cholmod-extra Documentation

Release 0.1

Jaakko Luttinen

Oct 10, 2018

Contents

1 Introduction **1**

2 Installation **3**

3 Functions **5**

3.1 Sparse inverse 5

4 License and copyright **7**

Bibliography **9**

CHAPTER 1

Introduction

This package provides additional functions for SuiteSparse, its CHOLMOD module in particular.

CHAPTER 2

Installation

The module depends on SuiteSparse, thus install it if you do not have it installed. Instructions can be found at http://www.cise.ufl.edu/research/sparse/SuiteSparse/. After installing the dependencies, clone the Git repository:

```
git clone https://github.com/jluttine/cholmod-extra.git
```

Compile and install the module:

```
cd cholmod-extra
make
make install
```

TODO: Check the installation directory in Makefile!

This documentation can be found in Docs/ folder. The documentation source files are readable as such in reStructuredText format. If you have Sphinx installed, the documentation can be compiled to, for instance, HTML or PDF using

```
cd Docs
make html
make latexpdf
```

The documentation can be found also at http://cholmod-extra.readthedocs.org/ in various formats.

CHAPTER 3

Functions

3.1 Sparse inverse

cholmod_sparse ***cholmod_spinv** (cholmod_factor **L*, cholmod_common **Common*)

Return the sparse inverse given the Cholesky factor. The sparse inverse contains elements from the inverse matrix but has the same sparsity structure as the Cholesky factor (symbolically).

Although the inverse of a sparse matrix is dense in general, it is sometimes sufficient to compute only some elements of the inverse. For instance, in order to compute $\mathrm{tr}(\mathbf{K}^{-1}\mathbf{A})$, it is sufficient to find those elements of $\mathbf{K}^{-1}$ that are non-zero in $\mathbf{A}^{\mathrm{T}}$. If $\mathbf{A}^{\mathrm{T}}$ has the same sparsity structure as $\mathbf{K}$ (e.g., $\mathbf{A}^{\mathrm{T}} = \partial\mathbf{K}/\partial\theta$), one only needs to compute those elements of the inverse $\mathbf{K}^{-1}$ that are non-zero in $\mathbf{K}$. These elements can be computed using an efficient algorithm if $\mathbf{K}$ is symmetric positive-definite *[Takahashi:1973]*. The resulting sparse matrix is called the sparse inverse.

The algorithm for computing the sparse inverse can be derived as follows *[Vanhatalo:2008]*. Denote the inverse as $\mathbf{Z} = \mathbf{K}^{-1}$ and the Cholesky decomposition as $\mathbf{L}\mathbf{L}^{\mathrm{T}} = \mathbf{K}$, where $\mathbf{L}$ is a lower triangular matrix. We have the identity

$$\mathbf{Z}\mathbf{L} = \mathbf{L}^{-\mathrm{T}}. \tag{3.1}$$

Taking the diagonal elements of the Cholesky factor, $\mathbf{\Lambda} = \mathrm{mask}(\mathbf{L}, \mathbf{I})$, the equation (3.1) can be written as

$$\mathbf{Z}\mathbf{\Lambda} + \mathbf{Z}(\mathbf{L} - \mathbf{\Lambda}) = \mathbf{L}^{-\mathrm{T}}.$$

Subtracting the second term on the left and multiplying by $\mathbf{\Lambda}^{-1}$ from the right yields

$$\mathbf{Z} = \mathbf{L}^{-\mathrm{T}}\mathbf{\Lambda}^{-1} - \mathbf{Z}(\mathbf{L} - \mathbf{\Lambda})\mathbf{\Lambda}^{-1}. \tag{3.2}$$

One can also use Cholesky decomposition of the form $\tilde{\mathbf{L}}\mathbf{D}\tilde{\mathbf{L}}^{\mathrm{T}} = \mathbf{K}$, where $\tilde{\mathbf{L}}$ has unit diagonal and $\mathbf{D}$ is a diagonal matrix. In that case, equation (3.2) transforms to

$$\mathbf{Z} = \tilde{\mathbf{L}}^{-\mathrm{T}}\mathbf{D}^{-1} - \mathbf{Z}(\tilde{\mathbf{L}} - \mathbf{I}).$$

These formulae can be used to solve the inverse recursively. The recursive update formulae are shown for the supernodal factorization, because the update formulae for the simplicial factorization can be seen as a special case, and possible permutations are ignored.

The inverse is computed for each supernodal block at a time starting from the lower right corner. Now, consider one iteration step. Let $\mathbf{Z}_C$ denote the lower right part of the inverse which has already been computed. The supernodal block that is updated consists of $\mathbf{Z}_A$ and $\mathbf{Z}_B$ as

$$\mathbf{Z} = \begin{bmatrix} \ddots & \vdots & \vdots \\ \cdots & \mathbf{Z}_A & \mathbf{Z}_B^{\mathrm{T}} \\ \cdots & \mathbf{Z}_B & \mathbf{Z}_C \end{bmatrix},$$

where $\mathbf{Z}_A$ and $\mathbf{Z}_C$ are square matrices on the diagonal. Using the same division to blocks for $\mathbf{L}$, from (3.2) follows that

$$\begin{aligned} \mathbf{Z}_B &= -\mathbf{Z}_C \mathbf{L}_B \mathbf{L}_A^{-1}, \\ \mathbf{Z}_A &= \mathbf{L}_A^{-\mathrm{T}} \mathbf{L}_A^{-1} - \mathbf{Z}_B^{\mathrm{T}} \mathbf{L}_B \mathbf{L}_A^{-1}. \end{aligned}$$

For the first iteration step, the update equation is $\mathbf{Z}_A = \mathbf{L}_A^{-\mathrm{T}} \mathbf{L}_A^{-1}$.

Instead of computing the full inverse using this recursion, it is possible to gain significant speed-up if one computes the sparse inverse, because then it is sufficient to compute only those elements that are symbolically non-zero in $\mathbf{L}$. It follows that one can discard those rows from the block B that are symbolically zero in $\mathbf{L}_B$. Also, the same rows and the corresponding columns can be discarded from the block C. Thus, the blocks B and C are effectively very small for the numerical matrix product computations. For the simplicial factorization, each block is one column wide, that is, $\mathbf{Z}_A$ is a scalar and $\mathbf{Z}_B$ is a vector.

For $\mathbf{K} = \tilde{\mathbf{L}} \mathbf{D} \tilde{\mathbf{L}}^{\mathrm{T}}$ factorization, the update equations are

$$\begin{aligned} \mathbf{Z}_B &= -\mathbf{Z}_C \tilde{\mathbf{L}}_B \tilde{\mathbf{L}}_A^{-1}, \\ \mathbf{Z}_A &= \tilde{\mathbf{L}}_A^{-\mathrm{T}} \mathbf{D}_A^{-1} \tilde{\mathbf{L}}_A^{-1} - \mathbf{Z}_B^{\mathrm{T}} \tilde{\mathbf{L}}_B \tilde{\mathbf{L}}_A^{-1}, \end{aligned}$$

and for the first iteration step, $\mathbf{Z}_A = \tilde{\mathbf{L}}_A^{-\mathrm{T}} \mathbf{D}_A \tilde{\mathbf{L}}_A^{-1}$.

The following methods have been implemented in cholmod-extra.

Table 1: Implemented sparse inverse methods.

	$\mathbf{L}\mathbf{L}^{\mathrm{T}}$			$\tilde{\mathbf{L}}\mathbf{D}\tilde{\mathbf{L}}^{\mathrm{T}}$		
	Real	Complex	Zomplex	Real	Complex	Zomplex
Simplicial	no	no	no	yes	no	no
Supernodal	yes	no	no	no	no	no

CHAPTER 4

License and copyright

CHOLMOD Extra Module (cholmod-extra) including this documentation is licensed under Version 3.0 of the GNU General Public License. See LICENSE file for a text of the license or visit http://www.gnu.org/copyleft/gpl.html.

For this documentation, copyright (C) 2012 Jaakko Luttinen.

Bibliography

[Takahashi:1973] Takahashi K, Fagan J, and Chen M-S (1973). Formation of a sparse bus impedance matrix and its application to short circuit study. In *Power Industry Computer Application Conference Proceedings*. IEEE Power Engineering Society.

[Vanhatalo:2008] Vanhatalo J and Vehtari A (2008). Modelling local and global phenomena with sparse Gaussian processes. In *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence*. AU AI Press.

Index

C

cholmod_spinv (C++ function), 5