Second-order optimization for tensors of fixed tensor-train rank
Michael Psenka (Princeton University), Nicolas Boumal (EPFL)

Problem Setup
Suppose we want to solve the following smooth optimization problem:

$$\min f(X), \text{ where } X \text{ is a high-order tensor.}$$

For tensors of high-order, it is often only tractable to work with the Tensor Train (TT) decomposition of $X$. The size of these decompositions induces a notion of "rank" for tensors: the TT-rank. We then consider the following constrained optimization problem:

$$\min f(X), X \in M_r \quad M_r = \{ X \in \mathbb{R}^{n_1 \times \cdots \times n_d}; \text{ rank}_{TT}(X) = r \}$$

The set $M_r$ is an embedded submanifold, which allows us to use techniques from Riemannian optimization to develop Riemannian versions of first and second-order optimization algorithms.

The first-order tools were developed by M. Steinlechner in his 2016 PhD thesis [1] (see papers of his with D. Kressner and B. Vandereycken [2]).

We add second-order methods to the story.

Curvature & Second-Order Optimization
For Riemannian optimization algorithms, first-order methods use the Riemannian gradient. The Riemannian gradient (a tangent vector) is given by orthogonally projecting the Euclidean gradient to the tangent space of $M_r$ at $X$:

$$\text{grad } f(X) = P_X (\partial f(X))$$

However, the Riemannian Hessian (a linear operator on tangent vectors) requires computing more than a projection:

$$\text{Hess } f(X)[V] = P_X \partial^2 f(X)[V] + P_X (D_V P_X ) \partial f(X)$$

Specifically, the term $P_X(D_V P_X) \partial f(X)$ is closely related to the Weingarten map (or shape operator), which is closely knit with the curvature of the manifold.

Previously, there was no algorithm to efficiently compute the Weingarten map for $M_r$. However, using structure from $M_r$ in several ways, we can efficiently compute the Weingarten map, leading to efficient second-order optimization methods.

Results
We use our method to develop a Riemannian Trust Regions (RTR) algorithm for tensor completion. Below are results comparing RTR to Alternating Least Squares (ALS), Riemannian conjugate gradients (RTTC), and Riemannian trust regions with a finite difference-approximated Hessian (FD-TR). The presented experiments have progressively worse conditioned Hessians at the target point.

Conclusions
By using structure from the Tensor Train decomposition, we design an efficient algorithm to compute the Riemannian Hessian: an essential ingredient for second-order optimization over tensors with a fixed TT-rank.

Our experiments confirm the intuition that, as the conditioning of a problem worsens, having access to the true Hessian adds more and more value in terms of performance.

Future Directions
There are still many problems outside of tensor completion that could benefit from true second-order optimization. For example, solving linear systems that arise from discretizing high-dimensional PDE, as studied in Steinlechner’s thesis.

There are also several recent papers that represent neural network components using a Tensor Train decomposition [3-5]; our algorithm could then be used to efficiently train these components using second-order methods.

References