|  |  |
| --- | --- |
| **Capstone Project**  **Machine Learning Engineer Nanodegree**  **“Learning” from Chinese Learning Log Data** | Jed Isom  April 18th, 2016 |

**Definition**

**Project Overview**

In 2011, I was interested in learning Mandarin Chinese and as part of that interest I decided to read a book in Mandarin. The book I decided to read had 3 columns on each page. In the 1st column, it had traditional Chinese characters, in the 2nd column it had the pinyin pronunciation of the characters in the 1st column, and the 3rd column had an English translation. As I was reading, if I didn’t know a Chinese character, I would look to the pinyin to see if it would jog my memory. If that didn’t work, I would look up the character in [this iOS app](https://itunes.apple.com/ca/app/pleco-chinese-dictionary/id341922306?mt=8), write the character, its pinyin and English translation in a notebook.

A couple of days into this reading process, I decided to track my progress because I’m a data driven guy and thought it would be interesting. I did this by recording how long I read each day (in minutes) as well as how many Chinese characters I read. I expected the average seconds taken to read a character to follow the traditional logarithmic scale learning curve. As I plotted my progress, there it mostly followed the expected curve, but there were some very obvious exceptions to the curve. I found that there were certain sections of the book, or periods of time, where I read much slower than I had previously. There were also instances where I read quicker than I would have expected.

In this project I am looking to find a way to explain more of the variance in my reading speed performance. I’ll use the logarithmic learning curve as the baseline performance for this supervised learning project. Luckily, this book can also be found and read online for free [here](https://www.lds.org/scriptures/bofm?lang=zho). To create the dataset, I used my handwritten notes to create a raw dataset where each row represents a date/study session. The columns in the raw dataset include

* Date
* Time spent reading (in minutes)
* Text read; copied and pasted text from the book that represents the text read during the study session. This was recreated digitally for this project based on my detailed notes I took while reading the book.

I plan to use the information within the ‘Text read’ field to help improve the prediction of my reading speed. Some potential features I’ll explore to explain some of the error in the baseline model include

* % of characters in study session text not seen before in the book
* Get the number of days since each character in the text was last read, and then take the mean.
* Potentially use Latent Dirichlet allocation (LDA) to create topics and use the posterior probabilities of the text being in a certain topic as a predictor
* …

I **can’t** use the number of characters in the text read field or the time spent values as predictors in this project. These 2 features are what I used to create the actual output (y variable) and so would easily fit the data, but not solve the underlying question about why my reading speed did not follow the traditional learning curve rate.

**Problem Statement**

I plan to determine what underlying variables in the text that I was reading could have better predicted my Chinese character reading speed while reading this book. I will do this by treating this project as a supervised learning problem, where my output (y) is my reading speed in seconds per Chinese character, and my inputs (X) will be features I will derive from the text data and cumulative time spent or cumulative characters read. These last 2 features will represent the baseline learning curve model I’m using to compare against the supervised learning performance.

Tasks required to complete this project include

1. Use hand-written notes to create the digital dataset for analysis
2. Import the MS Excel entered dataset into Python
3. Scrub the Chinese Unicode text by removing numbers, punctuation, non-Chinese characters, etc.
4. Calculate preliminary characters per study session and seconds per character metrics
5. Validate data entry against handwritten notes to make there are no big mistakes in the data entry, go back and correct #1 as necessary
6. Create X features from the dataset to predict seconds per character. Potential features to create that might help explain the variation are listed again here.
   1. % of characters in study session text not seen before in the book
   2. Get the number of days since each character in the text was last read, and then take the mean.
   3. Find the frequency of characters in the current study session text based on frequency of those characters in all previous study sessions
   4. Potentially use Latent Dirichlet allocation (LDA) to create topics and use the posterior probabilities of the text being in a certain topic as a predictor
   5. …
7. Segregate training and test set data. This segregation will be random and static so that the same examples are always in the training set and test set. This will prevent data snooping.
8. Perform supervised learning on training set with cross validation to estimate generalized model performance
9. If cross validated model performance is worse than current best, then go back to #6 or #8 and try to create new/better features or learning models for predicting reading speed.
10. I’m not sure how I’ll know when to stop this loop from 6 to 9, but I will likely start to get diminishing returns on new/different features and stop at that point. At this point, I’ll have a solid set of predictors that will help me understand what other features determined my reading speed through the book.

In the end, I believe that using supervised learning I will be able to determine what features of the text I read have the most predictive power for my Chinese character reading speed. This will be valuable because I can use this information to improve my study habits based on this information. In a future project, this type of information might also be used to help someone else learn to read Chinese characters faster.

**Metrics**

As mentioned previously, the output variable, y, is the seconds per character reading speed for the study session. Since this value is very large in the beginning of the book and decreases exponentially due to the learning curve effect, I plan on taking the natural logarithm of this value for model fitting purposes. This will help the model error to be more uniform across the study sessions and thus not over fit study sessions near the beginning of the book.

The project is a regression problem. Kaggle has a [list of metrics to consider using for regression problems](https://www.kaggle.com/wiki/Metrics):

* Mean absolute error
* Weighted mean absolute error
* Root mean squared error
* Root mean squared logarithmic error

Based on what I know about the problem either simple mean absolute error or root mean squared error should work fine. I’ll be using root mean squared error because it is more commonly used, and also because it penalizes predictions farther from the actual values much more than predictions closer to the actual values due to the squared term. This helps incentivize the model to fit these data points farther from the predictive model better, which is essentially what this project is about; fitting the data that varies from the baseline model better.

I also want this model to generalize well because I am trying to use the information about what features work well in predicting reading speed to help me understand how to learn better/faster and improve my reading speed in the future. So I will be using the cross-validation root mean squared error of the model applied to my training set as my target to optimize. Once I have confidence that the model will generalize well in cross-validation I’ll run the model on the test set and report the root mean squared error on the test set with the cross-validated model.

**Analysis**

*(approximately 2 - 4 pages)*

**Data Exploration**

As described previously, I copied and pasted text that I read during each study session into a digital dataset for this project’s analysis. Below is a sample row/instance of one study session of data.

|  |  |  |
| --- | --- | --- |
| Date | Time\_Spent | Text\_Read |
| 2011/09/07 | 30 | 他​心中​愚蠢​的​幻想。   12 最年長​的​拉曼​和​雷米爾​這樣​抱怨​他們的​父親。​他們​抱怨，​因為​他們​​不​知道​創造​他們的​神​的​作為。   13 他們​也​不信​耶路撒冷​那​座​大​城 |

In Excel, I was able to create a custom date format YYYY/MM/DD that would make it easier to transfer the dataset to the Python datetime package that takes (YYYY, MM, DD). The “Time\_Spent” column is the time spent reading during that study session in minutes. It was too much effort to record this data in a more granular form like seconds, so this will be a somewhat lumpy source of error because the seconds read during a session will jump in increments of 60 seconds. The “Text\_Read” column includes the unformatted/unfiltered text I read during the study session and copied into the dataset. Since the text data are Chinese characters, the “Text\_Read” feature is Unicode text, not ASCII text/characters. While Unicode text is the future of digital text representation, figuring out how to use this text in a machine learning problem was a bit of a challenge for me (more discussion on this point later).

At a high level, this dataset contains 974 records (study sessions) spanning 29,227 minutes of study time. After preprocessing the text data (see the discussion on how this was done in the preprocessing section below) I was able to count 334,195 total characters read. This gives an overall reading speed average of 5.247 seconds per character when not taking the natural log of the reading speed.

The majority of the values for “Time\_Spent” are 30 minutes because of the habit I developed of reading for ½ hour in the morning each work day before going to work. There are a few outlier days where I spent an hour, or an hour and a half studying. I am accounting for this variation by selecting reading speed as my output metric instead of characters read during the study session. Essentially this normalizes the output variable

As discussed previously, because this dataset represents a learning/experience curve process, the reading speed values are exponentially larger in the beginning of the dataset compared to the end of the dataset. To deal with this order of magnitude problem, I plan to take the natural logarithm of the reading speed and use that as my output variable for this supervised learning problem.

**Exploratory Visualization**

To visualize the experience/learning curve baseline I used in this project, I plotted the natural logarithm of the reading speed against the natural logarithm of both the cumulative time spent reading and the cumulative characters read. Taking the logarithm of both the X and the Y axis allows for a linear fit to most [experience curves](https://en.wikipedia.org/wiki/Experience_curve_effects). Both charts have roughly the same shape with a somewhat flat beginning section and then follow a linear downward trend fairly well. After viewing both of these charts, I’ve decided to use the natural logarithm of the cumulative characters read as the baseline learning curve because the data fits the linear model better, and because experience curves are normally based on “units produced” which in my case would be characters read.



Need to Redo justification for baseline picked

You can also see in these charts the reason for this analysis. In an experience/learning curve like this you would expect the reading speed to almost monotonically decrease (with some noise). If you look at the region between 5,000 and ~17,000 reading minutes, there appears to be a hump or plateau in the data. There is also relatively more noise in the data near the end of the data set between 25,000 and 30,000 reading minutes. Finding a way to explain these anomalies in the data with the latent information in the text read is the focus of this project.

**Algorithms and Techniques**

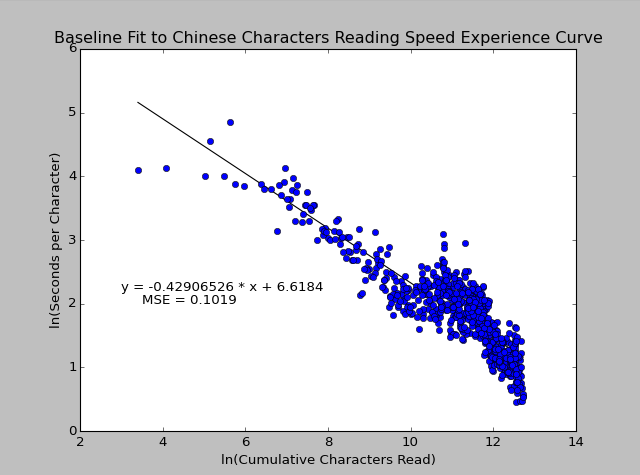
In this section, you will need to discuss the algorithms and techniques you intend to use for solving the problem. You should justify the use of each one based on the characteristics of the problem and the problem domain. Questions to ask yourself when writing this section:

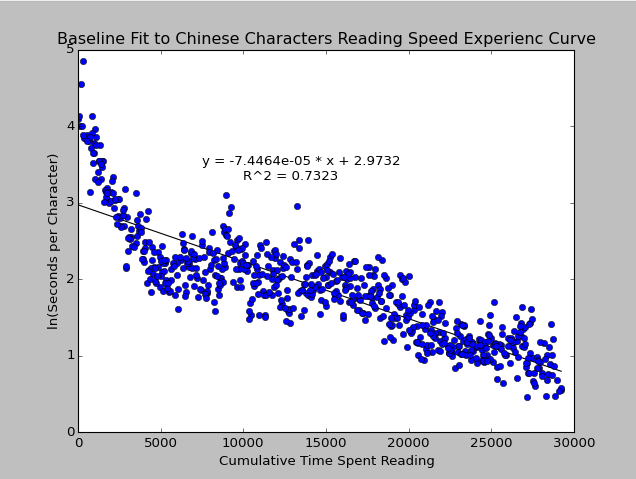
* *Are the algorithms you will use, including any default variables/parameters in the project clearly defined?*
* *Are the techniques to be used thoroughly discussed and justified?*
* *Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen?*

**Benchmark**

As discussed previously, I used a traditional experience curve model as the benchmark for this project. [Experience curves](https://en.wikipedia.org/wiki/Experience_curve_effects) look non-linear on a standard scale, but *should* look linear is both the Y and the X axes are logarithmic scale. So, my baseline model is a linear regression model with X = ln(cumulative time) and Y = ln(seconds per character). This baseline model along with its linear equation and MSE are shown in the plot below

I need to update this graphic so that is has MSE instead of R^2





In this section, you will need to provide a clearly defined benchmark result or threshold for comparing across performances obtained by your solution. The reasoning behind the benchmark (in the case where it is not an established result) should be discussed. Questions to ask yourself when writing this section:

* *Has some result or value been provided that acts as a benchmark for measuring performance?*
* *Is it clear how this result or value was obtained (whether by data or by hypothesis)?*

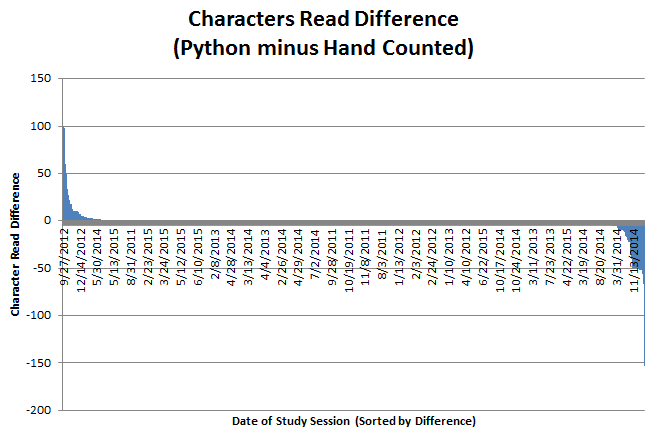
**Methodology**

*(approximately 3 - 5 pages)*

**Data Preprocessing**

In order supervised machine learning to be possible on this dataset there were several transformations required on the raw dataset

1. The raw data needed to be imported into a pandas dataframe from a Microsoft Excel file
   1. The ‘Date’ field needed to be parsed and formatted to be turned into a python datetime variable type
   2. The ‘Time\_Spent’ field was read in as text and transformed to an integer (only whole numbers were used in this field)
   3. The ‘Text\_Read’field was copied in as Unicode text
2. The Unicode text in the ‘Text\_Read’ field had to be stripped of all non-Chinese characters
   1. All numbers and whitespace (‘\n’ and ‘ ‘) were removed. There were no tabs (‘\t’) in this dataset
   2. Punctuation was removed
   3. A couple weird characters that were found during data entry quality checks (see 2.d.ii below) were removed. The Unicode text for these characters include
      1. u'\u200b': a zero width blank character that was throwing off character counts
      2. u'\u2500': a dash that for some reason was not part of the punctuation list used
      3. u'\xa0': a Unicode space character that was not removed during standard whitespace removal
   4. Character counts were calculated for each study session and validated against my hand written counts. During this important data entry validation process I found and corrected a couple problems.
      1. There were records where I accidentally copied some of the text into the wrong day (e.g. I actually read more text the first day, but accidentally copied some of the text into the second day). These were easy to find because the differences between the python counts and hand counts were equal and opposite (e.g. -13 and +13) on days right next to each other)
      2. As I checked other records where the python counts and hand counts didn’t match, I found the extra characters in 2.c above and modified the python code to remove them
      3. I was not able to remove all of the differences between the 2 because, as it turns out, I don’t count large numbers very well by hand. I double checked many of my manual counts and found that my original manual count was wrong and the python count was right. Below is a chart summarizing the discrepancies between the final python character count and the original manual count. You can see that there are a couple days where the count was very off. I had a habit of counting by 50’s and at times I accidentally lost track of how many groups of 50 I had counted. You can see on the right portion of the chart that I had a bias of assuming I had read more than I actually had.

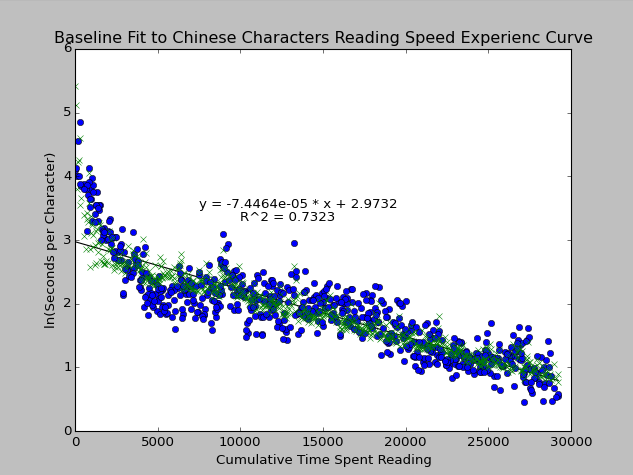


1. Features were created from the cleaned up text in order to use them for supervised learning
   1. Features representing the cumulative time spent reading and cumulative characters read were created to establish the baseline (experience curve) model. These were used to create the baseline experience curve model
   2. Create a feature that give the percentage of characters in the record’s text that have previously been in the ‘text read’ field in the dataset. I had to create this feature before splitting the training and test datasets. To create this feature I did the following
      1. Use CountVectorizer from the sklearn.feature\_extraction.text package to create a sparse document term matrix for the entire dataset corpus.

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section:

* *If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented?*
* *Based on the****Data Exploration****section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected?*
* *If no preprocessing is needed, has it been made clear why?*

**Implementation**



In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

* *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?*
* *Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?*
* *Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

**Refinement**

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

* *Has an initial solution been found and clearly reported?*
* *Is the process of improvement clearly documented, such as what techniques were used?*
* *Are intermediate and final solutions clearly reported as the process is improved?*

**Results**

*(approximately 2 - 3 pages)*

**Model Evaluation and Validation**

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model’s solution is affected (this is called *sensitivity analysis*). Questions to ask yourself when writing this section:

* *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?*
* *Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?*
* *Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?*
* *Can results found from the model be trusted?*

**Justification**

In this section, your model’s final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

* *Are the final results found stronger than the benchmark result reported earlier?*
* *Have you thoroughly analyzed and discussed the final solution?*
* *Is the final solution significant enough to have solved the problem?*

**Conclusion**

*(approximately 1 - 2 pages)*

**Free-Form Visualization**

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

* *Have you visualized a relevant or important quality about the problem, dataset, input data, or results?*
* *Is the visualization thoroughly analyzed and discussed?*
* *If a plot is provided, are the axes, title, and datum clearly defined?*

**Reflection**

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

* *Have you thoroughly summarized the entire process you used for this project?*
* *Were there any interesting aspects of the project?*
* *Were there any difficult aspects of the project?*
* *Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

**Improvement**

Further improvements:

* Group characters by radicals and give memorizing complexity score?
* N-grams for common combinations of 2+ character words (word association mining & analysis)
* Part of speech tagging?
* Use Entropy calculation as predictor?

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

* *Are there further improvements that could be made on the algorithms or techniques you used in this project?*
* *Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?*
* *If you used your final solution as the new benchmark, do you think an even better solution exists?*

Before submitting your report, ask yourself…

* Does the project report you’ve written follow a well-organized structure similar to that of the project template?
* Is each section (particularly Analysis and Methodology) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
* Would the intended audience of your project be able to understand your analysis, methods, and results?
* Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?
* Are all the resources used for this project correctly cited and referenced?
* Is the code that implements your solution easily readable and properly commented?
* Does the code execute without error and produce results similar to those reported?