Efficient Algorithm for Mode-k Update in Tensor Completion

Summary

a. Verdict

I have successfully solved the problem. The final answer is a complete algorithm using the Preconditioned Conjugate Gradient (PCG) method to solve the mode-k update efficiently.

b. Method Sketch

The problem requires solving a linear system $\mathcal{A}\,\mathrm{vec}(W) = \mathbf{b}$ of size $nr \times nr$, where $\mathcal{A} = (Z \otimes K)^T SS^T (Z \otimes K) + \lambda(I_r \otimes K)$.

1. **Implicit Matrix-Vector Multiplication:** We compute the product $\mathcal{A}\,\mathrm{vec}(W)$ without forming $\mathcal{A}$. By exploiting the sparsity of the selection matrix S, we calculate the term $(Z \otimes K)^T SS^T (Z \otimes K)\,\mathrm{vec}(W)$ using a sparse MTTKRP (Matricized Tensor Times Khatri-Rao Product) operation on the q observed entries. The cost is $O(qr + n^2 r)$.

2. **Preconditioner:** We approximate the sampling operator SS^T with its expectation ρI_N, where $\rho = q/N$. This yields a structured preconditioner $\mathcal{M} = \rho(Z^T Z \otimes K^2) + \lambda(I_r \otimes K)$.

3. **Efficient Preconditioner Solve:** The system $\mathcal{M}\mathbf{z} = \mathbf{r}$ is equivalent to a generalized Sylvester equation $\rho K^2 W_{new}(Z^T Z) + \lambda K W_{new} = R$. We solve this by simultaneously diagonalizing the kernel K and the Gram matrix $Z^T Z$. The cost per iteration is dominated by basis transformations, $O(n^2 r)$.

The total complexity per PCG iteration is $O(qr + n^2 r)$, which avoids any $O(N)$ computation and is significantly faster than the $O(n^3 r^3)$ direct solver.

Detailed Solution

We wish to solve the linear system for the unknown matrix $W \in \mathbb{R}^{n \times r}$:

$$\mathcal{A}\,\mathrm{vec}(W) = \mathbf{b},$$

where the system matrix and right-hand side are defined as:

$$\mathcal{A} = (Z \otimes K)^T SS^T (Z \otimes K) + \lambda(I_r \otimes K), \quad \mathbf{b} = (I_r \otimes K)\,\mathrm{vec}(B).$$

Here, n is the size of mode k, r is the rank, q is the number of observed entries, and $K \in \mathbb{R}^{n\times n}$ is the positive semi-definite RKHS kernel matrix. The matrix SS^T is a diagonal matrix of size $N \times N$ (where $N = \prod n_i$) with ones at indices corresponding to observed entries and zeros otherwise.

Since $\mathcal{A}$ is symmetric and positive definite (assuming $\lambda > 0$ and K is positive definite), we use the Preconditioned Conjugate Gradient (PCG) method. The efficiency of PCG depends on two components: a fast matrix-vector multiplication and an effective preconditioner.

1. Efficient Matrix-Vector Multiplication

We need to compute $\mathbf{y} = \mathcal{A}\mathbf{x}$ for an arbitrary vector $\mathbf{x} = \mathrm{vec}(W)$ efficiently. The operation decomposes into two terms:

$$\mathbf{y} = \underbrace{(Z \otimes K)^T SS^T (Z \otimes K)\,\mathrm{vec}(W)}_{\text{Term 1}} + \underbrace{\lambda(I_r \otimes K)\,\mathrm{vec}(W)}_{\text{Term 2}}.$$

Computing Term 2: Using the identity $\mathrm{vec}(AXB) = (B^T \otimes A)\,\mathrm{vec}(X)$, we have:

$$\lambda(I_r \otimes K)\,\mathrm{vec}(W) = \mathrm{vec}(\lambda K W).$$

Computing KW costs $O(n^2 r)$.

Computing Term 1:

1. **Forward Projection:** First, compute $(Z \otimes K)\,\mathrm{vec}(W) = \mathrm{vec}(KWZ^T)$.

   Let $U = KW \in \mathbb{R}^{n\times r}$. The full product UZ^T is an $n \times M$ matrix, which is too large to form explicitly ($M \approx N/n$).

   However, the matrix SS^T selects only the entries corresponding to the observed set Ω. Let $\mathcal{V}$ be the sparse matrix containing the values of UZ^T at indices in Ω:

   $$\mathcal{V}_{i,\mu} = \begin{cases} (UZ^T)_{i,\mu} & \text{if } (i,\mu) \in \Omega \\ 0 & \text{otherwise} \end{cases}$$

   For each observed entry indexed by $(i,\mu) \in \Omega$, we compute the dot product of the i-th row of U and the μ-th row of Z:

   $$\mathcal{V}_{i,\mu} = \sum_{j=1}^{r} U_{i,j} Z_{\mu,j}.$$

   Note that $Z_{\mu,j}$ corresponds to the row of the Khatri-Rao product associated with the multi-index of the other modes.

   Cost: $O(qr)$.

2. **Backward Projection:** We now compute $(Z \otimes K)^T \operatorname{vec}(\mathcal{V}) = (Z^T \otimes K) \operatorname{vec}(\mathcal{V})$.

   Using the vectorization property again, this is $\operatorname{vec}(K\mathcal{V}Z)$.

   Let $Y = \mathcal{V}Z \in \mathbb{R}^{n \times r}$. Since $\mathcal{V}$ is sparse, we compute Y using a sparse MTTKRP operation:

$$Y_{i,j} = \sum_{\mu:(i,\mu)\in\Omega} \mathcal{V}_{i,\mu} Z_{\mu,j}.$$

   Cost: $O(qr)$.

3. **Final Multiplication:** Compute KY.

   Cost: $O(n^2 r)$.

Total MatVec Algorithm:

1. $U \leftarrow KW$ $(O(n^2 r))$.
2. Calculate sparse values $\mathcal{V}$ on Ω using U and Z $(O(qr))$.
3. Calculate $Y \leftarrow \mathcal{V}Z$ $(O(qr))$.
4. Result $\leftarrow \operatorname{vec}(K(Y + \lambda W))$ $(O(n^2 r))$.

Total Complexity: $O(qr + n^2 r)$.

2. Preconditioner Design

The matrix $\mathcal{A}$ is ill-conditioned and unstructured due to SS^T. We construct a preconditioner $\mathcal{M}$ by approximating SS^T with a scaled identity matrix. Assuming the observed entries are sampled uniformly at random with probability $\rho = q/N$, we have $\mathbb{E}[SS^T] = \rho I_N$. Replacing SS^T with ρI_N in $\mathcal{A}$:

$$\mathcal{M} = \rho(Z \otimes K)^T (Z \otimes K) + \lambda(I_r \otimes K).$$

Using the mixed-product property $(A \otimes B)(C \otimes D) = (AC \otimes BD)$:

$$(Z \otimes K)^T (Z \otimes K) = (Z^T Z \otimes K^T K) = (Z^T Z \otimes K^2).$$

Thus, the preconditioner is:

$$\mathcal{M} = \rho(Z^T Z \otimes K^2) + \lambda(I_r \otimes K).$$

3. Solving the Preconditioner System

In each PCG iteration, we solve $\mathcal{M}\mathbf{z} = \mathbf{r}_k$, where $\mathbf{z} = \text{vec}(W_{new})$ and $\mathbf{r}_k = \text{vec}(R)$. The equation is:

$$\rho(Z^T Z \otimes K^2)\,\text{vec}(W_{new}) + \lambda(I_r \otimes K)\,\text{vec}(W_{new}) = \text{vec}(R).$$

In matrix form:

$$\rho K^2 W_{new}(Z^T Z) + \lambda K W_{new} = R.$$

Let $G = Z^T Z \in \mathbb{R}^{r\times r}$. This is the Hadamard product of the Gram matrices of the other modes.

$$\rho K^2 W_{new} G + \lambda K W_{new} = R.$$

To solve this efficiently, we use the eigendecompositions of K and G.

1. **Spectral Decompositions (Precomputed):**
   - $K = U_K \Sigma_K U_K^T$, where $\Sigma_K = \text{diag}(\sigma_1, \dots, \sigma_n)$.
   - $G = Q_G \Lambda_G Q_G^T$, where $\Lambda_G = \text{diag}(\gamma_1, \dots, \gamma_r)$.
   - Setup Cost: $O(n^3 + r^3)$.

2. **Diagonalization Step:** Substitute the decompositions into the Sylvester equation. Let $\tilde{W} = U_K^T W_{new} Q_G$ and $\tilde{R} = U_K^T R Q_G$. Multiplying the equation by U_K^T from the left and Q_G from the right, and using $K^2 = U_K \Sigma_K^2 U_K^T$:

   $$\rho \Sigma_K^2 \tilde{W} \Lambda_G + \lambda \Sigma_K \tilde{W} = \tilde{R}.$$

   This system is diagonal. For each element (i, j):

   $$(\rho \sigma_i^2 \gamma_j + \lambda \sigma_i)\tilde{W}_{ij} = \tilde{R}_{ij}.$$

   We solve for $\tilde{W}_{ij}$:

   $$\tilde{W}_{ij} = \frac{\tilde{R}_{ij}}{\sigma_i(\rho \sigma_i \gamma_j + \lambda)}.$$

   (If $\sigma_i = 0$, we set $\tilde{W}_{ij} = 0$ assuming R is in the range of K).

3. **Reconstruction:** Recover $W_{new} = U_K \tilde{W} Q_G^T$.

Preconditioner Solve Complexity:

- Compute $\tilde{R}$: $O(n^2 r + n r^2)$.
- Element-wise division: $O(nr)$.
- Compute W_{new}: $O(n^2 r + n r^2)$.
- Total per iteration: $O(n^2 r)$.

4. Complexity Analysis

Let k_{iter} be the number of PCG iterations.

- **Setup:** $O(n^3)$ to diagonalize K and G.
- **Per Iteration:**
  - MatVec: $O(qr + n^2 r)$.
  - Preconditioner Solve: $O(n^2 r)$.
- **Total Complexity:** $O(n^3 + k_{iter}(qr + n^2 r))$.

Given $n, r < q \ll N$, this approach is highly efficient and avoids any computation scaling with the full tensor size N.

References

1. **Kolda, T. G., & Bader, B. W. (2009).** *Tensor decompositions and applications.* SIAM Review, 51(3), 455-500.
   - `https://doi.org/10.1137/07070111X`
   - Provides the foundational definitions for CP decomposition, MTTKRP, and the Alternating Least Squares framework.
2. **Saad, Y. (2003).** *Iterative methods for sparse linear systems.* SIAM.
   - `https://doi.org/10.1137/1.9780898718003`
   - The authoritative reference for the Preconditioned Conjugate Gradient (PCG) method and the general theory of preconditioning.
3. **Gandy, S., Recht, B., & Yamada, I. (2011).** *Tensor completion and low-n-rank tensor recovery via convex optimization.* Advances in Neural Information Processing Systems.
   - `https://arxiv.org/abs/1012.5660`
   - Supports the use of the sampling operator approximation ($SS^T \approx \rho I$) in the context of tensor completion and reconstruction.