CSL 341 – Fundamentals of Machine Learning

Automatic Essay Scoring

Project Report

Submitted By:

Amol Mittal - 2010MT50584

Ashwin Kumar - 2010CS10211

Vedant - 2010CS10263

1. Overview

This project aims to build a machine learning system for automatic scoring of essays written by students. The project idea has taken from kaggle.com (http://www.kaggle.com/c/ASAP-AES).

The basic idea is to search for features which can model the attributes like language fluency, vocabulary, structure, organization, content etc. As we pointed out in our initial draft, such a system can have a high utility in many places. For instance, currently, evaluation of essay writing section in exams like GRE, GMAT, and TOEFL is done manually. And, so automating such a system may prove to be highly useful.

We have built a linear regression model with polynomial basis function to predict the score of a given essay. The subsequent sections explain the input data, features extraction, detailed approach, results, and future scope of the work.

---

2. Problem Formulation and Input Data

We have taken the input data from Kaggle.com. We are given ~13000 essays written by school students of Grade 7, 8 and 10. These essays are divided into 8 sets - each set of essays from a different context - to ensure variability of the domain. Each set of essays was generated from a single prompt. Along with the ASCII text of each essay, we also have scores given to each essay by two human evaluators and a combined resolved score.

We split this data into two sets: training set and testing set, as follows:

Essay set	Total number of Essays	Number of Essays used for training	Number of Essays used for testing
1	1783	1200	583
2	1800	1200	600
3	1726	1200	526
4	1772	1200	572
5	1805	1200	605
6	1800	1200	600
7	1569	1200	369
8	723	500	223
TOTAL	**12978**	**8900**	**4078**

Once we have the training and testing data, we can extract features from each of the document and train our model. These are explained in subsequent sections.

---

3. Features Extraction

Feature extraction is the most important part of any machine learning task and so is the case with us. To build effective essay scoring algorithm, our aim is to try to model attributes like language fluency, grammatical and syntactic correctness, vocabulary and types of words used, essay length, domain information etc.

At present, our model is using the following set of features extracted from the ASCII text of the essays:

i) Word count and Sentence count: These are very basic features of any text document and do influence the scoring of the document as well. So, to extract these features, we use the "*textmining*" library (https://pypi.python.org/pypi/textmining/1.0) in python. Using this library, we extracted the term-document matrix for our training corpus (8900 documents). This library also provides a list of 276 common stop words in English language. Now, since these stop words are of not much importance, we skipped them while calculating the word count of each document from the term-document matrix.

To get the sentence count, we simply split the document using '.' and thus, count the number of segments obtained.

ii) POS Tags: Another crucial set of features for evaluating any piece of writing is the number of words in various syntactic classes like nouns, adverbs, verbs, adjectives etc. These features are crucial for evaluating the quality of content in the essay. To get the counts of words in each POS (part-of-speech) class, we use the NLTK library in python (http://nltk.org/). This library gives us the POS tag for each word in an essay, and thus, we extract the number of nouns, adverbs, adjectives and verbs.

iii) Spelling Mistakes: An important parameter while scoring an essay is the spelling mistakes. So, number of spelling mistakes in an essay is also a feature for our model. To get this number, we use the spell checker provided in python by library named 'enchant'(https://pypi.python.org/pypi/pyenchant).

iv) **Domain Information Content:** It is perhaps the most crucial feature in our model as it tries to captures the semantics and information content of an essay. To get this feature, we first took the best essay from each set (highest scored essay) then, we pulled out nouns from that essay. These serve as keywords for the particular domain. Then, we fire these words into 'Wordnet' (http://wordnet.princeton.edu/) and take out their synonyms. In this way, for each set, we get a bunch of words, most relevant to its domain. Then, we count the number of domain words in the given essay.

---

4. Approach, Evaluation and Results

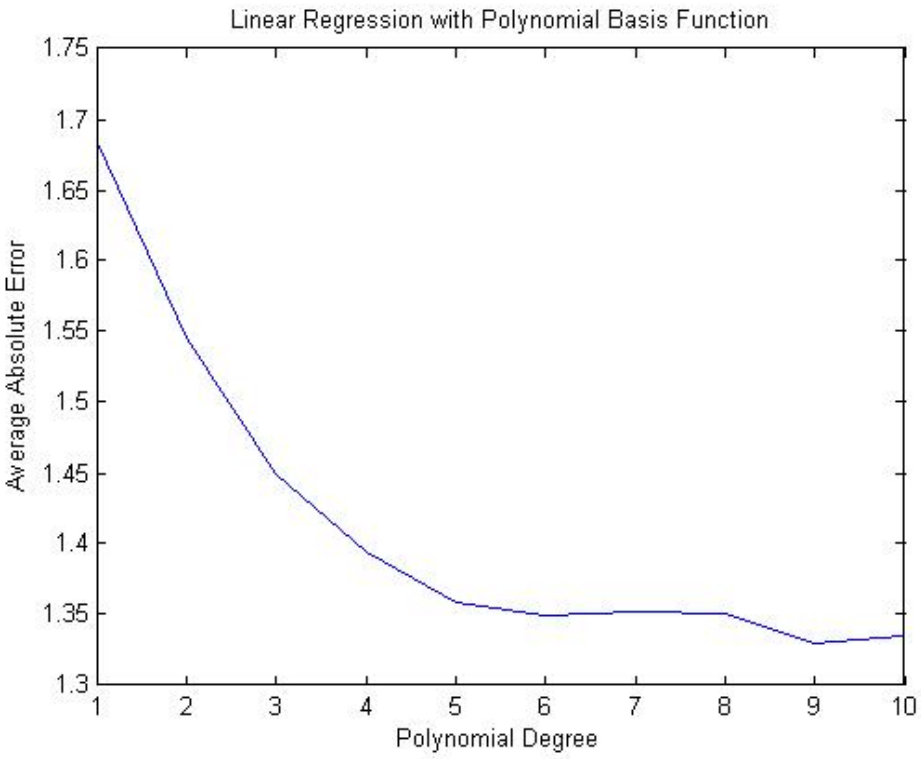
Once we are done with the task of extracting features, we use linear regression with polynomial basis functions to find out the scores of each essay.

4.1 Combined Modelling

Firstly, we trained our model using all the **8900 training documents at once**. Now, since, all the 8 sets had different scale for marking, we first normalized all the scores to interval [0-10].

For training, we varied the degree (d) of the polynomial basis function from d = 1 to d = 10 and looked at the testing error. The testing error in our case is the average of absolute difference between the actual and predicted scores. As the case with training, the testing is also done on essays from all the 8 sets at once.

Following plot shows the behavior of test error as we vary the value of the parameter d:



These are the values of the test error with varying d:

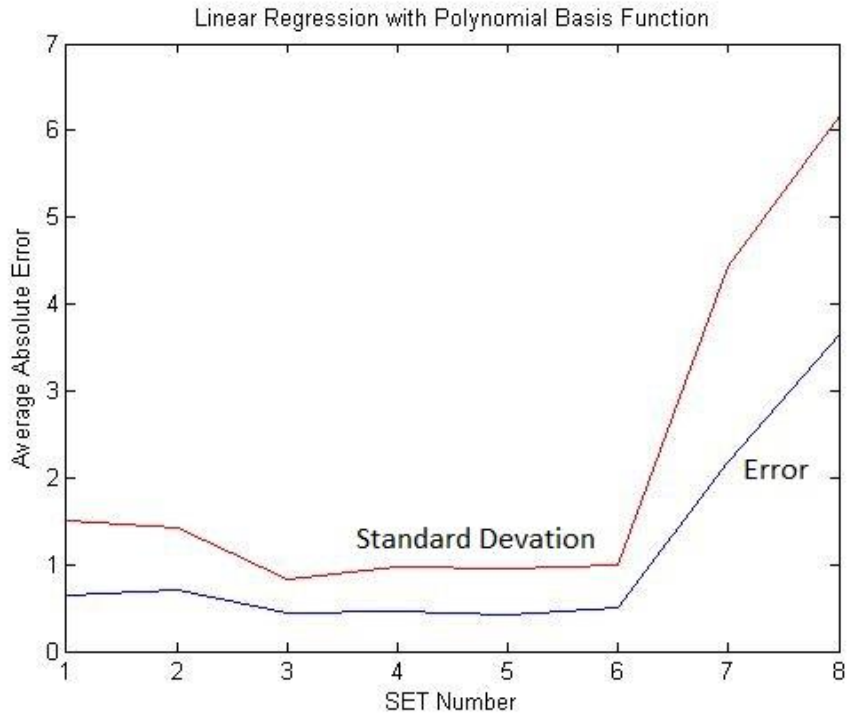
Value of d	Test Error
1	1.6854
2	1.5454
3	1.4494
4	1.3937
5	1.3588
6	1.3490
7	1.3521
8	1.3504
9	1.3296
10	1.3344

As evident from the plot and the figures, given the current setting, the best results are obtained by fitting a polynomial of degree 9 on the data. The average discrepancy between the scores generated by our model and the actual human scores is about 1.3296 marks on a scale of [0-10].

4.2 Domain-Wise Modelling

In the above setting, we trained our model on all 8 sets in one go and then test our model on essays from all sets. Thus, we could not really model the domain information well.

So, to take into account that all the 8 sets are very different, we train our model using the training data from one particular set and then test for that set. And this process is done on all of the 8 sets. So, unlike the previous case, no normalization of given scores is required. For each set, we calculate relative absolute error on the test set. The results obtained are as follows:



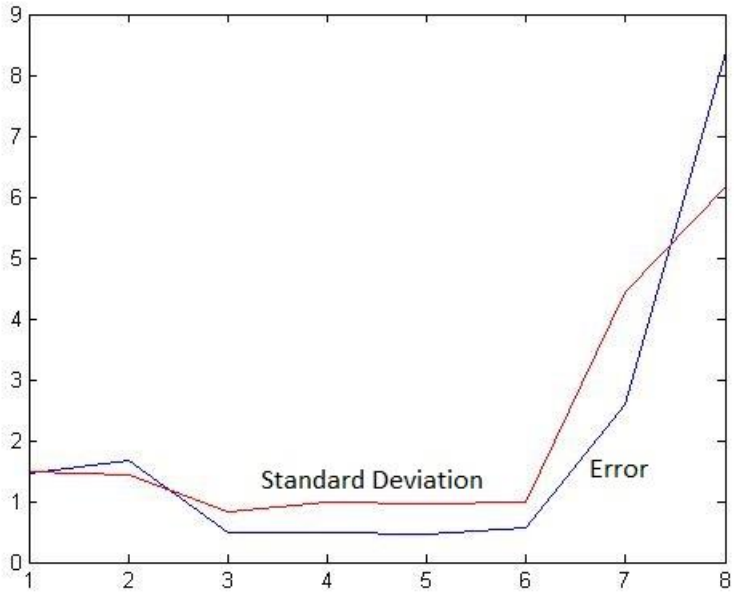
This figure shows the **average absolute error** for the set and the standard deviation of the given scores in that test set. This was plotted to ensure that the error of the proposed model is less than the **standard deviation** in the given data. The following table gives the values of the above plot:

Essay Set	Scale of Marking	Standard Deviation	Average Absolute Error
1	2-12	1.4996	0.6522
2	2-10	1.4294	0.6963
3	0-3	0.8280	0.4473
4	0-3	0.9781	0.4697
5	0-4	0.9579	0.4196
6	0-4	0.9995	0.5059
7	0-30	4.4248	2.1839
8	0-60	6.1788	3.6492

The relative trend of the error (and standard deviation) values is due to different scale of marking across different data sets.

As evident from the error values, the proposed model seems to be reasonably good and reliable.

Instead of linear regression with polynomial basis function, we also tried using Support vector regression, and got the following error (and standard deviation) plot.



Clearly, this alternative approach doesn't work and so, we continued with linear regression with polynomial basis function.

5. Conclusion and Future Scope

Automatic essay grading is a very useful machine learning application. It has been studied quite a number of times, using various techniques like latent semantic analysis etc. The current approach tries to model the language features like fluency, grammatical correctness, domain information content of the essays, and tries to fit the best polynomial in the feature space using linear regression with polynomial basis functions.

The results obtained seem quite encouraging. We achieve average absolute error to be significantly less than the standard deviation of the human scores. Across all domains, the proposed approach appears to work very well.

The future scope of the given problem can extend in various dimensions. One such area is to search and model good semantic and syntactic features. For this, various semantic parsers etc. can be used. Other area of focus can be to come up with a better tool than linear regression with polynomial basis functions like neural networks etc.

6. References

[1] "*An Overview of Current Research on Automated Essay Grading*", S. Valenti, F. Neri and A. Cucchiarelli, DIIGA, Italy, Journal of Information Technology Information, Vol. 2 2003

[2] *"Automated Essay Grading"* P. Reddy and G. Jambagi, University of Illinois, Chicago, USA, 2003

[3] Python libraries :

textmining (https://pypi.python.org/pypi/textmining)

enchant (https://pypi.python.org/pypi/pyenchant)

NLTK (http://nltk.org/)

Wordnet (http://www.wordnet.princeton.edu)