Project 1: K-Nearest-Neighbor Regression

CS 5342

Points: 100; Due: 11:59 pm, Feb 7

Consider a single dimension (variable) X. Obtain N = 100 *iid* samples $x_1, x_2, \cdots$ of X uniformly randomly between 1 and 10, and then obtain the corresponding y values as the natural logarithm of x plus a Gaussian noise (mean 0, standard deviation 0.1), with different points having different amounts of noise. Now use K-NN regression to obtain $\hat{y}$ values (= estimates of y) at x-values of 1, 3, 5, 7 and 9 for each of the following three schemes:

- the K neighbors contribute equally (separately for K = 1, 3, 50)
- each of the K neighbors has an influence that is inversely proportional to the distance from the point (separately for K = 1, 3, 50)
- all the N points contribute, with each contribution proportional to $e^{-\frac{1}{2}d^2}$, where d represents distance.

Print the numerical values of the $(x, \hat{y})$ pairs for each of the above cases (there should be a total of 3 + 3 +1 = 7 cases and 5 $(x, \hat{y})$ pairs for each case).

Also, plot the (x', y') and $(x, \hat{y})$ points for each of these seven cases, where x' is the point (out of the 100 sample points) closest to x and y' is the y-value of x' (there should be a total of 7 plots for this, each plot showing the (x', y') and $(x, \hat{y})$ points). It is possible but unlikely that x and x' coincide.

Use of Python as the implementation laguage is preferred (but no penalties for using Java/C++/C). R is not encouraged; if you do not know any language/package other than R, please email me ASAP. While I recommend that you write all (most) of the code from scratch without using off-the-shelf packages (we learn best when we write code to implement algorithms from scratch), you may use packages, including the ones where K-NN regression is available as a ready-to-use function. E.g., you may use numpy, scipy, sklearn (sklearn.neighbors.KNeighborsRegressor may come in handy), matplotlib, and seaborn. There will be no penalty for using packages.

Set the seed at the beginning of your program so that your results are reproducible.

This is a group project, with up to four students per group. There will be only one submission from a group (it doesn't matter which member submits it; the submissions of the other members will remain blank on Canvas.) Please write the names of the group members at the top of the very first page of the submission. Form your own groups by interacting amongst yourselves but please do NOT use Canvas's features to store group compositions. A student may choose to be in different groups for different projects. Working in groups is highly recommended but not mandatory; a student may choose to work independently.

Please submit on Canvas a single pdf file (no other file type, please) containing the source code and all output. If you are converting ipynb files into pdf, please be careful that end-of-line text is not chopped off (double-check the wrap-around, if any).