Preconditioned Conjugate Gradient for RKHS-Constrained CP Tensor Decomposition

AI Mathematician — AI Reviewer

Prompted by Scott Armstrong, Julia Kempe and Remi Munos

February 10, 2026

Abstract

We give a fully detailed, self-contained derivation of an efficient preconditioned conjugate gradient (PCG) method for the mode-k subproblem in RKHS-constrained CP tensor decomposition with missing data. We (i) prove the system is SPD under standard assumptions, (ii) derive matrix-vector products in time proportional to the number of observations, and (iii) justify a Kronecker preconditioner with cheap application.

Standing convention about tensor order: The tensor order d is treated as a fixed small constant. When desired, we will also display dependence on d explicitly.

Important note on definiteness: RKHS kernel matrices satisfy $K \succeq 0$ always. Standard CG/PCG requires an SPD matrix, so the main text assumes $K \succ 0$ and $\lambda > 0$. We then state precisely what happens when K is singular in Remark 5.

Contents

1 Problem Statement

Problem 10: *Preconditioned CG for RKHS-constrained CP tensor decomposition*
Domain: Numerical Linear Algebra / Tensor Decomposition
Author: Tamara G. Kolda

Let $\mathcal{T} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be a d-way tensor with missing entries. We compute a CP decomposition of rank r, and for an infinite-dimensional mode k we impose an RKHS constraint using a kernel matrix K. In alternating optimization, all factors except mode k are fixed, and we solve the **mode-k subproblem**.

1.1 Notation

- $N = \prod_{i=1}^{d} n_i$: total number of tensor entries.
- $n \equiv n_k$: size of mode k.
- $M = \prod_{i \neq k} n_i = N/n$: product of all dimensions except k.
- $q \ll N$: number of observed index tuples (observed positions). Assume $n, r < q \ll N$.
- $\Omega = \{(i_1(\ell), \ldots, i_d(\ell))\}_{\ell=1}^{q}$: list of observed index tuples. We assume **unique observed positions** (no duplicates). If duplicates exist, combine them by summing/averaging into a single entry; this yields an equivalent least-squares objective and removes multiplicity without changing the algebra below.
- $T \in \mathbb{R}^{n \times M}$: the mode-$k$ unfolding of $\mathcal{T}$ with missing entries set to zero.
- $S \in \mathbb{R}^{N \times q}$: selection matrix formed from q columns of I_N corresponding to Ω, so that $S^T\vec{(}T)$ extracts the observed entries. We never form S explicitly; we use Ω.
- $Z = A_d \odot \cdots \odot A_{k+1} \odot A_{k-1} \odot \cdots \odot A_1 \in \mathbb{R}^{M \times r}$: Khatri-Rao product of all factor matrices except mode k.
- $B = TZ \in \mathbb{R}^{n \times r}$: MTTKRP.
- $K \in \mathbb{R}^{n \times n}$: RKHS kernel matrix for mode k.
- Unknown: $A_k = KW$ with $W \in \mathbb{R}^{n \times r}$.
- $\lambda > 0$: regularization parameter. I_r: $r \times r$ identity.
- $\otimes$: Kronecker product, $\odot$: Khatri-Rao product, $*$: Hadamard product.

1.2 The linear system

The subproblem yields the linear system

$$\left[(Z \otimes K)^T S S^T (Z \otimes K) + \lambda (I_r \otimes K)\right] \vec{(}W) = (I_r \otimes K)\vec{(}B). \tag{1}$$

Define

$$H := (Z \otimes K)^T S S^T (Z \otimes K) + \lambda (I_r \otimes K), \qquad b := (I_r \otimes K)\vec{(}B).$$

This is an $nr \times nr$ system. A dense direct method costs $O((nr)^3) = O(n^3 r^3)$ and requires $O((nr)^2)$ memory, which is prohibitive for even moderate n, r.

Our goal: solve $H\vec{(}W) = b$ with PCG such that each iteration costs

$$O\big(q(d-1)r + n^2 r\big) = O(qr + n^2 r) \quad \text{when } d \text{ is constant},$$

and with no operations of order N.

2 Background identities

Lemma 1 (Kronecker–vec identity)**.** *For $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times q}$, and $X \in \mathbb{R}^{n \times q}$,*

$$(B \otimes A)\vec{(}X) = \vec{(}AXB^T).$$

Lemma 2 (Transpose)**.** $(A \otimes B)^T = A^T \otimes B^T$.

Lemma 3 (Inverse)**.** *If A, B invertible, then $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$.*

Lemma 4 (Khatri–Rao Gram identity)**.** *If $Z = A_d \odot \cdots \odot A_1$, then*

$$Z^T Z = (A_d^T A_d) * (A_{d-1}^T A_{d-1}) * \cdots * (A_1^T A_1).$$

3 Step 0: Symmetry and positive definiteness

Write

$$H = H_1 + H_2, \qquad H_1 := (Z \otimes K)^T S S^T (Z \otimes K), \qquad H_2 := \lambda (I_r \otimes K).$$

3.1 $H_1 \succeq 0$

For any $\mathbf{x} \in \mathbb{R}^{nr}$,

$$\mathbf{x}^T H_1 \mathbf{x} = \|(S^T (Z \otimes K)\mathbf{x})\|_2^2 \geq 0.$$

Thus H_1 is symmetric psd.

3.2 $H_2 \succ 0$ if $K \succ 0$

Assume $K \succ 0$ and $\lambda > 0$. Then $I_r \otimes K \succ 0$, hence $H_2 = \lambda (I_r \otimes K) \succ 0$.

3.3 Conclusion

Under $K \succ 0$ and $\lambda > 0$,

$$H = H_1 + H_2 \succ 0,$$

so CG/PCG is applicable.

Remark 5 (If K is singular then H is singular)**.** *Suppose $K \succeq 0$ is singular. Then H is* always singular*, regardless of S, Z, λ.*

Indeed, pick any nonzero $u \in \ker(K)$ and any $v \in \mathbb{R}^r$, and set $W = uv^T \in \mathbb{R}^{n\times r}$. Then $KW = 0$, so $(Z \otimes K)\vec{(}W) = \vec{(}KWZ^T) = 0$ and hence $H_1\vec{(}W) = 0$. Also $H_2\vec{(}W) = \lambda\vec{(}KW) = 0$. Therefore $H\vec{(}W) = 0$ with $\vec{(}W) \neq 0$.

Practical SPD remedy: *replace K by $K_\epsilon := K + \epsilon I_n$ with $\epsilon > 0$. Then $K_\epsilon \succ 0$ and the entire analysis below applies. This is the standard stabilization in kernel methods.*

4 Step 1: Fast matrix-vector products with H

Goal: compute $H\vec{(}W)$ in time $O(q(d-1)r + n^2 r)$ without forming H, Z, S, or any dense length-N vectors.

4.1 Regularization term

By Lemma 1,

$$(I_r \otimes K)\vec{(}W) = \vec{(}KW).$$

Thus $H_2\vec{(}W) = \lambda\vec{(}KW)$, cost $O(n^2 r)$ for dense K.

4.2 Data-fitting term

We compute

$$H_1\vec{(}W) = (Z \otimes K)^T SS^T (Z \otimes K)\vec{(}W)$$

by three stages, using only the index list Ω.

Stage (a): compute sampled values $\mathbf{f} = S^T(Z \otimes K)\vec{(}W)$

By Lemma 1,

$$(Z \otimes K)\vec{(}W) = \vec{(}KWZ^T).$$

Define the intermediate matrix (renamed to avoid collision with the preconditioner)

$$U := KW \in \mathbb{R}^{n\times r}.$$

We never form $UZ^T \in \mathbb{R}^{n\times M}$ densely. For each observation ℓ, let $i := i_k(\ell)$ and let $m := m(\ell)$ be its mode-k unfolding column index. Then

$$[UZ^T]_{i,m} = \sum_{j=1}^{r} U_{i,j}\, Z_{m,j}. \tag{2}$$

We compute the row $Z_{m,:}$ on the fly using the Khatri–Rao structure:

$$Z_{m(\ell),:} = A_d(i_d(\ell),:) * \cdots * A_{k+1}(i_{k+1}(\ell),:) * A_{k-1}(i_{k-1}(\ell),:) * \cdots * A_1(i_1(\ell),:). \tag{3}$$

This costs $O((d-1)r)$ per ℓ. Then compute the dot product (2) in $O(r)$. So Stage (a) costs

$$O(n^2 r) \text{ (to form } U = KW) \quad + \quad O(q(d-1)r) \text{ (to compute all needed rows and dots).}$$

It outputs $\mathbf{f} \in \mathbb{R}^q$ with $f_\ell = [S^T\vec{(}UZ^T)]_\ell$.

Stage (b): apply SS^T implicitly

The operator SS^T is the diagonal mask that keeps observed entries and zeroes out others. Thus

$$SS^T(Z \otimes K)\vec{(}W) = S\mathbf{f}.$$

We represent $S\mathbf{f}$ implicitly by the pair $(\Omega, \mathbf{f})$.

Stage (c): compute $(Z \otimes K)^T S\mathbf{f}$ via sparse accumulation

Using Lemma 2 and symmetry of K,

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

Let $\widetilde{T} \in \mathbb{R}^{n\times M}$ be the sparse matrix supported on observed indices with $\widetilde{T}_{i_k(\ell),m(\ell)} = f_\ell$ and all other entries zero. Then $S\mathbf{f} = \vec{(}\widetilde{T})$ and Lemma 1 gives

$$(Z^T \otimes K)\vec{(}\widetilde{T}) = \vec{(}K\widetilde{T}Z).$$

Compute $G := \widetilde{T}Z \in \mathbb{R}^{n\times r}$ without forming $\widetilde{T}$: initialize $G = 0$ and for each observation ℓ compute $Z_{m(\ell),:}$ via (3) and accumulate

$$G_{i_k(\ell),j} \mathrel{+}= f_\ell\, Z_{m(\ell),j} \qquad (j = 1, \ldots, r). \tag{4}$$

This costs $O(q(d-1)r + qr) = O(q(d-1)r)$, and then compute KG at cost $O(n^2 r)$. Finally,

$$H_1\vec{(}W) = \vec{(}KG).$$

4.3 Total matvec cost and no-N guarantee

To compute $H\vec{(}W) = H_1\vec{(}W) + H_2\vec{(}W)$:

1. Compute $U = KW$ once. Cost $O(n^2 r)$.
2. Sample $\mathbf{f} = S^T\vec{(}UZ^T)$ at observed indices using on-the-fly Khatri–Rao rows. Cost $O(q(d-1)r)$.
3. Accumulate $G = \widetilde{T}Z$ via (4), then compute KG. Cost $O(q(d-1)r + n^2 r)$.
4. Output $\vec{(}KG) + \lambda\vec{(}U)$.

Therefore,

$$\boxed{\text{one matvec with } H \text{ costs } O\big(q(d-1)r + n^2 r\big)}$$

which reduces to $O(qr + n^2 r)$ when d is a fixed constant. At no point do we form Z, $Z \otimes K$, S, or any dense object of dimension N or M.

5 Step 2: Preconditioner choice and application

5.1 Definition

Define the Kronecker preconditioner

$$P := (Z^T Z + \lambda I_r) \otimes K. \tag{5}$$

(Here P is *only* the preconditioner; the intermediate matrix is $U = KW$.)

Under $K \succ 0$ and $\lambda > 0$, both factors are SPD, hence $P \succ 0$.

5.2 Algebraic expansion (clarity)

Note that

$$P = (Z^T Z) \otimes K \;+\; \lambda(I_r \otimes K),$$

so P matches the regularization term $\lambda(I_r \otimes K)$ *exactly* and uses the same $Z^T Z$ information as appears in the full-data normal equations. The remaining difference between P and H comes from: (i) the sampling mask SS^T and (ii) the appearance of K^2 in the full-data leading term

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

The choice (5) is a practical compromise: it preserves separability and is cheap to invert.

5.3 Efficient application of P^{-1}

By Lemma 3,

$$P^{-1} = (Z^T Z + \lambda I_r)^{-1} \otimes K^{-1}.$$

For $V \in \mathbb{R}^{n \times r}$, Lemma 1 yields

$$P^{-1}\vec{(}V) = \vec{(}K^{-1} V (Z^T Z + \lambda I_r)^{-1}\big).$$

Thus applying P^{-1} reduces to:

- r solves with K (using Cholesky of K): cost $O(n^2 r)$;
- right-side solve with the $r \times r$ Cholesky of $Z^T Z + \lambda I_r$: cost $O(nr^2)$.

So

$$\boxed{\text{one preconditioner apply costs } O(n^2 r + nr^2).}$$

5.4 Precomputation to build P

Compute $Z^T Z$ without forming Z using Lemma 4:

$$(A_d^T A_d) * \cdots * (A_{k+1}^T A_{k+1}) * (A_{k-1}^T A_{k-1}) * \cdots * (A_1^T A_1).$$

Each $A_i^T A_i$ costs $O(n_i r^2)$, so total $O((\sum_{i \neq k} n_i) r^2)$, plus Cholesky $O(r^3)$ and Cholesky of K in $O(n^3)$.

6 Step 3: Right-hand side computation

The right-hand side is $b = (I_r \otimes K)\vec{(}B) = \vec{(}KB)$.

Compute $B = TZ$ using the observed index list: for each ℓ compute $Z_{m(\ell),:}$ via (3) and accumulate into row $i_k(\ell)$ of B:

$$B_{i_k(\ell),j} \mathrel{+}= \mathcal{T}_{i_1(\ell),\ldots,i_d(\ell)}\, Z_{m(\ell),j}.$$

This costs $O(q(d-1)r)$, then compute KB in $O(n^2 r)$. Hence RHS setup is $O(q(d-1)r + n^2 r)$.

7 Step 4: PCG algorithm

With H SPD, we solve $H\mathbf{w} = b$ by standard PCG with preconditioner P:

- each iteration uses one matvec with H (Section 3) and one apply of P^{-1} (Section 4),
- plus $O(nr)$ vector operations.

8 Step 5: Complexity and avoidance of $O(N)$

Let L be the number of PCG iterations to reach tolerance.

8.1 Per-iteration cost

Per iteration:

$$O\big(q(d-1)r + n^2 r\big) \quad \text{(matvec)} \; + \; O(n^2 r + nr^2) \quad \text{(preconditioner)} \; + \; O(nr),$$

so

$$\boxed{O\big(q(d-1)r + n^2 r + nr^2\big)}.$$

8.2 Precomputation cost

$$O\left(n^3 + q(d-1)r + n^2 r + \Big(\sum_{i \neq k} n_i\Big) r^2 + r^3\right).$$

8.3 Total cost

$$\boxed{O\left(n^3 + q(d-1)r + n^2 r + \Big(\sum_{i \neq k} n_i\Big) r^2 + r^3 + L \cdot \big(q(d-1)r + n^2 r + nr^2\big)\right).}$$

8.4 Comparison with direct methods (corrected)

A dense direct solve for an $nr \times nr$ system costs $O((nr)^3) = O(n^3 r^3)$ arithmetic and $O((nr)^2)$ memory.

Explicitly forming the *data-fitting term* H_1 can be expressed as follows: let $C := S^T(Z \otimes K) \in \mathbb{R}^{q \times nr}$. Then

$$H_1 = (Z \otimes K)^T S S^T (Z \otimes K) = C^T C.$$

The full system matrix is $H = H_1 + \lambda(I_r \otimes K)$, so explicit formation would require forming H_1 (and then adding the dense Kronecker term). Even if one could materialize C, computing $C^T C$ costs $O(q(nr)^2) = O(qn^2 r^2)$ arithmetic and storing H costs $O((nr)^2)$ memory. PCG avoids explicit formation and reduces work to repeated structured matvecs.

8.5 No $O(N)$ computation

No step forms Z (size $M \times r$), $Z \otimes K$ (size $N \times nr$), S (size $N \times q$), or any dense length-N vector. All operations depend only on q observed indices and dense $n \times n$ kernel multiplications.

9 Final theorem

Theorem 6. *Assume $K \succ 0$ and $\lambda > 0$. Then the system (1) can be solved by PCG with:*

1. *SPD system matrix H, so PCG is well-defined.*
2. *Matvec cost $O(q(d-1)r + n^2 r)$ without any order-N computation.*
3. *Kronecker preconditioner $P = (Z^T Z + \lambda I_r) \otimes K$ that is SPD and applies in $O(n^2 r + nr^2)$ time after $O(n^3 + (\sum_{i \neq k} n_i) r^2 + r^3)$ setup.*
4. *Total complexity*

$$O\left(n^3 + q(d-1)r + n^2 r + \Big(\sum_{i \neq k} n_i \Big) r^2 + r^3 + L \cdot \big(q(d-1)r + n^2 r + nr^2 \big) \right),$$

with no computation of order $N = \prod_i n_i$.

If K is singular, then H is singular (Remark 5); a standard practical fix is to replace K by $K + \epsilon I$ with $\epsilon > 0$.

This completes the proof. □