

Intro

Gaussian Function은 흔히 정규분포함수의 기본 형태로 잘 알려져 있으며, 데이터 포인트들의 패턴을 이러한 함수에 근사시키는 **Gaussian Fitting**은 과학기술 분야에서 자주 하게 되는 작업이기도 합니다. IDL에서도 이러한 기능을 담당하는 내장함수들이 지원되는데, **1차원** 데이터의 경우는 **GAUSSFIT** 함수 그리고 **2차원** 데이터의 경우는 **GAUSS2DFIT** 함수가 그 역할을 담당합니다. 오늘은 이 두 함수를 이용하여 **1차원 및 2차원 데이터를 Gaussian 함수로 근사하는 방법**을 소개하고자 합니다.

1차원 예제 데이터의 생성

예제 데이터는 Gaussian 함수의 형태와 어느 정도 유사한 모습을 띄도록 하기 위하여, 다음과 같이 Gaussian 함수의 기본 형태를 참조하면서 난수값들을 가미하여 생성하였습니다.

```
n = 101
x = (FINDGEN(n) - (n/2)) / 4
a = [4.0, 1.0, 2.0, 1.0, 0.25, 0.01]
z = (x - a[1]) / a[2]
y = 0.4 * RANDOMN(-5, n) + $
    a[0] * EXP(-z^2/2)
```

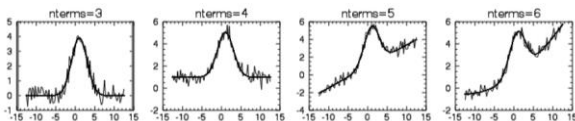
여기서 생성된 x 배열은 101개의 값들로 구성되며 대략 -15~+15의 범위 내에 존재합니다. 그리고 y 배열을 생성하는데 있어서는 Gaussian 함수의 수식을 이용하기 위하여 관련 계수값들로 구성된 배열 a를 생성하여 활용하였습니다. 물론 함수의 패턴과 똑같으면 안되기 때문에, RANDOMN 함수를 함께 이용하여 난수값들을 더해줌으로써 어느 정도 실제 데이터와 유사한 느낌이 되도록 조정하였습니다.

GAUSSFIT 함수의 개요

Gaussian 함수는 기본적으로는 다음과 같은 형태를 가지는데, 이 경우에는 3개의 계수값들이 존재하게 됩니다.

$$f(x) = a \cdot \exp\left(-\frac{(x-b)^2}{2c^2}\right)$$

그러나 베이스라인(Baseline) 자체가 직선 또는 곡선의 형태를 갖는 경우까지 고려하면 계수값들의 갯수가 6개까지도 가능합니다. IDL의 **GAUSSFIT 함수**에서는 이와 같은 **3~6개 계수값들이 존재하는 모든 경우들에 대하여 근사 결과를 얻는 것이 가능합니다.**



GAUSSFIT 함수의 사용 및 결과 산출

이제 앞서 생성했던 예제 데이터 x, y에 대하여 GAUSSFIT 함수를 적용하여 근사 결과를 얻어봅시다. 계수값이 3개인 경우로 가정하여 근사 결과를 얻는 방법은 다음과 같습니다.

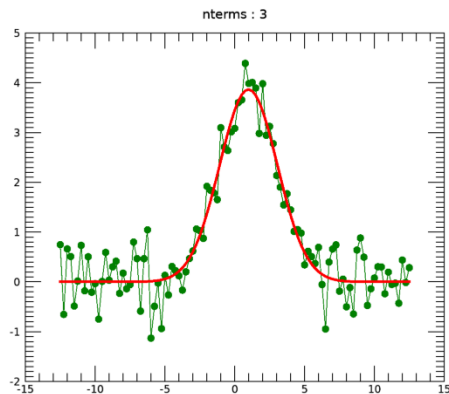
```
yfit = GAUSSFIT(x, y, coeff, NTERMS=3)
PRINT, 'Result : ', coeff
```

여기서는 NTERMS 키워드를 사용하면서 그 값을 3으로 설정해야 한다는 점에 유의해야 합니다. 그리고 계산 결과는 세번째 인수인 coeff라는 항목 그리고 좌변의 yfit이라는 항목을 통하여 얻게 됩니다. 먼저 coeff는 계수값 3개로 구성된 배열이 됩니다. 실제로 출력된 값들은 다음과 같습니다.

```
Result : 3.86547 0.987056 2.06947
```

그리고 yfit은 근사된 이론적 수식에 근거하여 산출된 Y값들로 구성된 배열로서, 원본 데이터와 동일하게 101개의 값들로 구성됩니다. 따라서 이 배열을 표출에도 활용할 수 있습니다. 원래 데이터 및 근사된 결과를 함께 표출하는 과정 및 그 결과 그림은 다음과 같습니다.

```
win = WINDOW(DIMENSIONS=[600, 500], $
    /NO_TOOLBAR)
p1 = PLOT(x, y, SYMBOL='circle', $
    /SYM_FILLED, COLOR='green', $
    TITLE='nterms : 3', MARGIN=0.1, $
    /CURRENT)
p2 = PLOT(x, yfit, COLOR='red', $
    THICK=3, /OVERPLOT)
```



이 그림을 보면 근사의 결과가 원래의 데이터를 어느 정도 잘 반영하는 것으로 보입니다. 이렇게 계수가 3개인 경우 외에도 4, 5, 6개인 경우에 대한 예제들 및 더 자세한 내용은 아래 링크를 통하여 보실 수 있습니다.

<http://blog.daum.net/swrush/572>

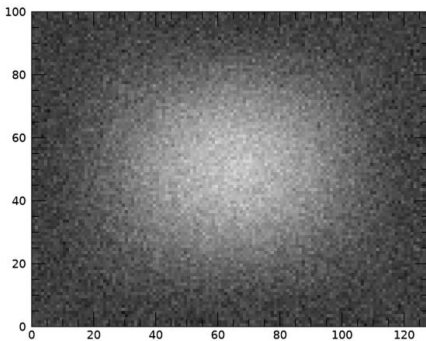
2차원 예제 데이터의 생성

2차원 Gaussian Fitting을 위한 예제 데이터는 다음과 같이 HANNING 및 RANDOMN 함수를 활용하여 128x100의 구조를 갖는 2차원 배열로 생성합니다.

```
nx = 128
ny = 100
z = HANNING(nx, ny)*10+5+ $
    RANDOMN(-1, nx, ny)
HELP, z
PRINT, MIN(z), MAX(z)
```

여기서 출력된 결과를 보면, 생성된 2차원 배열 z의 값들은 대략 1.4~16.9의 범위 내에 있게 됩니다. 일단 배열 z의 모습을 이미지의 형태로 표출해봅시다. 그 과정 및 표출 결과는 다음과 같습니다.

```
win1 = WINDOW(DIMENSION=[nx, ny]*5, $
    /NO_TOOLBAR)
im1 = IMAGE(z, AXIS_STYLE=2, $
    MIN_VALUE=0, MAX_VALUE=20, $
    MARGIN=0.1, FONT_SIZE=11, /CURRENT)
```



GAUSS2DFIT 함수의 사용 및 결과 산출

GAUSS2DFIT 함수의 바탕이 되는 수식은 기본적으로는 **2차원 타원형 Gaussian 함수의 형태**입니다. 이 수식에 관한 세부 사항은 IDL 도움말에서 GAUSS2DFIT 함수에 관한 섹션에 자세히 소개되어 있습니다. 이제 앞서 생성했던 예제 데이터 z에 대하여 GAUSS2DFIT 함수를 다음과 같이 적용하여 근사 결과를 얻어봅시다.

```
zfit = GAUSS2DFIT(z, a, /TILT)
PRINT, 'Result : ', STRING(a, $
    FORMAT='(7f8.3)')
```

여기서 계산 결과는 두번째 인수인 a라는 항목 그리고 좌변의 zfit이라는 항목을 통하여 얻게 됩니다. 먼저 **a는 2차원 Gaussian 수식을 구성하는 7개의 계수값들로 구성된 배열**이 됩니다. 이 값들의 개요는 다음과 같습니다.

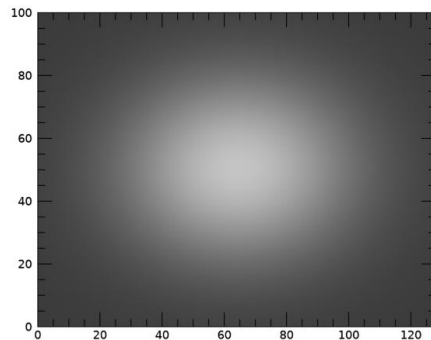
```
A[0] : constant term
A[1] : scale factor
A[2] : Gaussian width (X direction)
A[3] : Gaussian width (Y direction)
A[4] : center X location
A[5] : center Y location.
A[6] : rotation angle of the
    ellipse from the X axis in radians,
    counter-clockwise.
```

그리고 실제로 출력된 값들은 다음과 같습니다.

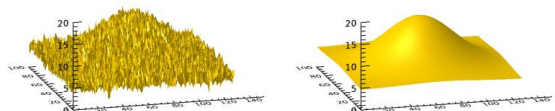
```
Result :      4.615  10.761  26.929
          21.115  63.858  49.947  -0.009
```

또 다른 결과인 zfit는 그 배열 구조는 원본 데이터인 z와 동일하지만 내부적으로는 근사 결과값들로 구성된 2차원 배열입니다. 즉 a로 얻은 7개의 계수값들을 Gaussian 수식에 투입하여 계산된 **근사 결과 2차원 배열이 zfit**이라고 보면 됩니다. 별도의 그래픽창을 띄워서 이 결과를 이미지의 형태로 표출해봅시다. 그 과정 및 표출 결과는 다음과 같습니다.

```
win2 = WINDOW(DIMENSION=[nx, ny]*5, $
    /NO_TOOLBAR)
im2 = IMAGE(zfit, AXIS_STYLE=2, $
    MIN_VALUE=0, MAX_VALUE=20, $
    MARGIN=0.1, FONT_SIZE=11, /CURRENT)
```



이와 같이 이미지 형태로 표출된 두 결과를 비교해보면, 근사의 결과가 원본 데이터와 충분히 유사한 모습을 보이는 것 같습니다. 이러한 유사성을 좀 더 잘 나타내기 위하여, SURFACE 함수를 이용하여 비교 표출해본 모습은 다음과 같습니다.



GAUSS2DFIT 함수의 사용에 관한 더 자세한 내용은 아래 링크를 통하여 보실 수 있습니다.

<http://blog.daum.net/swrush/573>