Homework 4 Part B

10-605/805: Machine Learning with Large Datasets

Due Wednesday, November 1st at 11:59PM Eastern Time

Instructions: There are two parts to this homework, which will have **different deadlines**.

- Part A is due on October 25th and is worth 20% of the grade. No Grace days can be used on this part of the homework.
- Part B (i.e., this document) is due on November 1st and is worth the remaining 80% of the grade.

For 10-805 students: You are required to complete all of the homework for a total point value of 80.

For 10-605 students: You should **not** complete questions 2.1.3 and 2.1.4. The total point value for 10-605 students is 72.

IMPORTANT: Be sure to highlight where your solutions are for each question when submitting to Gradescope otherwise you will be marked 0 and will need to submit a regrade request for every solution unhighlighted in order for fix it!

Note that Homework 4 Part B consists of two parts: this written assignment, and a programming assignment. Remember to fill out the collaboration section found at the end of this homework as per the course policy.

Programming: The programming in this homework is **NOT** autograded however you are required to upload your completed notebooks to Gradescope, otherwise you will not receive credit for the programming sections.

1 Part A: Data Conversion and Preparation

Please complete Homework 4 Part A following its write-up before attempting the coding part below.

2 Part B: Written

2.1 Regression, regression, regression

In this problem you will learn a regression model with the same dataset in several different ways. You may use any mix of manual calculation and computer code that you like. (You should only need short segments of code.) If you use any code, please include your code in the text of your written (part B) submission — not as part of the programming submission.

The dataset for the regressions is below:

x_1	x_2	const	y
1	1	1	1
1	0	1	2
0	1	1	1
2	1	1	-1

For each of the parts 2.1.1–2.1.4 below, you will need to solve a system of linear equations. Please be sure to include the **actual numerical equations** that you solve in the answer to your question, i.e., write out the matrix A and the vector b if you solve $Ax = b$. Hint: the function `numpy.linalg.lstsq` can be very helpful for solving systems of linear equations, including if you need to find the minimum norm solution. But please *do not* use its least squares functionality: i.e., if you want to solve a least squares problem, do it by constructing a set of linear equations that we can solve with zero residual error.

Note that 10-605 students need to solve only three of the five parts below (the first, second, and last), while 10-805 students need to solve all five parts.

2.1.1 Exact linear regression *[4 points]*

Solve the linear regression problem for this dataset exactly using the covariance matrix method. Be sure to report the numerical matrix and vector for the normal equations, as well as the learned weight vector.

2.1.2 Random projections method *[4 points]*

Solve the same linear regression problem approximately using random projections. Use the fast Johnson-Lindenstrauss method, with the following choices:

- Use one round of fast JL.
- Use the Walsh-Hadamard transform.
- Use random signs (not Gaussians).
- Use a 2d random projection. Recall that we get such a projection by sampling random rows from the transformed matrix or vector.

To make everyone's answers more uniform, please *do not* use truly random numbers. Instead, here are some tables of random numbers; please pick your "random" numbers from these tables in left-to-right order. Use as many as you need; there is no need to use all of them.

Random signs:

$$+1, +1, -1, +1, -1, -1, -1, +1, +1$$

Random indices in $1 \dots 4$:

$$3, 1, 4, 1, 2, 3, 3, 2, 1, 2$$

For reference, the 4×4 WHT matrix is:

$$\frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

Hint: one step of fast JL applied to a vector u can be written HSu, where H is a WHT matrix and S is a diagonal matrix of random signs. We can then pick out a subset of the entries of the vector HSu corresponding to the projection dimension we want.

Please include in your answer:

- The projection matrix you use.
- The projected dataset (both features and labels).
- The normal equations (in covariance form) for your projected dataset, as a numerical matrix and vector.
- The learned weight vector.

2.1.3 Kernel version (10-805 only) *[4 points]*

Now switch to using the squared-exponential kernel,

$$k(x, x') = \exp(-\tfrac{1}{2}\|x - x'\|^2)$$

Solve the kernel ridge regression problem exactly using the Gram matrix form, using ridge parameter $\lambda = 1$. Be sure to report the numerical matrix and vector for the regression equations, as well as the resulting example weight vector.

Make a prediction of the label for the following new point: $x_{\text{new}} = (0, 0, 1)$. Report the formula that you use to make the prediction (including numerical values for all quantities) as well as the final prediction.

2.1.4 Random Fourier features (10-805 only) *[4 points]*

Finally, use random Fourier features to approximate the same kernel ridge regression problem as in the previous part. As before, please *do not* use truly random numbers. Instead, use the table of "random" Gaussian samples below. Use just one random feature; this is obviously too few for a real problem, but will make calculation shorter.

Random Gaussian samples:

$$0.5, -1.2, -0.7, 0.3, 1.9, 2.1, -0.1$$

Be sure to report the numerical matrix and vector for the normal equations that you solve, as well as the resulting weight vector. Finally, make a prediction of the label for x_{new} (the same point as in the previous part). Report the formula that you use to make the prediction (including numerical values for all quantities) as well as the final prediction.

Hint: you can do complex arithmetic in Python. A complex number is something like `2+3j`. You can use `numpy.exp(2j)` for the complex exponential, and you can use `x.conj() @ y` for complex dot product. Note that Python uses j instead of i for the imaginary unit, and that you have to type something like `1j` instead of `j` to disambiguate an imaginary number from a variable named `j`.

2.1.5 When and why? *[4 points]*

For each of the methods that you used above (exact covariance-form regression, random projection for covariance form, exact kernel ridge regression, and random Fourier features for kernel ridge regression), please give a few sentences saying when we might use this method and why. (10-605 students: please answer for all four methods, even though you only solved two of them.)

3 Part B: Programming

In this part of the homework, you will perform exploratory data analysis (EDA) and data cleaning, and then train models with the original features. You will then perform feature engineering similar to what we did in Homework 1 (TF-IDF and Bag-of-Words), and then train models with these new features.

3.1 Setting up EMR and Spark

With our data ready in S3, it's now time to configure and create an EMR (Elastic MapReduce) cluster and run Spark, starting from the notebook `hw4.ipynb`. Include at least Hadoop, JupyterHub, and Spark in your cluster software configuration. Set `maximizeResourceAllocation` to true (see here) in software settings. Select your EC2 key pair.

Then, log in to jupyter (learn about login credentials here), upload your notebook, and start working!

Again, cost management is key. Because we might be running a cluster of machines, this could easily blow up your budget. We recommend using at most 1 Driver and 1 Core of type `m5.xlarge` while developing and debugging on a subset of MSD. You could scale this up to multiple Core workers when doing the final run. You may also utilize AWS spot instances to save cost during development.

Note that, although EMR is made up of EC2 instances, unlike EC2, you cannot stop an EMR cluster—you can only terminate it. You should plan your strategy accordingly. **Do not forget to download your code** before you terminate a cluster when you are done.

3.2 Preprocessing

Open JuptyerHub on your EMR cluster, similar to what you did on HW3, and upload your `hw4.ipynb` to begin working on it!

3.3 Exploratory Data Analysis *[11 points]*

(a) *[1 points]* Explain why the two features seem problematic (after performing .summary() operation).

(b) *[3 points]* Histograms (remember to label them)

(c) *[1 points]* Explain what is strange about `year`'s distribution and what might cause this. Describe how you could filter `year` to make its histogram look more balanced.

(d) *[1 points]* New histogram for `year`.

(e) *[3 points]* Provide plots for the three pairs. Describe your findings.

(f) *[2 points]* Think about what simple technique you could use to visualize large datasets while retaining a similar data distribution. Briefly describe what you did.

3.4 Data Cleaning *[8 points]*

(a) *[1 points]* Your justification for dropping the two features.

(b) *[2 points]* Compare the two numbers and explain the advantages and potential problem of doing this step. What other techniques could you use to potentially do better?

(c) *[1 points]* State the two features.

(d) *[2 points]* Explain your proposed solution and discuss its pros and cons.

(e) *[2 points]* Report the percentage:

3.5 Baseline *[13 points]*

(a) *[2 points]* Explain why treating this as a classification problem might be a sensible choice.

(b) *[1 points]* Report what percentage of songs are assigned the "popular" label.

(c) *[1 points]* Explain why we shift the year.

(d) *[2 points]* Explain what scaling means and why we want to perform scaling before the learning step.

(e) *[3 points]* Explain the difference between these two metrics and when AUC might be more useful than accuracy.

(f) *[4 points]* Calculate the train and test AUC of both models and report them.

Models	Train AUC	Test AUC
Logistic Regression		
Random Forest		

3.6 Featurization: Bag-of-Words and TF-IDF *[8 points]*

(a) *[3 points]* Explain what the `vocabSize` hyperparameter means in the context of Bag-of-Words.

(b) *[3 points]* Other than featurizing texts, what other feature engineering would you do on the dataset? Briefly describe one.

(c) *[2 points]* Explain where this number "31" comes from.

3.7 Modeling with New Features *[10 points]*

(a) *[4 points]* Evaluate train and test AUC for each model and report them.

Models	Train AUC	Test AUC
Logistic Regression		
Random Forest		

(b) *[6 points]* Include the plot and your explanations.

3.8 Do Your Best *[7 points]*

(a) *[2 points]* Your final AUC:.

(b) *[3 points]* Your model and hyperparameters.

(c) *[2 points]* Describe your approach.

3.9 Reflection

[3 points] What challenges did you face in HW4 Section 3? How did you overcome these challenges? What did you learn from HW4 ?”

4 Collaboration Questions

1. (a) Did you receive any help whatsoever from anyone in solving this assignment?

   (b) If you answered ‘yes’, give full details (e.g. “Jane Doe explained to me what is asked in Question 3.4”)

2. (a) Did you give any help whatsoever to anyone in solving this assignment?

   (b) If you answered ‘yes’, give full details (e.g. “I pointed Joe Smith to section 2.3 since he didn’t know how to proceed with Question 2”)

3. (a) Did you find or come across code that implements any part of this assignment?

   (b) If you answered ‘yes’, give full details (book & page, URL & location within the page, etc.).