Read me

Compiling

To compile targets type **make** on terminal.

To run the program call **make run** on terminal

To delete binary files call **make clean** on terminal

The code works for the given data only

Note: the code uses an external library Eigen which is found on the following link

http://eigen.tuxfamily.org/index.php?title=Main_Page

Below are some of the Information used
version used 3.37

Eigen Library

Requirements

Eigen doesn't have any dependencies other than the C++ standard library.

We use the CMake build system, but only to build the documentation and unit-tests, and to automate installation. If you just want to use Eigen, you can use the header files right away. There is no binary library to link to, and no configured header file. Eigen is a pure template library defined in the headers.

License

Eigen is Free Software. Starting from the 3.1.1 version, it is licensed under the MPL2, which is a simple weak copyleft license. Common questions about the MPL2 are answered in the official MPL2 FAQ.

Earlier versions were licensed under the LGPL3+.

Note that currently, a few features rely on third-party code licensed under the LGPL: SimplicialCholesky, AMD ordering, and constrained_cg. Such features can be explicitly disabled by compiling with the EIGEN_MPL2_ONLY preprocessor symbol defined. Furthermore, Eigen provides interface classes for various third-party libraries (usually recognizable by the <Eigen/*Support> header name). Of course you have to mind the license of the so-included library when using them.

Virtually any software may use Eigen. For example, closed-source software may use Eigen without having to disclose its own source code. Many proprietary and closed-source software projects are using Eigen right now, as well as many BSD-licensed projects.

See the **MPL2 FAQ** for more information, and do not hesitate to contact us if you have any questions.

Compiler support

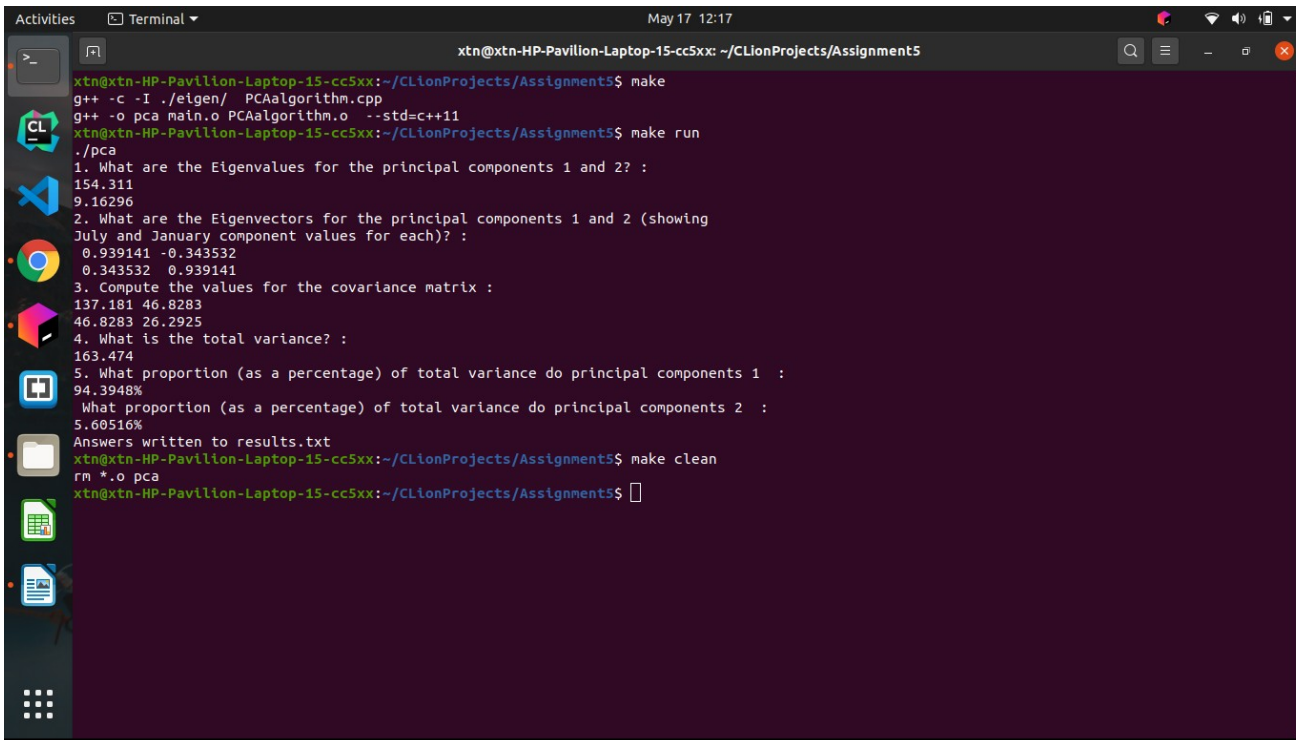
Eigen is standard C++98 and so should theoretically be compatible with any compliant compiler. Whenever we use some non-standard feature, that is optional and can be disabled.

Eigen is being successfully used with the following compilers:

- GCC, version 4.8 and newer. Older versions of gcc might work as well but they are not tested anymore.
- MSVC (Visual Studio), 2012 and newer. Be aware that enabling IntelliSense (/FR flag) is known to trigger some internal compilation errors. The old 3.2 version of Eigen supports MSVC 2010, and the 3.1 version supports MSVC 2008.
- Intel C++ compiler. Enabling the `-inline-forceinline` option is highly recommended.
- LLVM/CLang++, version 3.4 and newer. (The 2.8 version used to work fine, but it is not tested with up-to-date versions of Eigen)
- XCode 7 and newer. Based on LLVM/CLang.
- MinGW, recent versions. Based on GCC.
- QNX's QCC compiler.

Regarding performance, Eigen performs best with compilers based on GCC or LLVM/Clang. See the page for some known compilation issues.

The screen shots below show examples


Activities
Terminal
May 17 12:17
xtn@xtn-HP-Pavilion-Laptop-15-cc5xx: ~/CLionProjects/Assignment5
xtn@xtn-HP-Pavilion-Laptop-15-cc5xx:~/CLionProjects/Assignment5$ make
g++ -c -I ./eigen/ PCAalgorithm.cpp
g++ -o pca main.o PCAalgorithm.o --std=c++11
xtn@xtn-HP-Pavilion-Laptop-15-cc5xx:~/CLionProjects/Assignment5$ make run
./pca
1. What are the Eigenvalues for the principal components 1 and 2? :
154.311
9.16296
2. What are the Eigenvectors for the principal components 1 and 2 (showing
July and January component values for each)? :
0.939141 -0.343532
0.343532 0.939141
3. Compute the values for the covariance matrix :
137.181 46.8283
46.8283 26.2925
4. What is the total variance? :
163.474
5. What proportion (as a percentage) of total variance do principal components 1 :
94.3948%
What proportion (as a percentage) of total variance do principal components 2 :
5.60516%
Answers written to results.txt
xtn@xtn-HP-Pavilion-Laptop-15-cc5xx:~/CLionProjects/Assignment5$ make clean
rm *.o pca
xtn@xtn-HP-Pavilion-Laptop-15-cc5xx:~/CLionProjects/Assignment5$