User Manual for CP-CALS

CHRISTOS PSARRAS, Rheinisch-Westfälische Technische Hochschule Aachen, Germany
LARS KARLSSON, Umeå Universitet, Sweden
RASMUS BRO, University of Copenhagen, Denmark
PAOLO BIENTINESI, Umeå Universitet, Sweden

1 REQUIREMENTS

1.1 Mandatory

- `CMake 3.17.5` or higher. A bash script is provided, to help install CMake 3.17.5 (for Linux) in the `extern` folder and updates the `PATH` environment variable to point to it. Try using it by navigating to the directory where you cloned CALS and running: `source scripts/environment_setup.sh`. Then, using the same terminal session, running `cmake –version` should return version 3.17.5.
- `OpenMP`
- `BLAS/LAPACK` (not required for MATLAB MEX generation). Either of the following libraries has been tested:
  – Intel MKL
  – OpenBLAS

1.2 Optional

- CUDA 11
- MATLAB 2019b

1.3 Compilers

CALS should compile with any compiler supporting C++17 features. It has been tested to compile with `g++-8`, `g++-10` and `clang-10`.

2 COMPILATION

2.1 Cloning

Clone the CP-CALS repository using:

```
git clone https://github.com/HPAC/CP-CALS.git
```

and navigate in the directory of the CP-CALS repository using:

```
cd CP-CALS
```

* Corresponding authors.

Authors' addresses: Christos Psarras, Rheinisch-Westfälische Technische Hochschule Aachen, Aachen, Germany, christos.psarras@rwth-aachen.de; Lars Karlsson, Umeå Universitet, Umeå, Sweden, larsk@cs.umu.se; Rasmus Bro, University of Copenhagen, Copenhagen, Denmark, rb@food.ku.dk; Paolo Bientinesi, Umeå Universitet, Umeå, Sweden, pauldj@cs.umu.se.

2.2 MKL backend

Use the following commands to compile the MKL version of CP-CALS.

```
cmake -S . -B build -DCMAKE_BUILD_TYPE=Release -DWITH_MKL=ON
cmake --build build -j 8
```

2.3 OpenBLAS backend

Use the following commands to compile the OpenBLAS version of CP-CALS.

```
cmake -S . -B build -DCMAKE_BUILD_TYPE=Release -DWITH_OPENBLAS=ON
cmake --build build -j 8
```

3 CUDA SUPPORT

Use the following commands to enable CUDA support for CP-CALS. (works for either MKL or OpenBLAS backends; MKL used in the example)

```
cmake -S . -B build -DCMAKE_BUILD_TYPE=Release -DWITH_MKL=ON -DWITH_CUBLAS=ON
cmake --build build -j 8
```

4 MATLAB INTERFACE

Use the following commands to compile the MATLAB MEX files (no prior compilation of CP-CALS needed).

```
cmake -S . -B build -DCMAKE_BUILD_TYPE=Release -DWITH_MATLAB=ON -DMATLAB_PATH=</path/to/Matlab>
cmake --build build -j 8
```

`</path/to/Matlab>` is the absolute path to the MATLAB install directory.

5 DOXYGEN

To produce the DOXYGEN documentation files for CP-CALS, use the following command:

```
doxygen DOXYGEN
```

6 FUNCTIONS

6.1 C++

The function `cp_cals`, located in the `include/cals.h` header file, is the function that computes multiple Concurrent Alternating Least Squares algorithms, for the Canonical Polyadic Decomposition.
The function definition follows:

```
CalsReport cp_cals(const Tensor &X, KtensorQueue &kt_queue, CalsParams &params);
```

Input.

- `X` : Target tensor to be decomposed.
- `kt_queue` : A queue of references to the input Ktensors, which need to be fitted to the target tensor using ALS.
- `params` : Parameters for the algorithm, of type `CalsParams` .

`CalsParams` is a struct containing all the parameters that can be given to `cp_cals` . Those include:

- `update_method` : Method for updating factor matrices. Possible values {UNCONSTRAINED, NNLS}
- `dim_t max_iterations` : Maximum number of iterations before evicting a model.
- `double tol` : Tolerance of fit difference between consecutive iterations.
- `bool line_search` : Whether to use line search.
- `int line_search_interval` : Number of iterations when line search is invoked.
- `double line_search_step` : Factor for line search.
- `bool cuda` : Use CUDA (to use this verify taht code is compiled with CUDA support).
- `dim_t buffer_size` : Maximum size of the multi-factor columns (sum of ranks of concurrent models)

6.2 MATLAB

The function `cp_cals` in file `matlab/matlab_src/cp_cals.m` is the function that computes CP-CALS for MATLAB. Please, refer to that file for detailed documentation on how to invoke this function within MATLAB.

7 EXAMPLES

`build/src/examples/driver` contains a demonstration of how to use CALS (The -h flag is supported for a look at possible input arguments).

7.1 MATLAB examples

After compiling the MATLAB MEX, one can execute file `matlab/matlab_src/TTB_vs_CALS.m` in MATLAB. This executable performs a comparison of TensorToolbox and CALS. The user first needs to point MATLAB to the CALS MEX and the Tensor Toolbox source code by editing the first two lines of the file.