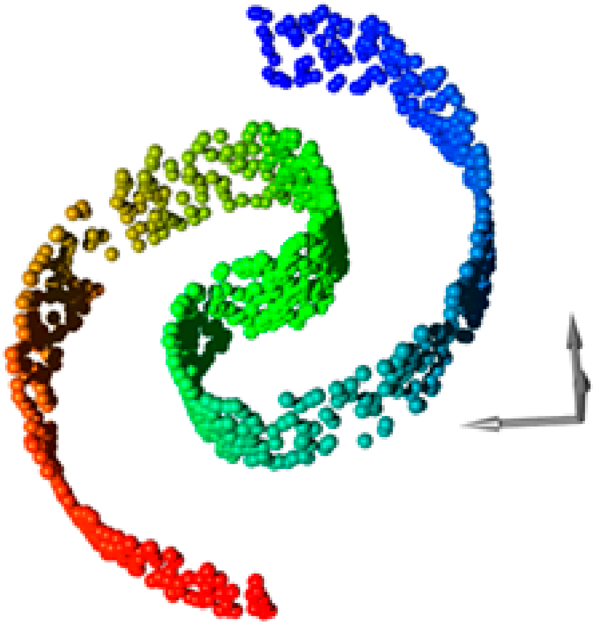

PYSOMAP

web.vscht.cz/spiwokv/pysomap
July 2007

Pysomap is python library for application of isometric feature mapping (Isomap) algorithm [Tenenbaum, de Silva, Langford (2000) Science 290, 2319.] using python. It does not have any graphical user interface.

DOWNLOAD

`web.vscht.cz/spiwokv/pysomap/pysomap-July2007.tar.gz` (0.2 MB).

INSTALATION

For Pysomap you need python (version 2.4.3 tested) and numpy (version 1.0.1 tested). Additional implementation of Floyd's algorithm is coded in C using SWIG (version 1.3.29). To compile this code on your platform you need SWIG and GNU C compiler. Other compilers were not tested.

Uncompress the file:

```
[unix]$ tar xzf pysomap-July2007.tar.gz
```

If you are using Red Hat or Fedora Core Linux you can use pre-compiled library for Floyd's algorithm. In some Linux installations (with SELinux) you can get an error message like: "*cannot restore segment prot after reloc: Permission denied*" when importing this library. This can be fixed by typing:

```
[unix]$ chcon -t textrel_shlib_t _floyd.so
```

You can also compile this library by typing:

```
[unix]$ ./build_floyd.sh
```

Similarly to using the pre-compiled library, it might be necessary to type:

```
[unix]$ chcon -t textrel_shlib_t _floyd.so
```

You can store isomap files in your working directory or you can add this directory to PYTHONPATH.

USING

Example:

```
#!/bin/env python

form pysomap import *

data = open("data", "r").readlines()
smatrix = []
for line in data:
  sline = str.split(line)
  aline=[]
  for item in sline:
    aline.append(float(item))
  smatrix.append(aline)
M = numpy.array(smatrix)

A = isodata()                                 # creates a new object A
A.load_isodata(M)                             # loads python array X (N x M) into object A
A.reduce_isodata(isomap_type="e", e=0.5, O=2) # performs dimensionality reduction of A
                                              #   isomap_type is "e" or "K" for ε-
                                              #      or K-isomap, respectively
                                              #   epsilon or K value must be set
                                              #   O is output dimensionality

results = open("results", "w")
for line in A.outdata():
  for item in line:
    results.write("  %f" % item)
  results.write("\n")
results.close()
```

INSTANCES OF THE CLASS `isodata` :

Methods:

`I.reduce_isodata(isomap_type, K, e, O, verbose)`	dimensionally reduces input data
`I.reduce_isodata2(isomap_type, K, e, O, verbose)`	dimensionally reduces distance matrix
`I.load_isodata(indata)`	loads input data
`I.distance_isodata()`	calculates distance matrix from input data
`I.graph_isodata()`	calculates graph matrix from distance matrix
`I.path_isodata()`	calculates path matrix from graph matrix
`I.mds_isodata()`	calculates output data from path matrix

Matrices:

`I.indata`	input data [N, M] (numpy array)
`I.dismat`	distance matrix [N, N] (numpy array)
`I.graph`	graph matrix [N, N] (numpy array)
`I.outdata`	output matrix [N, O] (numpy array)

Integers:

`I.N`	number of measurements
`I.M`	input dimensionality
`I.O`	output dimensionality
`I.K`	K for K-isomap

Floats:

`I.e`	epsilon for ε-isomap

Others:

`I.isomap_type`	”K” for K-isomap, ”e” for ε-isomap
`I.verbose`	verbose output if equal to ”v”

ACKNOVLEDGMENT

Thanks to Doc. Jiří Demel for helpful discussion on graph theory.