Machine Learning Final Report
Data Completion and Interpolation

Bowei Feng(bf289)
Cong Deng(cd745)

May 11, 2019

1 Model

1.1 Approach

We use linear regression, knn and natural language processing methods to train and interpolate the missing data.

1.1.1 Linear Regression

Linear regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.
The model is described as below:

$$y = wx + b$$

Loss function is defined as below:

$$J(w) = \frac{1}{2m} \sum_{i}^{m} (h_w x^{(i)} - y^{(i)})^2$$

$$h_w(x) = \sum_{i=0}^{n} w_i x_i$$

And our goal is to calculate:

$$minJ(w)$$

1.1.2 Natural Language Processing

Natural language processing is a kind of method to program computers to process and analyze large amounts of natural language data. NLP frequently involves speech recognition, natural language understanding, and natural language generation. Here we mainly implement the natural language understanding in this paper.

1.1.2.1 Natural Language Understanding

The system uses a lexicon of the language and a parser and grammar rules to break sentences into an internal representation and labels them with different numbers.

1.2 Design Choices

For different type data, we design different method to predict the value of them.
For the int and float data, we would like to choose linear regression, set the predict parameter as the y and other parameters as x, we want to find a suitable weights for each x to determine the y, and since these data are int or float, which means the calculation will be quick and accurate.
For the string data, we would like to use two different way to see the accuracy of them. First, we use label encoder to process the string data, which will transform the string data to the int data, which could make these data be predicted by the linear regression. And also, we try the knn to classify these string data, we just think these data as different type, which is determined by other data.

1.3 Represent Data

For the int and float data, we just transform the NA data into -1, and put them into test dataset.
For the string data, one way is to use the label encoder to transform the data into the digital data and NA to -1, which is now like the int and float data. Another way is to think these string data as label, and just put the NA data into the test dataset.

1.4 Evaluate

We first run our regression algorithm on the train dataset to get the weights and the bias, the we use the train x data to do the predict, compare the result with the train y data to evaluate the goodness of fit.

1.5 Irrelevant Variables and Missing Data

First, when modeling, we took a look at the data and its corresponding problems. We found that there were some researcher/experimental environment features which were not useful for our model like notes, date, attention test nand etc. These features are so called irrelevant variables but different from the irrelevant variables that we will find later in the training process. Because those found in the training are only proved “irrelevant” in this particular model. In 1.5, we mainly talk about those irrelevant variables that can not be useful in any model.

As for the missing data, we regard them as the test data set and others as train data set. By this way, we will use those existing data to fit/predict the missing data.

2 Training Algorithm

2.1 Fit the Data

We write a split function in our data class, each time we just choose one column data which is set as the y data. And then, depend on the value of the y, as we have talked above, when y is -1, we will split these data into test data, and the rest data will become the train dataset, at last, we we could get the x_train, y_train, x_test and y_test four dataset.

2.2 Were you able to train over all features?

When we do the linear regression on the int and float data, we will ignore all the string data since these data is hard to use in our regression, and only care about the same type data could help our regression quick and accurate.

2.3 Tradeoffs

There are two main tradeoffs in our algorithms.First is the alpha in the lasso regression, when the alpha is too big, all weights are approaching 0, and when the alpha is too small, which will make the regression not converge, and may cause precision problems. Second is the lambda in the ridge regression, we try different lambda in the ridge regression, and see the accuracy of the result, and finally we choose the 0.2 as the lambda value.

3 Model Validation

3.1 Avoid Overfitting

We use regularization method to avoid overfitting in this case. To be more specific, we use L2 Regularization and early stopping when training. L2 regularization is a common method to reduce the value of some parameters to help simplify the model.

3.1.1 L2 Regularization

we use the ***L2 Regularization*** which is also called ***Ridge Regression***. It adds “squared magnitude” of coefficient as penalty term to the loss function.

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Here, if lambda is zero then we get back ordinary least squares. However, if lambda is very large then it will add too much weight and it will lead to underfitting. It’s important how lambda is chosen.

Regularization reduces all the parameter values (weight) of our loss function to achieve the reduce on the unimportant parameters and optimize the degree of these penalties. We do not know which parameter needs penalty, therefore, we need to modify the loss function with regularization. When we add an additional regularization, we reduce each parameter.

But why can we avoid overfitting by reducing the parameters? In fact, the smaller the value of these parameters, the smoother the function, which is also simple. Therefore, the problem of overfitting is less likely to occur, thereby improving the return capability of the model.

3.1.2 Early Stopping

Early stopping is a simple regularization method that only needs to monitor the performance of the testing set. If program finds that the performance of the test set is no longer improved, stop training. This method is very important in the absence of big data, because the model usually begins to fit after 5 to 10 or even fewer iterations.

3.2 Size of Data

Originally, we have 3,000 pieces of the data and after data cleaning, we have 2,500 for training and interpolating. This is not a big data set, we need to make use of every data as we can.

4 Evaluate

4.1 Success and Lack

To evaluate the accurate and the performance of the model, when we get the prediction of one feature of the test data, we try to find the same data in the train data at first, if we

could not find a same data in the train dataset, we will find the most similar data in the train dataset, and then, we we now see from the weights data of this feature, and make sure how many other features work on this prediction feature, and at last, we would like to compare these features of two data, and see the differences of these two data, and see the least square error of two data, and the less the error is, the accurate of the model is better.
There is an apparent lack for our model, which is that when one feature has two many values, which means it has a large range, the model could not converge well, at the same time, the robustness of the model will decrease.

4.2 Good and Poor Features

Weight is the basis for us to judge whether a feature is important for forecasting. The screenshot below is our weight output.

```
For the feature 'backcount1':
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.05754958411968758, 0, 0,
 0, 0, 0, 0, 0, 0, -0.2174774939916304, 0, 0, 0, 0, 0, 0, 0.03013802201115673, 0.08238691427315126, 0, -0.07441857
754681236, 0, 0.09842992037468346, 0.17694061538761385, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -0.09725607932558411,
-0.17217270537902074, -0.12226139826001181, -0.1462186940664716, 0, -0.14578233567645782, -0.0636405433814429, 0,
0, 0.06784442518703038, -0.04710318080321665, -0.06112855951847396, 0, 0.06241075402297272, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0.048355526968126734, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.12157788291995908, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.1316881498351199, 0
, 0, 0, 0, -0.05721772277440085, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.14784322851477372, 0, 0, 0, 0, 0, 0.7127564638474642, 0, -0.21658852469103795, 0, 0, 0, 0.1
1962109244961877, 0, 0, 0.134663535302447, 0.09453020785989397, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -0.04415245019
597158, 0, 0, 0]
```

It can be seen from the picture, many features have been reduced to 0 after the regularization process. We can say that these features are not useful at all. It can also be seen that some features take a weight more than 0.5 and these are the useful features.
From the weight result, we found that the several features connected with stroop effects are particularly relevant to predicting answers. There stroop effects are shown below:

GZ	HA	HB	HC
stroop_order	stroopinstructions_order	stroopinstructionstest_order	strooprac_order

Just a gauss, stroop effect can reflect the degree of concentration of the testee. This may somehow affect the experimental results(people who are concentrated may make the same answer on a certain question as the unconcentrated people do)

5 Data Analysis

5.1 Invalid Data

There are not only many empty data in the data set, but also many unrelated data. These are invalid data. It is very necessary to classify and clean the data before training and testing. Otherwise, invalid data will put a negative effect on the training result.

5.1.1 Irrelevant Data

Irrelevant data, in this paper, is defined as the objective data like site, date, timestamp, participant id, and etc.

5.1.2 Empty Data

Empty data, in this paper, is defined as the data only contains only irrelevant data (defined in 5.1.1) but no useful information. For this kind of data, we do casewise deletion.
Casewise deletion is the removal of a sample containing missing values. The results of this approach may result in a significant reduction in the effective sample size and the inability to make full use of the data already collected. Therefore, it is only suitable for cases where the key variable is missing, or the sample with invalid or missing values is small.

6 Data Generation

6.1 How good does it look?

When we use the system to generate realistic data, we try to generate 1000 data, and for each data, we just random pick from the original dataset for the ML3ALLSite.csv, and now, we generate a new data in which there are still some missing data, and we use our system to train and do the prediction, we can finally get a dataset in which all data is filled.

6.2 What does it mean for it to ‘look good’?

And now, we want to compare the data with the real data, for each row of data, we try to find the most same data in the real data, the most same means two data has as many

as same features, and we compare their different features, and we use the ratio of the number of different features with the number of same features to evaluate weather it looks good.