**OCMA User’s Manual r0.2**

Oct 10, 2018

OCMA factorizes very large matrices using disk-based out-of-core technology and Intel Math Kernel Library (Intel MKL), and it adopts the technology of memory-mapped file I/O.

**Installation**

After downloading, the users will see two files: ocma-0.1.zip for Windows system, and ocma-0.1.tar.gz for Linux/Unix system.

After decompression, one will see two folders: src and bin. The folder src contains all the source code, and the folder bin contains the compiled binary executable. Under the folder bin, two small matrices, SM10 and M3\_4 for testing purpose are also provided. M3\_4 is a 3 x 4 matrix and SM10 is a 10 x 10 symmetric matrix. The suffix \_Single, \_Double and \_Text indicate the format of single precision, double precision, and text file, respectively.

The program is written in the C language, and it uses Intel Math Kernel Library. So, in order to compile the source code, one needs to install C compiler, e.g., the Intel C/C++ Compilers, the GNU C Compiler or Visual Studio C Compiler. Additionally, one needs to install Intel MKL.

When C compiler and MKL are available in the system, one can enter the src folder and modify the makefile based on the C compiler and the MKL paths; then type make or nmake to compile. After that, the ocma executable will be generated in the bin folder.

Compiling options

We have provided two options for compiling the executable.

> make (or nmake) static

will generate an executable that doesn’t rely on the Intel MKL libraries. However, it still relies on libiomp5.so (Linux) or libiomp5md.dll (Windows) for multi-threading OpenMP settings. We have provided a version for Windows 7 and a version for CentOS 7.3 in the release.

> make (or nmake) dynamic

will generate a smaller executable that doesn’t contain the Intel MKL libraries. As such, the executable will be dynamically linked to the libraries during the course.

**Usage**

* Eigen-decomposition for Symmetric Matrices

> ocma eigen single/double disk/memory n A E Q

Parameter explanation:

single/double: Specifies whether using single or double precision.

disk/memory: Specifies whether using disk or memory.

n (input): The row number and column number of matrix A.

A (input): The filename of the file that stores matrix A. The size of the file should be n\*n\*sizeof(float/double).

E (output): The filename of the file that stores the eigenvalues of matrix A. The file size is n\*sizeof(float/double).

Q (output): The filename of the file that stores the eigenvectors of matrix A. The file size is n\*n\*sizeof(float/double).

* Singular Value Decomposition

> ocma singular single/double disk/memory m n A S U V

Parameter explanation:

single/double: Specifies whether using single or double precision.

disk/memory: Specifies whether using disk or memory.

m (input): The row number of matrix A.

n (input): The column number of matrix A.

A (input): The filename of the file that stores matrix A. The size of the file should be m\*n\*sizeof(float/double).

S (output): The filename of the file that stores the singular values of matrix A. The size of the file is min(m,n)\*sizeof(float/double).

U (output): The filename of the file that stores the left-singular vectors of matrix A. The size of the file is m\*min(m,n)\*sizeof(float/double).

V (output): The filename of the file that stores the right-singular vectors of matrix A. The size of the file is n\*min(m,n)\*sizeof(float/double).

* Partial Singular Value Decomposition

> ocma singularpart single/double disk/memory m n k A S U V

Parameter explanation:

single/double: Specifies whether using single or double precision.

disk/memory: Specifies whether using disk or memory.

m (input): The row number of matrix A.

n (input): The column number of matrix A.

k (input): The number of singular values. k <= min(m,n) is required.

A (input): The filename of the file that stores matrix A. The size of the file should be m\*n\*sizeof(float/double).

S (output): The filename of the file that stores the singular values of matrix A. The size of the file is k\*sizeof(float/double).

U (output): The filename of the file that stores the left-singular vectors of matrix A. The size of the file is m\*k\*sizeof(float/double).

V (output): The filename of the file that stores the right-singular vectors of matrix A. The size of the file is n\*k\*sizeof(float/double).

* Matrix Format Conversion

> ocma format single/double txt2bin/bin2txt m n A B

Parameter explanation:

single/double: Specifies whether using single or double precision format.

txt2bin/bin2txt: Specifies to convert text to binary format or in reverse.

m (input): The row number of matrix A.

n (input): The column number of matrix A.

A (input): The filename of the file that stores matrix A in text/binary format.

B (output): The filename of the file that stores the matrix in binary/text format.

* Multi-threading

Please use the environment variable MKL\_NUM\_THREADS to specify number of threads. If this is not specified, OCMA will automatically select a number of threads based on available CPU cores in the system.

Note:

For all file names, the file path must be specified if the file being used is not present in the current working directory.

Contacts

Zhi Xiong, zxiong@stu.edu.cn

Quan Long, quan.long@ucalgary.ca

Copyright License (MIT Open Source)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Acknowledgements

Funding. This work was supported by Natural Science Foundation of Guangdong Province (2018A030313438) and Special Funds for Discipline and Specialty Construction of Guangdong Higher Education Institutions (2016KTSCX040) to Z.X.; University of Calgary Startup and URGC Seed Grants, Canada Foundation for Innovation, and NSERC Discovery Grant to Q.L.; NSERC Discovery Grant to Q.Z.; and NIH to G.D.L.C. (R01 NIGMS 101219) and X.S. (R15HG009565).