# JENNRICH'S ALGORITHM - A MATLAB IMPLEMENTATION

CHAU Dang Minh

## 1 Jennrich's Algorithm

Let there be a three-order tensor $\mathcal{X} \in \mathbb{R}^{m \times n \times p}$. Suppose that $\mathcal{X}$ has a canonical polyadic decomposition

$$\mathcal{X} = [[\mathbf{A}, \mathbf{B}, \mathbf{C}]] = \sum_{i=1}^{r} \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{m \times r}$, $\mathbf{B} \in \mathbb{R}^{n\mathbf{C}r}$ and $\mathbf{C} \in \mathbb{R}^{p \times r}$. Choose a unit vector $\mathbf{x}_i \in \mathbb{R}^p$ uniformly. We have

$$\mathbf{M}_x = \sum_{i=1}^{p} a_i \mathcal{X}_{:,:,i} = \sum_{j=1}^{r} \langle \mathbf{c}_j, \mathbf{x} \rangle \mathbf{a}_j \mathbf{b}_j^\top = \mathbf{A}\text{diag}(\langle \mathbf{c}_1, \mathbf{x} \rangle, \ldots, \langle \mathbf{c}_r, \mathbf{x} \rangle)\mathbf{B}^\top. \tag{2}$$

Let $\mathbf{D}_x = \text{diag}(\langle \mathbf{c}_1, \mathbf{x} \rangle, \ldots, \langle \mathbf{c}_r, \mathbf{x} \rangle)$ for brevity. We write

$$\mathbf{M}_x = \mathbf{A}\mathbf{D}_x\mathbf{B}^\top. \tag{3}$$

Similarly, we can choose a unit vector $\mathbf{b}$ uniformly and construct the matrix

$$\mathbf{M}_y = \mathbf{A}\mathbf{D}_y\mathbf{B}^\top. \tag{4}$$

If $\mathbf{B}$ is full column rank, we have $\mathbf{B}^\top(\mathbf{B}^\top)^\dagger = \mathbf{I}_r$. Therefore,

$$\mathbf{M}_x\mathbf{M}_y^\dagger = \mathbf{A}\mathbf{D}_x\mathbf{B}^\top(\mathbf{B}^\top)^\dagger\mathbf{D}_y^\dagger\mathbf{A}^\dagger = \mathbf{A}(\mathbf{D}_x\mathbf{D}_y^\dagger)\mathbf{A}^\dagger, \tag{5}$$

where $\mathbf{D}_x\mathbf{D}_y^\dagger = \text{diag}\left(\dfrac{\langle \mathbf{C}_1, \mathbf{a} \rangle}{\langle \mathbf{C}_1, \mathbf{b} \rangle}, \ldots, \dfrac{\langle \mathbf{C}_r, \mathbf{a} \rangle}{\langle \mathbf{C}_r, \mathbf{b} \rangle}\right)$. Since $\mathbf{x}$ and $\mathbf{y}$ are chosen uniformly, the elements of $\mathbf{D}_x\mathbf{D}_y^\dagger$ are distinct with probability 1. Hence, the columns of $\mathbf{A}$ are eigenvectors of $\mathbf{M}_x\mathbf{M}_y^\dagger$. Similarly, if $\mathbf{A}$ is full column rank, then

$$\mathbf{M}_x^\top(\mathbf{M}_y^\top)^\dagger = \mathbf{B}(\mathbf{D}_x\mathbf{D}_y^\dagger)\mathbf{B}^\dagger, \tag{6}$$

which means that the columns of $\mathbf{B}$ are eigenvectors of $\mathbf{M}_x^\top(\mathbf{M}_y^\top)^\dagger$. Finally, we recover $\mathbf{C}$ using

$$\mathcal{X}_{(3)} = \mathbf{C}(\mathbf{B} \odot \mathbf{A})^\top. \tag{7}$$

## 2 Implementation

Function and class dependencies are given in Figure 1. We use Alternative Least Square CP to compare the results. Furthermore, the decomposed factors are standardized to unit-normed columns when compose back to a tensor to compare with the original one. If the difference between the original and the reconstructed tensors is less than a tolerance, the algorithm ran successfully. If the rank determined by Jennrich's algorithm is smaller than that by CP and the difference is larger than the tolerance after two run times, Jennrich's algorithm is concluded to be unable to solve such case. Detailed testcases are given in `testcase_generator.m`. A sample of testing process is illustrated in Figure 2

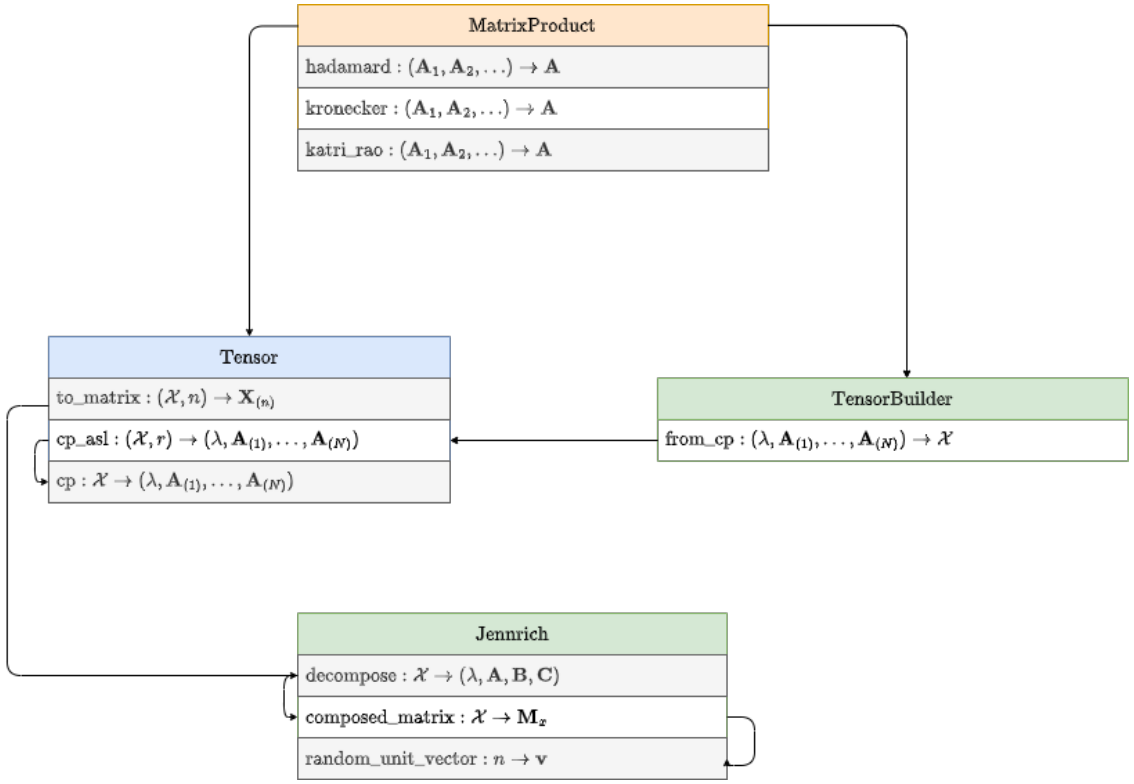


Figure 1: Jennrich's Algorithm Implementation

```
>> testcase_generator
-------------------------------------
Testcase 1
CP decomposition...
Tensor of dimensions:      2     2     2

Max possible rank: 4
Approximating by rank-1 tensor...
Converged after 1 iterations.
Final error 0.000000e+00.
-------------------------------------------------------------
Runtime 1...
Rank determined by Jennrich's: 1
Difference: 0.000000
-------------------------------------
Testcase 2
CP decomposition...
Tensor of dimensions:      3     3     2

Max possible rank: 6
Approximating by rank-1 tensor...
Reached maximum 5000 iterations.
Final error 2.137928e+01.
-------------------------------------------------------------
Approximating by rank-2 tensor...
Reached maximum 5000 iterations.
Final error 1.170680e-03.
-------------------------------------------------------------
Approximating by rank-3 tensor...
Converged after 85 iterations.
Final error 8.807993e-11.
-------------------------------------------------------------
Runtime 1...
There are two equal eigenvalues!
A is not full column-rank, exit...
```

Figure 2: Testing