



< Previous



Next >

Problem 1

🔖 Bookmark this page

Problem 1: Curve Fitting

15.0/15.0 points (graded)

Implement the `generate_models` function.

- `x` and `y` are two lists corresponding to the x-coordinates and y-coordinates of the data samples (or data points); for example, if you have N data points, `x = [x1 , x2 , ..., xN]` and `y = [y1 , y2 , ..., yN]`, where `x_i` and `y_i` are the x and y coordinate of the i-th data points. In this problem set, each x coordinate is an integer and corresponds to the year of a sample (e.g., 1997); each corresponding y coordinate is a float and represents the temperature observation (will be computed in multiple ways) of that year in Celsius. This representation will be used throughout the entire problem set.
- `degs` is a list of integers indicating the degree of each regression model that we want to create. For each model, this function should fit the data (x,y) to a polynomial curve of that degree.
- This function should return a list of models. A model is the numpy 1d array of the coefficients of the fitting polynomial curve. Each returned model should be in the same order as their corresponding integer in `degs`.

Example:

```
print(generate_models([1961, 1962, 1963],[4.4,5.5,6.6],[1, 2]))
```

Should print something close to:

```
[array([ 1.10000000e+00, -2.15270000e+03]), array([ -8.86320195e-14, 1.10000000e+00, -2.15270000e+03])]
```

The above example was generating a linear and a quadratic curve on data samples $(x_i, y_i) = (1961, 4.4), (1962, 5.5),$ and $(1963, 6.6)$. The resulting models are in the same order as specified in `degs`. Note that it is fine you did not get the exact number because of numerical errors.

Note: If you want to use numpy arrays, you should add the following lines at the beginning of your code for the grader:

```
import os
os.environ["OPENBLAS_NUM_THREADS"] = "1"
```

Then, do `import numpy as np` and use `np.METHOD_NAME` in your code. Unfortunately, `pylab` does not work with the grader.

```
1 import numpy as np
2
3 def generate_models(x, y, degs):
4     """
5     Generate regression models by fitting a polynomial for each degree in degs
6     to points (x, y).
7     Args:
8         x: a list with length N, representing the x-coords of N sample points
9         y: a list with length N, representing the y-coords of N sample points
10        degs: a list of degrees of the fitting polynomial
11    Returns:
12        a list of numpy arrays, where each array is a 1-d array of coefficients
13        that minimizes the squared error of the fitting polynomial
14    """
15    # TODO
```

Press ESC then TAB or click outside of the code editor to exit

Correct

Test results

CORRECT

[See full output](#)

[See full output](#)

 Calculator

Submit

< Previous

Next >

Problem 1: Curve Fitting

Topic: Problem Set 4 / Problem 1: Curve Fitting

Hide Discussion

© All Rights Reserved



edX

[About](#)

[Affiliates](#)

[edX for Business](#)

[Open edX](#)

[Careers](#)

[News](#)

Legal

[Terms of Service & Honor Code](#)

[Privacy Policy](#)

[Accessibility Policy](#)

[Trademark Policy](#)

[Sitemap](#)

Connect

[Blog](#)

[Contact Us](#)

[Help Center](#)

[Security](#)

[Media Kit](#)



© 2022 edX LLC. All rights reserved.

深圳市恒宇博科技有限公司 [粤ICP备17044299号-2](#)

Calculator