scipy.interpolate 패키지 - 수치 분석 라이브러리.

1) 1, 2차원 외삽

from scipy import interpolate

interp1d(x, y, z, kind='linear', copy=True, bounds_error=False, fill_value=None)

interp2d(x, y, z, kind='linear', copy=True, bounds_error=False, fill_value='extrapolate')

x, y : 데이터의 좌표(1차원 배열)

z : 데이터(2차원 배열)

kind : 보간방법(linear - 선형 보간, cubiq - 3차 보간, quintic - 5차 보간)

copy : 데이터를 복사

bounds_error : Ture의 경우, 보간시의 입력 좌표가 x, y의 범위를 넘으면 에러를 반환하며, False의 경우 범위와의 같은 fill_value에서 설정한 값을 반환한다.

fill_value : 외삽영역(extrapolate)에 대해서 삽입할 값. None의 경우 최근 방법에 의한 값을 반환한다.

관련 정보 :

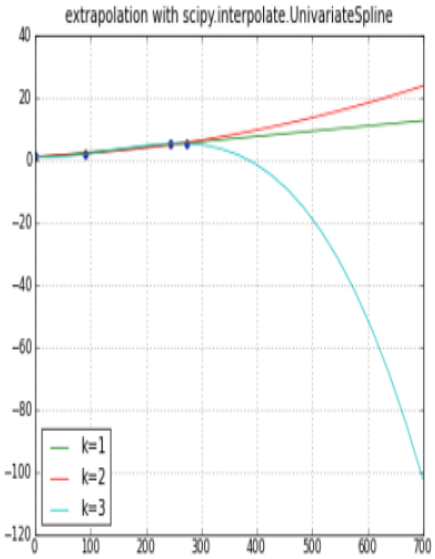
https://engineer-mole.tistory.com/313

https://stackoverflow.com/questions/2745329/how-to-make-scipy-interpolate-give-an-extrapolated-result-beyond-the-input-range

from scipy.interpolate import UnivariateSpline

scipy.interpolate.UnivariateSpine , scipy.interpolate.InterpolatedUnivariateSpline

```
""" extrapolate y,m,d data with scipy UnivariateSpline """
import numpy as np
from scipy.interpolate import UnivariateSpline
        # pydoc scipy.interpolate.UnivariateSpline -- fitpack, unclear
from datetime import date
from pylab import *     # ipython -pylab

__version__ = "denis 23oct"

def daynumber( y,m,d ):
        """ 2005,1,1 -> 0    2006,1,1 -> 365 ... """
        return date( y,m,d ).toordinal() -
date( 2005,1,1 ).toordinal()

days, values = np.array([
        (daynumber(2005,1,1), 1.2 ),
        (daynumber(2005,4,1), 1.8 ),
        (daynumber(2005,9,1), 5.3 ),
        (daynumber(2005,10,1), 5.3 )
        ]).T
dayswanted = np.array([ daynumber( year, month, 1 )
                for year in range( 2005, 2006+1 )
                for month in range( 1, 12+1 )])

np.set_printoptions( 1 )     # .1f
print "days:", days
print "values:", values
print "dayswanted:", dayswanted

title( "extrapolation with scipy.interpolate.UnivariateSpline" )
plot( days, values, "o" )
for k in (1,2,3):     # line parabola cubicspline
        extrapolator = UnivariateSpline( days, values, k=k )
        y = extrapolator( dayswanted )
        label = "k=%d" % k
        print label, y
        plot( dayswanted, y, label=label     )     # pylab

legend( loc="lower left" )
grid(True)
savefig( "extrapolate-UnivariateSpline.png", dpi=50 )
show()
```



관련 정보 :

https://stackoverflow.com/questions/1599754/is-there-easy-way-in-python-to-extrapolate-data-points-to-the-future?noredirect=1&lq=1

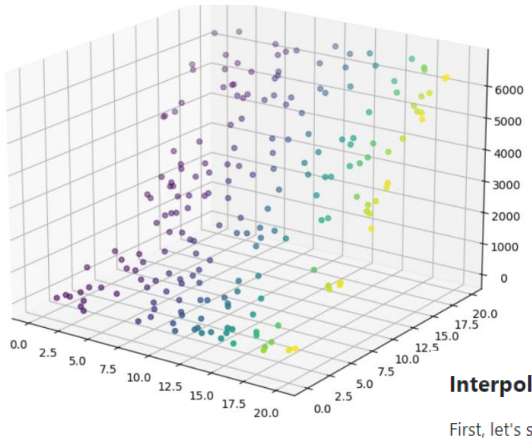
https://stackoverflow.com/questions/29862139/how-to-extrapolate-curves-in-python?noredirect=1&lq=1

https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.UnivariateSpline.html

https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.InterpolatedUnivariateSpline.html

3, 4 차원 외삽

from scipy,interpolate import Rbf




```
x, y, z, w = get_data(N=200)
plot_3d(x, y, z, w)
```

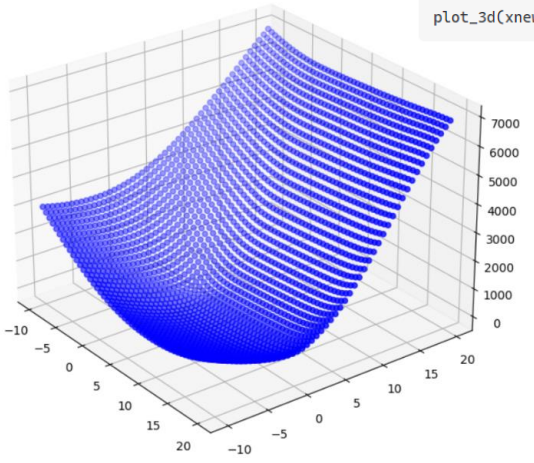
Interpolation & extrapolation in 3d

First, let's setup new x and y coordinates. To make this more interesting, let's extrapolate to minus x and minus y direction. This forms the new x and y range of interest.

```
xs = np.linspace(-10, 20) # some extrapolation to negative numbers
ys = np.linspace(-10, 20) # some extrapolation to negative numbers
xnew, ynew = np.meshgrid(xs, ys)
xnew = xnew.flatten()
ynew = ynew.flatten()
```

Interpolation with scipy.interpolate.Rbf. Now,

```
from scipy.interpolate import Rbf
rbf3 = Rbf(x, y, z, function="multiquadric", smooth=5)
znew = rbf3(xnew, ynew)

plot_3d(xnew, ynew, znew)
```



https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.Rbf.html

https://stackoverflow.com/questions/11214118/3d-extrapolation-in-python-basically-scipy-griddata-extended-to-extrapolate