COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS
DEPARTMENT OF APPLIED INFORMATICS

Introduction to Artificial Intelligence
Exercise 8 - Regression

November 9, 2022

8. LINEAR AND QUADRATIC REGRESSION

The task is to program functions that compute linear and quadratic regression from a given set of data.

Regression - formulas:

Derivation of formulas for regression (by minimalization of regression error) is quite complicated so we provide them already derived - you don't have to modify them, just implement them in your code as they are written here:

Linear regression: For n fitted points the result is a line $y = ax + b$ - with parameters a, b calculated as:

$$a = \frac{n\sum(x_i y_i) - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2}$$

$$b = \frac{\sum y_i - a\sum x_i}{n}$$

Quadratic regression: For n fitted points the result is a parabola $y = ax^2 + bx + c$ - with parameters a, b, c calculated as:

$$a = \frac{S(x^2y)S(xx) - S(xy)S(xx^2)}{S(xx)S(x^2x^2) - (S(xx^2))^2}$$

$$b = \frac{S(xy)S(x^2x^2) - S(x^2y)S(xx^2)}{S(xx)S(x^2x^2) - (S(xx^2))^2}$$

$$c = \frac{\sum y_i - b\sum x_i - a\sum x_i^2}{n}$$

Where:

$$S(xx) = \sum x_i^2 - \frac{1}{n}(\sum x_i)^2$$

$$S(xy) = \sum (x_i y_i) - \frac{1}{n}\sum x_i \sum y_i$$

$$S(xx^2) = \sum x_i^3 - \frac{1}{n}\sum x_i \sum x_i^2$$

$$S(x^2y) = \sum(x_i^2 y_i) - \frac{1}{n}\sum x_i^2 \sum y_i$$
$$S(x^2x^2) = \sum x_i^4 - \frac{1}{n}(\sum x_i^2)^2$$

Program:

When you run the program *regression.py*, two types of data are generated - points of a random line + noise and points of a random parabola + noise. These data are sent as arguments to functions *linear_regression* and *quadratic_regression*, which you have to implement. Data is represented as a list of tuples (x, y). Then the data (along with your calculated regression) is plotted in a graph.
The graph also contains computed regression error - Mean Squared Error (MSE), sum of squares of differences between actual data points and their predicted values:

$MSE = \sum_{i=0}^{n-1}(y_i - \tilde{y}_i)^2$, kde $\tilde{y}_i = ax_i + b$ in linear and $\tilde{y}_i = ax_i^2 + bx_i + c$ in quadratic regression.

There is also plotting of testing data included - these are from the same line or parabola, but for x values from different interval than the fitted data. The regression error is also computed for the testing data.
If your regression is implemented correctly, the error of quadratic regression on fitted data should never be higher than the error of linear regression.

Assignment 1 (0.5p): Implement the formula for calculating linear regression in function *linear_regression* in *regression.py*.
Assignment 2 (0.5p): Implement the formula for calculating quadratic regression in function *quadratic_regression* in *regression.py*.

You can use *numpy* to simplify some calculations (sums, matrix operations etc.) but **implement the formulas by yourself** - the use of existing functions that calculate regression (such as *np.polyfit()*) is forbidden. However it is recommended to use it while testing/debugging to check if your solution is correct.