CSCI 3223: Tashfeen's Machine Learning II

Homework 3

You can download all the code in this homework as hw3.zip.

Question 1. Recall that Ridge regression minimises the following error function, where α is the regularisation constant.

$$J(\beta) = (X\beta - Y)^T(X\beta - Y) + \underbrace{\overbrace{\alpha}^{\substack{\text{Regularisation}\\\text{Constant}}} \cdot \beta^T\beta}_{\substack{\text{Penalty}\\\text{Term}}}$$

Generate data as shown in listing 1 and then for each $\alpha_i \in$ `np.logspace(-10, -2)` use the Scikit-learns's `sklearn.linear_model.Ridge` and plot $(\alpha_i, \beta_{i,1\leq j\leq p})$ This is `Ridge.coef_` plotted against `Ridge.alpha`. Note that you need to pass `fit_intercept=False` to the Ridge constructor and scale your x–axis logarithmically. Use the error function $J(\beta)$ to explain the trend in the plot. Put the code for this question in a file called `regu.py`.

```
n = 7
X = 1 / (np.arange(1, n + 1) + np.arange(0, n).reshape(-1, 1))  # n by n
y = np.ones(n)
alphas = np.logspace(-10, -2)
```

Listing 1. Hilbert space data.

Question 2. Instead of smoothly minimising β_i like Ridge, Lasso regression minimises the less relevant β_i to zero. Therefore, we may use the β evaluated by Lasso for feature selection or dimensionality reduction. Load the Boston house-price data like this,

```
# https://lib.stat.cmu.edu/datasets/boston
PATH = '../media/BostonHousing.csv'
legends = [
    'crim', 'zn', 'indus', 'chas', 'nox', 'rm', 'age', 'dis', 'rad', 'tax',
    'ptratio', 'b', 'lstat', 'medv'
]  # Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
Xy = np.genfromtxt(PATH,
                   delimiter=',',
                   skip_header=1,
                   encoding='UTF-8',
                   dtype=None,
                   converters={
                       3: lambda s: float(s[1:-1]),
                       8: lambda s: float(s),
                       9: lambda s: float(s)
                   })
legends.remove(legends[-1])  # remove prediction value legend
X, y = Xy[:, :-1], Xy[:, -1]
X = StandardScaler().fit_transform(X)
```

Listing 2. Loading the BostonHousing.csv.

Gradually train 10 different Lasso models with each model's `max_iter` set to its index and $\alpha = 1$. For example, the first model will be trained for 1 iteration, the second for 2, third for 3 and so on. Record the β for each of these and you'll accumulate a 10 by 13 matrix. Plot each of the rows of this matrix in the same figure. Which features have converged to zero? What does that mean? Code should be put in `house.py`.

Question 3. While Ridge regression uses the ℓ^2 or the Euclidean norm in the penalty term, Lasso regression uses the ℓ^1 or the Taxicab norm, more commonly referred to as the Manhattan distance. Lasso minimises the same quadratic error function but with a penalty term that involves absolute values.

$$J(\beta) = (X\beta - Y)^T(X\beta - Y) + \alpha \sum_{i=1}^{p} |\beta_i|$$

1) As such, just like the Ridge or the Ordinary Least Squares regression, can you analytically workout a solution for β? Justify your answer.
2) Read the `sklearn.linear_model.Lasso` documentation and state the method that is used to minimise the Lasso regression's error function. Hint: How are the data points fit?
3) Read `titanic.py` (the csv file is the same from previous homework). What does it do?
4) At the end of the `titanic.py`, build a Lasso model using the transformed data. Then, build a bar graph where each bar is the Lasso regularised slope β_i. Label the x–ticks with the transformed feature names. According to this bar graph, which features are the most important?

Question 4. Assure that you can download and read in the following two MNIST database of handwritten digits' test files. Read them into the appropriate Numpy variables as in the previous assignments. Be sure to normalise the pixel values from $0 \leq x \leq (2^8 - 1)$ to $0 \leq x \leq 1$.

- `t10k-images-idx3-byte`
- `t10k-labels-idx1-ubyte`

Use the hyper-parameters given bellow for the Scikit-learn's K–means model. Fit the entire MNIST test set with clustering. Upon converging, what is the sum of the squared distances of the images to their closest cluster center (hereafter called inertia)?

```
n_init='auto', n_clusters=10, random_state=0
```

Question 5. Perform the principal component analysis (PCA) and transform the images in the MNSIT test set we clustered above.

1) How many of the principal components explain exactly 90% of the data variance? Illustrate this with a plot.
2) Use the amount of principal components determined in the previous part to obtain the equal number of transformed data columns. Use this trimmed data to perform clustering again as in question 4. What is the inertia upon convergence this time?

Question 6. Fit a Lasso regression model with $\alpha = 0.0112$ onto the image data,

(a) How many non-zero β_i coefficients remain once the coordinate descent stops?
(b) Pick the columns corresponding the $\beta_i \neq 0$ from the data and yet again, perform clustering as in question 4. What is the inertia?

Question 7. Between Lasso and PCA pre-processing which technique allowed for the best F1 score? What were the F1 scores for each of the techniques? Does better clustering necessarily mean better classification? Discuss.

Question 8. Between Lasso and PCA, which algorithm is better when,

(a) The labels are not known.
(b) Final features need to be interpretable.

Question 9. Find a dataset at `https://www.openml.org/` and read it as shown in the article Column Transformer with Mixed Types. Perform either Lasso or PCA reduction and report which features are the most important. Please briefly describe the dataset and the problem as well as what do all of the features mean. Place the code for this question in a file called `custom.py`.

Submission Instructions

1) Submit a PDF that answers the questions and contains all the plots that the assignment asks for. Please *read all the questions* and make sure you give the plots! Just turning in the code is not enough.
2) Submit your `regu.py`, `house.py`, `titanic.py` & `custom.py`. Place all the code between question 4 and question 7 in a file called `experiment.py`. If this Python file does not run on the console with `python3 experiment.py` after any necessary path adjustments, the submission will receive no credit.

Oklahoma City University, Petree College of Arts & Sciences, Computer Science