7.3.1.2 Alternating Least Squares Method

Another group of approaches for low-rank learning is based on alternating minimization. The main concept is to replace the high-dimensional coefficient tensor with a set of low-order tensors or matrices and update these factors alternatively. Specifically, in [15], the bilinear regression model is extended into a more general form as follows:

$$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{B}^{(1)} \cdots \times_M \mathbf{B}^{(M)} \times_{M+1} \mathbf{I}_N + \mathcal{E}, \tag{7.11}$$

where $\mathcal{Y} \in \mathbb{R}^{Q_1 \times \cdots \times Q_M \times N}$, $\mathcal{X} \in \mathbb{R}^{P_1 \times \cdots \times P_M \times N}$, $\mathbf{I}_N$ is an $N \times N$ diagonal matrix and $\mathbf{B}^{(m)} \in \mathbb{R}^{P_m \times Q_m}$, for $m = 1, \ldots, M$, is a coefficient matrix. But as shown in (7.11), there is a clear drawback for this model that the predictor and the response must be in the same order, which is not always the case in practice.

To get the model coefficients $\mathbf{B}^{(m)}$, $m = 1, \ldots, M$, the corresponding optimization can be given as follows:

$$\min_{\mathbf{B}^{(1)}, \ldots, \mathbf{B}^{(M)}} \|\mathcal{Y} - \mathcal{X} \times_1 \mathbf{B}^{(1)} \cdots \times_M \mathbf{B}^{(M)} \times_{M+1} \mathbf{I}_N \|_{\mathrm{F}}^2. \tag{7.12}$$

Similar to the ALS-based algorithm for tensor decomposition, this problem can be solved by updating each of the model parameters $\mathbf{B}^{(1)}, \ldots, \mathbf{B}^{(M)}$ iteratively while others are fixed. Specifically, with respect to each factor $\mathbf{B}^{(m)}$, problem (7.12) boils down to the following subproblem:

$$\min_{\mathbf{B}^{(m)}} \|\mathbf{Y}_{(m)} - \mathbf{B}^{(m)} \tilde{\mathbf{X}}^{(m)} \|_{\mathrm{F}}^2, \tag{7.13}$$

which can be easily solved by the least squares method, where $\tilde{\mathbf{X}}^{(m)} = \mathbf{X}_{(m)} (\mathbf{B}^{(M)} \otimes \cdots \otimes \mathbf{B}^{(m+1)} \otimes \mathbf{B}^{(m-1)} \otimes \cdots \otimes \mathbf{B}^{(1)})^{\mathrm{T}}$. Algorithm 38 provides a summary of the updating procedures.

For ALS-based approaches, the main computational complexity comes from the construction of the matrix $\tilde{\mathbf{X}}^{(m)}$ and its inverse. In addition, for large-scale datasets, the storage of the intermediate variables is also a big challenge.

Algorithm 38: Multilinear tensor regression

Input: predictor $\mathcal{X}$, response $\mathcal{Y}$
Output: $\mathbf{B}^m$, $m = 1, \cdots, M$
Initialize $\mathbf{B}^m$, $m = 1, \cdots, M$
Iterate until convergence:
  For $m = 1, \cdots, M$:
    $\tilde{\mathbf{X}}^{(m)} = \mathbf{X}_{(m)} (\mathbf{B}^{(M)} \otimes \cdots \otimes \mathbf{B}^{(m+1)} \otimes \mathbf{B}^{(m-1)} \otimes \cdots \otimes \mathbf{B}^{(1)})^{\mathrm{T}}$
    $\mathbf{B}^{(m)} = \mathbf{Y}_{(m)} (\tilde{\mathbf{X}}^{(m)})^{\mathrm{T}} (\tilde{\mathbf{X}}^{(m)} (\tilde{\mathbf{X}}^{(m)})^{\mathrm{T}})^{-1}$
  End for