CSE 6363 - Machine Learning

Fall 2022

Due Date: Nov 6, 2022, 11:59 PM

Data Set

Use the dataset given at the bottom of this file.

Do Not Use

You are not allowed to use any ML libraries other than NumPy.
You cannot use sklearn or any ML library. If used, you will receive a penalty of 90 points.
You cannot use pandas. If used, you will receive a penalty of 20 points.

Libraries

You are allowed to use NumPy, math.
You can use matplotlib to plot graphs.
If you want to use any other library apart from these, please check with your GTA and get their approval.

Where to code

1. We will provide you with a directory structure with python files for each part of every question. You must write your code in these files.
2. It will contain a script to execute the files. You must run this script and verify that your code runs before you submit. To run this script you must make it executable first or else you will get permission denied error.

Linear Regression:

1. Consider a simplified fitting problem in the frequency domain where we are looking to find the best fit of data with a set of periodic (trigonometric) basis functions of the form 1, $\sin^2(x)$, $\sin^2(k * x)$, $\sin^2(2 * k * x)$,..., where k is effectively the frequency increment. The resulting function for a given ”frequency increment”, k, and ”function depth”, d, and parameter vector Θ is then:

$$y = \Theta_0 * 1 + \sum_{i=1}^{d} (\Theta_i * \sin(i * k * x) * \sin(i * k * x)$$

Try ”frequency increment” **k** from 1-10

For example, if k = 1 and d = 1, your basis (feature) functions are: 1, $\sin^2(x)$

if k = 1 and d = 2, your basis (feature) functions are: 1, $\sin^2(x)$, $\sin^2(2.x)$

if k = 3 and d = 4, your basis (feature) functions are: 1, $\sin^2(3*1*x)$, $\sin^2(3*2*x)$, $\sin^2(3*3*x)$, $\sin^2(3*4*x)$

This means that this problem can be solved using linear regression as the function is linear in terms of the parameters Θ.

Try ”frequency increment” **k** from 1-10 and thus your basis functions as part of the data generation process described above.

a) Implement a linear regression learner to solve this best fit problem for 1 dimensional data. Make sure your implementation can handle fits for different ”function depths” (at least to ”depth” 6).
b) Apply your regression learner to the data set that was generated for Question 1b) and plot the resulting function for “function depth” 0, 1, 2, 3, 4, 5, and 6. Plot the resulting function together with the data points
c) Evaluate your regression functions by computing the error on the test data points that were generated for Question 1c). Compare the error results and try to determine for what “function depths” overfitting might be a problem. Which ”function depth” would you consider the best prediction function and why? For which values of k and d do you get minimum error?

d) Repeat the experiment and evaluation of part b) and c) using only the first 20 elements of the training data set part b) and the Test set of part c). What differences do you see and why might they occur?

Locally Weighted Linear Regression

2. Another way to address nonlinear functions with a lower likelihood of overfitting is the use of locally weighted linear regression where the neighborhood function addresses non-linearity and the feature vector stays simple. In this case we assume that we will use only the raw feature, x, as well as the bias (i.e. a constant feature 1). Thus the locally applied regression function is $y = \Theta_0 + \Theta_1 * x$

As discussed in class, locally weighted linear regression solves a linear regression problem for each query point, deriving a local approximation for the shape of the function at that point (as well as for its value). To achieve this, it uses a modified error function that applies a weight to each data point's error that is related to its distance from the query point. Here we will assume that the weight function for the ith data point and query point x is:

$$w^{(i)}(x) = e^{-\frac{(x^{(i)}-x)^2}{2\gamma^2}}$$

Use γ : 0.204

where γ is a measure of the "locality" of the weight function, indicating how fast the influence of a data point changes with its distance from the query point.

a. Implement a locally weighted linear regression learner to solve the best fit problem for 1 dimensional data.

b. Apply your locally weighted linear regression learner to the data set that was generated for Question 1b) and plot the resulting function together with the data points

c. Evaluate the locally weighted linear regression on the Test data from Question 1 c). How does the performance compare to the one for the results from Question 1 c) ?

d. Repeat the experiment and evaluation of part b) and c) using only the first 20 elements of the training data set. How does the performance compare to the one for the results from Question 1 d) ? Why might this be the case?

e. Given the results form parts c) and d), do you believe the data set you used was actually derived from a function that is consistent with the function format in Question 1 ? Justify your answer.

Logistic Regression

3. Consider again the problem from Questions 1 and 2 in the first assignment where we want to predict the gender of a person from a set of input parameters, namely height, weight, and age. Assume the same datasets you generated for the first assignment. Use learning rate = 0.01. Try different values for number of iterations.

a. Implement logistic regression to classify this data (use the individual data elements, i.e. height, weight, and age, as features). Your implementation should take different data sets as input for learning.

b. Plot the resulting separating surface together with the data. To do this plotting you need to project the data and function into one or more 2D space. The best visual results will be if projection is done along the separating hyperplane (i.e. into a space described by the normal of the hyperplane and one of the dimension within the hyperplane)

c. Evaluate the performance of your logistic regression classifier in the same way as for Project 1 using leave-one-out validation and compare the results with the ones for KNN and Naïve Bayes Discuss what differences exist and why one method might outperform the others for this problem.

d. Repeat the evaluation and comparison from part c) with the age feature removed. Again, discuss what differences exist and why one method might outperform the others in this case.

Data for Q1 and Q2

```
(( 0.5335 ), -2.4546402277793 )
(( 1.2765 ), -2.1009775221007 )
(( -1.425 ), -2.2346352508653 )
(( -2.031 ), -3.7262599354679 )
(( 0.982 ), -2.4267836125928 )
(( 1.949 ), -3.7226247858091 )
(( 0.727 ), -3.6958087651294 )
(( 1.7455 ), -1.9104672644596 )
(( 2.295 ), -1.7294822135928 )
(( 2.898 ), -3.6866360685813 )
(( 1.827 ), -1.7944419384599 )
(( 2.7015 ), -2.0430807348365 )
(( 2.1695 ), -2.1517891235202 )
(( -0.949 ), -1.6414658494919 )
(( 2.273 ), -1.929691155958 )
(( -1.2255 ), -3.5116333758746 )
(( 0.952 ), -1.8526518540251 )
(( 0.1625 ), -3.749177716962 )
(( 2.0925 ), -3.7403451861929 )
(( 2.232 ), -1.9618143533032 )
(( -0.6955 ), -3.516162185567 )
(( -2.9505 ), -3.7458335251531 )
(( -0.056 ), -1.6879498974046 )
(( 2.5645 ), -3.498931727753 )
(( 1.6915 ), -2.9329282884097 )
(( -2.069 ), -3.6792800420669 )
(( -0.9105 ), -2.1721208206997 )
(( -1.9965 ), -3.7350040474573 )
(( -1.8085 ), -2.0823871367996 )
(( -0.136 ), -3.5875390464557 )
(( -0.0225 ), -1.8622787522427 )
(( 2.893 ), -3.6507965335012 )
(( 1.3415 ), -1.9562954362255 )
(( -1.5515 ), -3.7017955778347 )
(( -0.2395 ), -3.6130434462801 )
(( 2.005 ), -3.6124544076089 )
(( -1.9 ), -2.9872314745436 )
(( 1.3805 ), -1.8804329525486 )
(( -1.6385 ), -3.8845713564878 )
(( 2.056 ), -3.7295226627743 )
(( -2.9665 ), -3.7086655161127 )
(( 0.094 ), -2.6710251694207 )
(( -1.0495 ), -3.6340627209245 )
(( 0.517 ), -1.9948333119549 )
(( -1.392 ), -1.930592300233 )
(( -1.0755 ), -3.8552973746459 )
(( 2.132 ), -3.2991619687281 )
(( 1.648 ), -3.5808183669749 )
(( 0.3605 ), -2.5422144072754 )
(( 1.59 ), -3.6494091506307 )
(( 1.071 ), -3.6376796631403 )
(( 1.2555 ), -2.4753481185741 )
(( 0.3165 ), -3.5246543638716 )
(( 1.711 ), -2.4015153414432 )
(( 1.4945 ), -3.8200731846583 )
(( 1.546 ), -3.7597970567707 )
(( -2.9935 ), -3.711755682811 )
(( -1.4655 ), -3.2837747364638 )
(( -2.212 ), -1.6569687344991 )
(( -0.8075 ), -2.4979167035923 )
(( 2.093 ), -3.7092282762502 )
(( 2.819 ), -3.511543393151 )
(( -2.1505 ), -2.6426900703635 )
(( 1.6255 ), -3.514472394378 )
(( -1.8465 ), -1.7337800623299 )
(( -0.02 ), -1.7591860203851 )
(( 0.4295 ), -2.2216444545995 )
(( 0.634 ), -3.5268029297223 )
(( 0.1305 ), -3.5220981785883 )
(( 1.6915 ), -3.1782524679925 )
(( -0.0155 ), -1.9459953737455 )
(( 0.4195 ), -1.8738423830371 )
(( 2.6985 ), -2.066808159228 )
(( -2.2915 ), -1.7531612474651 )
(( 2.0145 ), -3.8940653293599 )
(( -0.534 ), -2.5110706419569 )
(( 1.0775 ), -3.599780748224 )
(( 0.602 ), -3.5891068713484 )
(( 2.4275 ), -3.6896986933457 )
(( -2.6105 ), -2.3639441682841 )
(( -0.5375 ), -2.4530008771212 )
(( -2.5885 ), -3.0974058047486 )
(( -1.391 ), -1.8099241956004 )
(( 2.784 ), -2.5385213919572 )
(( 2.9055 ), -3.6416656607979 )
(( -2.207 ), -1.8477185535112 )
(( -1.1055 ), -3.8295012445337 )
(( 0.156 ), -3.6943375997081 )
(( 2.569 ), -3.5514365524153 )
(( 1.9485 ), -3.7891389210125 )
(( 1.5565 ), -3.7079924247506 )
(( 2.9065 ), -3.8914303590626 )
(( 1.9715 ), -3.6477399946517 )
(( -1.039 ), -3.5896071021435 )
(( 0.972 ), -2.3620440196746 )
(( 0.449 ), -2.0676317671988 )
(( 2.921 ), -3.6213112603092 )
(( 1.9795 ), -3.713967017383 )
(( 2.5435 ), -3.6428211579169 )
(( 2.31 ), -1.9593337784247 )
(( 0.3385 ), -3.1786029561839 )
(( 2.2885 ), -1.8570201743695 )
(( -1.3755 ), -2.0973893475077 )
(( -2.979 ), -3.6924088990707 )
(( 1.457 ), -3.2937218417944 )
(( -2.7805 ), -2.6070922086127 )
(( 0.5135 ), -1.8300516168375 )
(( 0.235 ), -3.7288960006295 )
(( 0.8745 ), -2.0457925497604 )
(( -0.9525 ), -1.8451241183642 )
(( -1.507 ), -3.686643065362 )
(( 1.2785 ), -2.0454953179933 )
(( 2.977 ), -3.6838245392547 )
(( 0.876 ), -2.1588475296108 )
(( 2.8545 ), -3.5578161609965 )
(( 0.7045 ), -3.6743195933816 )
(( 1.736 ), -1.8855427641422 )
(( 2.391 ), -3.6078136420464 )
(( 0.0875 ), -2.4999293394878 )
(( 1.034 ), -3.5781581202501 )
(( 0.3935 ), -1.7988766920722 )
(( -0.1515 ), -3.7317198224703 )
(( 1.6825 ), -3.2703369022325 )
(( 2.666 ), -1.9303761653202 )
(( -0.0275 ), -1.8714670599721 )
(( -2.565 ), -3.5772951189505 )
(( 2.366 ), -3.2280770130413 )
(( -2.3105 ), -2.0433564215449 )
```

Question 1 c) Test Data:

```
(( -0.407 ), -1.8158962969599 )
(( -2.4365 ), -3.6636571672605 )
(( 0.5695 ), -3.3911442075951 )
(( -2.3455 ), -2.9020426856979 )
(( 0.785 ), -3.2099868850666 )
(( 0.6865 ), -3.6904139857969 )
(( 2.786 ), -2.6472709646476 )
(( 0.5205 ), -2.0553292845697 )
(( 0.3395 ), -3.1245179169378 )
(( 2.4495 ), -3.6822392184811 )
```

Question 2 weight scaling factor γ : 0.204

Data for Q3 (same as the data for Project 1)

```
(( 1.5963600450124, 75.717194178189, 23), W )
(( 1.6990610819676, 83.477307503684, 25), M )
(( 1.5052092436, 74.642420817737, 21), W )
(( 1.5738635789008, 78.562465284603, 30), M )
(( 1.796178772769, 74.566117057707, 29), M )
(( 1.6274618774347, 82.250591567161, 21), W )
(( 1.6396843250708, 71.37567170848, 20), W )
(( 1.538505823668, 77.418902097029, 32), W )
(( 1.6488692005889, 76.333044488477, 26), W )
(( 1.7233804613095, 85.812112126306, 27), M )
(( 1.7389100516771, 76.424421782215, 24), W )
(( 1.5775696242624, 77.201404139171, 29), W )
(( 1.7359417237856, 77.004988515324, 20), M )
(( 1.5510482441354, 72.950756316157, 24), W )
(( 1.5765653263667, 74.750113664457, 34), W )
(( 1.4916026885377, 65.880438515643, 28), W )
(( 1.6755053770068, 78.901754249459, 22), M )
(( 1.4805881225567, 69.652364469244, 30), W )
(( 1.6343943760912, 73.998278712613, 30), W )
(( 1.6338449829543, 79.216500811112, 27), W )
(( 1.5014451222259, 66.917339299419, 27), W )
(( 1.8575887178701, 79.942454850988, 28), M )
(( 1.6805940669394, 78.213519314007, 27), W )
(( 1.6888905106948, 83.031099742808, 20), M )
(( 1.7055120272359, 84.233282531303, 18), M )
(( 1.5681965896812, 74.753880204215, 22), W )
(( 1.6857758389206, 84.014217544019, 25), W )
(( 1.7767370337678, 75.709336556562, 27), M )
(( 1.6760125952287, 74.034126149139, 28), M )
(( 1.5999112612548, 72.040030344184, 27), M )
(( 1.6770845322305, 76.149431872551, 25), M )
(( 1.7596128136991, 87.366395298795, 29), M )
(( 1.5344541456027, 73.832214971449, 22), W )
(( 1.5992629534387, 82.4806916967, 34), W )
(( 1.6714162787917, 67.986534194515, 29), W )
(( 1.7070831676329, 78.269583353177, 25), M )
(( 1.5691295338456, 81.09431696972, 27), M )
(( 1.7767893419281, 76.910413184648, 30), M )
(( 1.5448153215763, 76.888087599642, 32), W )
(( 1.5452842691008, 69.761889289463, 30), W )
(( 1.6469991919639, 82.289126983444, 18), W )
(( 1.6353732734723, 77.829257585654, 19), W )
(( 1.7175342426502, 85.002276406574, 26), M )
(( 1.6163551692382, 77.247935733799, 21), M )
(( 1.6876845881843, 85.616829192322, 27), M )
(( 1.5472705508274, 64.474350365634, 23), W )
(( 1.558229415357, 80.382011318379, 21), W )
(( 1.6242189230632, 69.567339939973, 28), W )
(( 1.8215645865237, 78.163631826626, 22), W )
(( 1.6984142478298, 69.884030497097, 26), M )
(( 1.6468551415123, 82.666468220128, 29), M )
(( 1.5727791290292, 75.545348033094, 24), M )
(( 1.8086593470477, 78.093913654921, 27), M )
(( 1.613966988578, 76.083586505149, 23), W )
(( 1.6603990297076, 70.539053122611, 24), M )
(( 1.6737443242383, 66.042005829182, 28), W )
(( 1.6824912337281, 81.061984274536, 29), M )
(( 1.5301691510101, 77.26547501308, 22), M )
(( 1.7392340943261, 92.752488433153, 24), M )
(( 1.6427105169884, 83.322790265985, 30), M )
(( 1.5889040551166, 74.848224733663, 25), W )
(( 1.5051718284868, 80.078271153645, 31), W )
(( 1.729420786579, 81.936423109142, 26), M )
(( 1.7352568354092, 85.497712687992, 19), M )
(( 1.5056950011245, 73.726557750383, 24), W )
(( 1.772404089054, 75.534265951718, 30), M )
(( 1.5212346939173, 74.355845722315, 29), W )
(( 1.8184515409355, 85.705767969326, 25), M )
(( 1.7307897479464, 84.277029918205, 28), W )
(( 1.6372690389158, 72.289040612489, 27), M )
(( 1.6856953072545, 70.406532419182, 28), W )
(( 1.832494802635, 81.627925524191, 27), M )
(( 1.5061197864796, 85.886760677468, 31), W )
(( 1.5970906671458, 71.755566818152, 27), W )
(( 1.6780459059283, 78.900587239209, 25), W )
(( 1.6356901170146, 84.066566323977, 21), W )
(( 1.6085494116591, 70.950456539016, 30), M )
(( 1.5873479102442, 77.558144903338, 25), M )
(( 1.7542078120838, 75.3117550236, 26), M )
(( 1.642417315747, 67.97377818999, 31), W )
(( 1.5744266340913, 81.767568318602, 23), M )
(( 1.8470601407979, 68.606183538532, 30), W )
(( 1.7119387468283, 80.560922353487, 27), W )
(( 1.6169930563306, 75.538611935125, 27), M )
(( 1.6355653058986, 78.49626023408, 24), M )
(( 1.6035395957618, 79.226052358485, 33), M )
(( 1.662787957279, 76.865925681154, 25), M )
(( 1.5889291137091, 76.548543553914, 28), W )
(( 1.9058127964477, 82.56539915922, 25), M )
(( 1.694633493614, 62.870480634419, 21), W )
(( 1.7635692396034, 82.479783004684, 27), M )
(( 1.6645292231449, 75.838104636904, 29), W )
(( 1.7201968406129, 81.134689293557, 24), W )
(( 1.5775563651749, 65.920103519266, 24), W )
(( 1.6521294216004, 83.312640709417, 28), M )
(( 1.5597501915973, 76.475667826389, 30), W )
(( 1.7847561120027, 83.363676219109, 29), M )
(( 1.6765690500715, 73.98959022721, 23), M )
(( 1.6749260607992, 73.687015573315, 27), W )
(( 1.58582362825, 71.713707691505, 28), M )
(( 1.5893375739649, 74.248033504548, 27), W )
(( 1.6084440045081, 71.126430164213, 27), W )
(( 1.6048804804343, 82.049319162211, 26), W )
(( 1.5774196609804, 70.878214496062, 24), W )
(( 1.6799586185525, 75.649534976838, 29), W )
(( 1.7315642636281, 92.12183674186, 29), M )
(( 1.5563282000349, 69.312673560451, 32), W )
(( 1.7784349641893, 83.464562543, 26), M )
(( 1.7270244609765, 76.599791001341, 22), W )
(( 1.6372540837311, 74.746741127229, 30), W )
(( 1.582550559056, 73.440027907722, 23), W )
(( 1.722864383186, 79.37821152354, 20), W )
(( 1.5247544081009, 70.601290492141, 27), W )
(( 1.580858666774, 70.146982323579, 24), W )
(( 1.703343390074, 90.153276095421, 22), W )
(( 1.5339948635367, 59.675627532338, 25), W )
(( 1.8095306490733, 86.001187990639, 20), M )
(( 1.7454786971676, 85.212429336602, 22), M )
(( 1.6343303342105, 85.46378358014, 32), M )
(( 1.5983479173071, 79.323905480504, 27), W )
```

Some rules to follow:

1. **Handwrite, sign, and date (with date of submission) a copy of the Honor Code (shown below) and share the image as part of your project; a handwritten, signed, and dated (with the date of submission) copy of the Honor Code must be included with every project and exam submission. (Failing to include will cost 20 points)**
2. **Students are required to NOT share their solutions to the project even after the semester is over or even after graduation. However, they can show their projects during their interviews. They are also required to not discuss the solution with others or use anyone else's solution. Any violation of the policy will result in a 0 for this project for all students concerned.**

HONOR CODE

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code

I will not participate in any form of cheating/sharing the questions/solutions.