

Numerical methods in Tensor Networks

Stefan Handschuh

In many applications that deal with high dimensional data, it is important to not store the high dimensional object itself, but its representation in a data sparse way. This aims to reduce the storage and computational complexity.

There is a general scheme for representing tensors with the help of sums of elementary tensors, where the summation structure is defined by a graph/network. This scheme allows to generalize commonly used approaches in representing a large amount of numerical data (that can be interpreted as a high dimensional object) using sums of *elementary tensors*. The classification does not only distinguish between elementary tensors and non-elementary tensors, but also describes the number of terms that is needed to represent an object of the *tensor space*. This classification is referred to as *tensor network (format)*.

This work uses the tensor network based approach and describes non-linear block Gauss-Seidel methods (*ALS* and *DMRG*) in the context of the general tensor network framework.

Another contribution of the thesis is the general conversion of different tensor formats. We are able to efficiently change the underlying graph topology of a given tensor representation while using the similarities (if present) of both the original and the desired structure. This is an important feature in case only minor structural changes are required.

In all approximation cases involving iterative methods, it is crucial to find and use a proper initial guess. For linear iteration schemes, a good initial guess helps to decrease the number of iteration steps that are needed to reach a certain accuracy, but it does not change the approximation result. For non-linear iteration schemes, the approximation result may depend on the initial guess. This work introduces a method to successively create an initial guess that improves some approximation results. This algorithm is based on successive rank 1 increments for the *r*-term format.

There are still open questions about how to find the optimal tensor format for a given general problem (e.g. storage, operations, etc.). For instance in the case where a physical background is given, it might be efficient to use this knowledge to create a *good* network structure. There is however, no guarantee that a *better* (with respect to the problem) representation structure does not exist.