
Algorithm 1 Nonparametric tensor completion

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

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

- 1: **for** $\pi \in \mathcal{H} = \{-1, \dots, -\frac{1}{H}, 0, \frac{1}{H}, \dots, 1\}$ **do**
 - 2: Estimate sign tensor $\text{sgn}(\mathcal{Z}_\pi)$ by performing weighted classification using sub-algorithm.
 - 3: **end for**
 - 4: 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}_Ω , rank r , target level π .

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

- 5: Random initialization of tensor factors $\mathbf{A}_k = [\mathbf{a}_1^{(k)}, \dots, \mathbf{a}_r^{(k)}] \in \mathbb{R}^{d_k \times r}$ for all $k \in [K]$.
 - 6: Normalize columns of \mathbf{A}_k to have unit-norm for $k \in [K-1]$, and absorb the scales into the columns of \mathbf{A}_K .
 - 7: **while** not convergence **do**
 - 8: **for** $k = 1, \dots, K$ **do**
 - 9: Update \mathbf{A}_k while holding others fixed: $\mathbf{A}_k \leftarrow \arg \min_{\mathbf{A}_k \in \mathbb{R}^{d_k \times r}} \sum_{\omega \in \Omega} |\mathcal{Y}(\omega) - \pi| F(\mathcal{Z}(\omega) \text{sgn}(\mathcal{Y}(\omega) - \pi))$,
where $F(\cdot)$ is the large-margin loss, and $\mathcal{Z} = \sum_{s \in [r]} \mathbf{a}_s^{(1)} \otimes \dots \otimes \mathbf{a}_s^{(K)}$.
 - 10: **end for**
 - 11: **end while**
-