

Class 4 (Monday 31 October)

Contents

- [Comparing dense and sparse storage](#)
- [Comparing sparse formats](#)
- [When is a sparse matrix worth it?](#)

These tasks are designed to be worked on in the practical class on Monday 31 October.

In this class, we will make heavy use of the [finite difference code for solving a Poisson problem](#) that we wrote in lectures. You can find this code [here](#).

Comparing dense and sparse storage

Copy the codes we that wrote to generate the matrix in dense and sparse formats.

For a sensible range of N , measure how much memory a dense matrix and a COO matrix use. You can print the amount of memory a dense matrix uses by running:

```
import numpy as np

a = np.zeros(...)
print(a.nbytes)
```

You can print the amount of memory a COO matrix uses by running:

```
from scipy.sparse import coo_matrix

b = coo_matrix(...)
print(b.row.nbytes + b.col.nbytes + b.data.nbytes)
```

Create a plot that shows the memory used by the dense and COO sparse format against N . What do you notice?

Use `scipy.sparse.linalg.spsolve` and `numpy.linalg.solve` to solve the problem for a range of values of N . Plot the time both solution methods taks against N . What do you notice?

Comparing sparse formats

SciPy can convert between different sparse formats, for example

```
from scipy.sparse import coo_matrix

matrix = coo_matrix(...)
csr_mat = matrix.tocsr()
```

For a range of values of N , measure how much storage space is needed to store the matrix for the Poisson problem if the matrix is stored as a COO matrix, a CSR matrix, or a CSC matrix. For a CSR matrix, the amount of memory used can be printed by running

```
from scipy.sparse import coo_matrix

matrix = coo_matrix(...)
csr_mat = matrix.tocsr()
print(c.data.nbytes + c.indices.nbytes + c.indptr.nbytes)
```

Make a plot showing the amount of memory needed vs N . Which format is the most memory efficient?

Optional extension: Scipy also supports LIL, DIA, DOK, and BSR sparse formats. Add these to your plot.

When is a sparse matrix worth it?

In this section, we will investigate how many zeros we need a matrix to have for sparse storage to be worth doing.

Create a 10 by 10 matrix which is all zeros except for M random numbers in random positions. Measure the amount of memory needed to store this as a dense and different sparse matrix formats. Make a plot showing the amount of memory needed against M . What proportion of the matrix needs to be zeros for sparse storage to use less space? When are different sparse formats more efficient?

Repeat this with a 40 by 40 matrix. What proportion of the matrix needs to be zeros for sparse storage to use less space? When are different sparse formats more efficient?

By Timo Betcke & Matthew Scroggs

© Copyright 2020-22.