

Программа для вычисления максимально скоррелированных мод и ЭОФ

Мария Тарасевич, mashatarasevich@gmail.com

Написанная на языке Python 3 программа состоит из двух файлов: `main.py` и `supersvd.py`. В файле `supersvd.py` находится алгоритм вычисления максимально скоррелированных мод, а в `main.py` — вспомогательный код, который анализирует ключи запуска программы, делает чтение входных данных из файлов, а также записывает в выходные файлы результаты работы алгоритма.

Функцию `supersvd` можно напрямую использовать из кода на Python, в этом случае не обязательно сохранять массивы в виде файлов на диске.

1 Описание

Функция `supersvd` по двум заданным наборам пространственно-временных полей строит матрицу ковариации, а затем вычисляет её неполное сингулярное разложение.

Функция `supersvd` принимает на вход 2 обязательных аргумента (два поля, максимально скоррелированные моды которых мы ищем) и 2 опциональных параметра: количество пар максимально скоррелированных мод (по умолчанию 3) и значение переключателя режима вычитания из поля его среднего по времени значения (по умолчанию `True`, то есть из поля *вычитается* его среднее по времени значение).

Пусть $X(t), Y(t)$ — два меняющихся во времени поля, максимально скоррелированные моды которых мы ищем, причём $\dim(X) = nT \times nX^1$ и $\dim(Y) = nT \times nY$, где nX и nY могут быть одним или несколькими измерениями массивов (в случае среднемесячных данных INMCM nX и $nY = 120 \times 180$). Функция `supersvd` вычисляет разложение вида:

$$\begin{aligned} X(t) &= \bar{X} + XV_1XC_1(t) + XV_2XC_2(t) + \dots + XV_kXC_k(t) + \dots, \\ Y(t) &= \bar{Y} + YV_1YC_1(t) + YV_2YC_2(t) + \dots + YV_kYC_k(t) + \dots, \end{aligned} \quad (1)$$

где

$$\bar{X} = \frac{1}{nT} \sum_{t=1}^{nT} X(t), \quad \bar{Y} = \frac{1}{nT} \sum_{t=1}^{nT} Y(t),$$

а k — количество пар максимально скоррелированных мод. В (1) каждое новое слагаемое получается максимизацией корреляции между $XC_k(t)$ и $YC_k(t)$, а XV_k, YV_k — два семейства ортогональных пространственных мод.

Моды XV_k, YV_k являются левыми и правыми сингулярными векторами матрицы ковариации

$$C = \frac{1}{nT} \sum_{t=1}^{nT} (X(t) - \bar{X})(Y(t) - \bar{Y})^T.$$

Функция `supersvd` возвращает:

- массивы `x_coeff`, `y_coeff` временных коэффициентов $XC(t), YC(t)$ разложения (1) (размерности $k \times nT$);

¹Здесь и далее размерности массивов указаны в порядке, принятом в C и Python. В Fortran размерности массивов следует развернуть в обратном порядке

- массив `x_vect` левых сингулярных векторов XV (размерности $k \times nX$);
- массив `y_vect` правых сингулярных векторов YV (размерности $k \times nY$);
- массив `corrcoeff`, содержащий k коэффициентов корреляции между $XC_k(t)$ и $YC_k(t)$;
- массив `x_variance_fraction` (`y_variance_fraction`), содержащий доли дисперсии, приходящиеся на каждый из k левых (правых) сингулярных векторов;
- массив `eigenvalue_fraction`, содержащий долю дисперсии матрицы ковариации, приходящуюся на k -ую пару сингулярных векторов;
- массив `eigenvalues` сингулярных значений матрицы ковариации C .

2 Использование

Функция `supersvd` может вызываться как и из другой Python-функции, принимая на вход массивы данных, так и из командной строки, принимая на вход бинарные файлы (`.STD`). Последняя возможность реализована в функции `main`.

Функция `main` принимает на вход 3 обязательных аргумента:

- x имя файла, содержащего первое из полей (например, `X.STD`);
- y имя файла, содержащего второе из полей (например, `Y.STD`)²;

`-t, --time` длину временного интервала (например, в случае среднемесячных данных исторического эксперимента с INMCM это 165 лет).

Также функция `main` принимает 7 необязательных (опциональных) параметров:

- `--type` тип используемых данных — `real` (4 байта) или `double` (8 байт), значение по умолчанию — `real`;
- `-k` количество вычисляемых пар максимально скоррелированных мод, значение по умолчанию — 3;
- `-xv` имя файла, в который запишется массив `x_vect`;
- `-yv` имя файла, в который запишется массив `y_vect`;
- `-xc` имя файла, в который запишется массив `x_coeff`;
- `-yc` имя файла, в который запишется массив `y_coeff`;
- `-stat` имя файла (предпочтительно в формате `.CSV`), в который для каждого k запишутся домноженные на 100% элементы массивов: `corrcoeff`, `x_variance_fraction`, `y_variance_fraction`, `eigenvalue_fraction`.

Функция `main` также может быть запущена с ключом `--dont-subtract-mean`: при этом из полей X , Y не будут вычитаться их средние по времени значения.

Итак, чтобы вычислить с помощью функции `main` 4 максимально скоррелированные моды аномалий температуры и давления (типа `float`) и сохранить все возможные результаты, достаточно в командной строке выполнить:

```
python3 main.py -x ts.std -y ps.std -t 1147 -k 4 -xv tsv.std -yv psv.std
                -xc tsc.std -yc psc.std -stat c.csv
```

Информацию, сохраняемую в файл `c.csv`, функция `main` также выводит на экран.

²Если нужно посчитать ЭОФы, то в качестве первого и второго нужно задать одно и то же поле, то есть передать два раза имя одного файла.