Algorithm 1 Nonparametric tensor completion

Input: Noisy and incomplete data tensor $\mathcal{Y}_\Omega$, rank r.

Output: Estimated signal tensor $\hat{\Theta} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$.

for $\pi \in \mathcal{H} = \{-1, \ldots, -\frac{1}{H}, 0, \frac{1}{H}, \ldots, 1\}$ **do**

  Estimate sign tensor $\text{sgn}(\mathcal{Z}_\pi)$ by performing weighted classification using sub-algorithm.

end for

Return estimated tensor $\hat{\Theta} = \frac{1}{2H+1} \sum_{\pi \in \mathcal{H}} \text{sgn}(\mathcal{Z}_\pi)$.

Sub-algorithm: Sign tensor estimation using weighted classification

Input: Noisy and incomplete data tensor $\mathcal{Y}_\Omega$, rank r, target level π.

Output: Sign tensor $\text{sgn}(\mathcal{Z}) \in \{-1, 1\}^{d_1 \times \cdots \times d_K}$ as the estimation of $\text{sgn}(\Theta - \pi)$.

Random initialization of tensor factors $\boldsymbol{A}_k = [\boldsymbol{a}_1^{(k)}, \ldots, \boldsymbol{a}_r^{(k)}] \in \mathbb{R}^{d_k \times r}$ for all $k \in [K]$.

Normalize columns of $\boldsymbol{A}_k$ to have unit-norm for $k \in [K-1]$, and absorb the scales into the columns of $\boldsymbol{A}_K$.

while not convergence **do**

  for $k = 1, \ldots, K$ **do**

    Update $\boldsymbol{A}_k$ while holding others fixed: $\boldsymbol{A}_k \leftarrow \arg\min_{\boldsymbol{A}_k \in \mathbb{R}^{d_k \times r}} \sum_{\omega \in \Omega} |\mathcal{Y}(\omega) - \pi| F(\mathcal{Z}(\omega)\text{sgn}(\mathcal{Y}(\omega) - \pi))$, where $F(\cdot)$ is the large-margin loss, and $\mathcal{Z} = \sum_{s \in [r]} \boldsymbol{a}_s^{(1)} \otimes \cdots \otimes \boldsymbol{a}_s^{(K)}$.

  end for

end while