d) Yahoo-music$_4$ ($C = 40$)

Fig. 14: Effects of T_{in} (i.e., inner iterations) on the convergence of SALS when C (i.e., the number of columns updated at a time) has large enough values. The effects of T_{in} on convergence speed and the quality of converged solutions are marginal.



(a) Netflix$_3$

(b) Yahoo-music$_4$

Fig. 15: Effects of the number of columns updated at a time (C) on the running time of SALS. Running time per iteration decreased until $C = 20$, then started to increase.

The maximum heap size per reducer was set to 8GB. Other experimental settings, including datasets and parameter values (λ and K), were the same as those in the main paper. The number of reducers was set to 20. We used the root mean square error (RMSE) on a held-out test set, which is commonly used in recommender systems, to measure the accuracy, as in the main paper.

3.2 Effects of the Number of Inner Iterations (i.e., T_{in}) on the Convergence of SALS

We compared the convergence properties of SALS with different T_{in} values. Especially, we focused on cases where C (i.e., the number of columns updated at a time) has large enough values. The effect of T_{in} when C is set to one and thus SALS is equivalent to CDTF can be found in the main paper. As seen in Figure 14, the effects of T_{in} on convergence speed and the quality of converged solutions are neither distinct nor consistent. When C is set to one, however, high T_{in} values are preferred (see Section 5.7 of the main paper for detailed experimental results).

3.3 Effects of the Number of Columns Updated at a Time (i.e., C) on Running Time of SALS

We measured the running times per iteration in SALS, as we increased C from 1 to K. As seen in Figure 15, running time per iteration decreased until $C = 20$, then started to increase. As C increases, the amount of disk I/O declines since it depends on the number of times that the entries of $\mathcal{R}$ or $\hat{\mathcal{R}}$ are streamed from disk, which is inversely proportional to C. Conversely, computational cost increases quadratically with regard to C. At small C values, the decrease in the amount of disk I/O was greater and leaded to a downward trend of running time per iteration. The opposite happened at large C values.