Multidimensional data compression using quantization of low-rank tensor models

---

Tensor decompositions [1,2] can be seen as extensions to data with more than two indexes of matrix decompositions such as singular value decompositions (SVD) and nonnegative matrix factorizations (NMF). They have been applied recently in many applications ranging from chemometrics to recommender systems.

Tensor decompositions such as the Tucker decomposition and the canonical polyadic decomposition (CP) approximate a N-way data block (block of data with N-indexes) through the multilinear product of N factor matrices and a core tensor (another block of data with N-indexes but with different size). If a dataset with dimensions $I_1, I_2, \cdots, I_N$ can be approximated with acceptable accuracy by a tensor decomposition with core of sizes $R_1 \ll I_1, R_2 \ll I_2, \cdots, R_N \ll I_N$ then the number of parameters describing the data with the tensor model will be much smaller than the number of original stored values. The objective of this project is to exploit this reduction of dimensionality to compress the data so that it uses much less memory storage than its original representation. For doing so, scalar quantization [3] with a small number of bits (<16 bits) of the elements of the tensor factors will be considered.

The students will first focus on studying the CP decomposition and they will code either in *python* or in *matlab* a function to generate artificially CP models of low rank (low R_i). They will then study and code the workhorse algorithm to find the CP decomposition, alternating least squares (ALS) [2]. On artificially generated datasets following approximately a low rank CP model, they will evaluate the data reconstruction error when using quantized tensor models for data storage. This evaluation will be carried out using different number of quantization bits {1, …, 16}. A similar evaluation will be done on a real dataset from a chemometrics application.

If good results are obtained, that is, if data can be reconstructed adequately even with high compression rates (small number of quantization bits), then the students will be asked to modify the ALS algorithm so that the tensor decomposition is directly obtained with quantized (integer-valued) factor elements and not with a two-step strategy (first step - obtain the factors with real values, second step – apply quantization).

In the case the students use *python* for coding, they will be asked to deliver a *python notebook* and its *pdf* version containing both the code and a brief presentation of the tensor model, the ALS algorithm, scalar quantization and some comments on the results. In the case the students use *matlab* they will be asked to deliver the *matlab* code along with a separate report with the main points studied in the project and the results.

References :

[1] Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. SIAM review, 51(3), 455-500.

[2] Comon, P., Luciani, X., & De Almeida, A. L. (2009). Tensor decompositions, alternating least squares and other tales. Journal of chemometrics, 23(7-8), 393-405.

[3] Gersho, A., & Gray, R. M. (2012). *Vector quantization and signal compression* (Vol. 159). Springer Science & Business Media.

Project supervisor : Rodrigo Cabral Farias
Assistant professor, Polytech MAM, I3S laboratory