Judging a Book by its Cover: A Modern Approach

Yien Xu
yien.xu@wisc.edu

Boyang Wei
bwei9@wisc.edu

Jiongyi Cao
jcao56@wisc.edu

1. Introduction

In this project we intend to explore several deep learning techniques to understand if a book can be judged by its cover. Specifically, we want to study the best book ever data set scraped from Goodreads (https://www.goodreads.com/list/show/1.Best_Books_Ever) [3] to learn the relationship between a book cover and its popularity. We plan to qualify the popularity of a book by the average number of reviews received across years, and classify them into 5 categories based on the quantile of the review counts. We first intend to apply deep learning techniques to train a model predicting the number of reviews based on book covers, and then focus on interpreting the model by figuring out which part of the cover contributes the most to a book's popularity. Finally, we hope to generate our own "most popular book cover ever" based on the information and insights we obtain from previous analysis.

Due to the ordinal nature of our target, i.e. the labels are ranked in the order based on the number of reviews, our model would be an ordinal regression. Previously, several works have been conducted on using neural network for ordinal targets. Particularly in our project, we will utilize the recent proposed work by W. Cao, V.Mirjalili and S.Raschka [1]. Their framework, namely the Consistent Rank Logits (CORAL), addressed the consistency problem existed in previous proposed classifiers and is empirically applicable to Convolutional Neural Networks (CNN).

Understanding and explaining the model, specifically which part of the image influences the most on a book's popularity, is another important goal in our project. In the past, machine learning models usually suffered from being treated as black boxes and cannot be interpreted. Recently, a novel explanation technique, Local Interpretive Model-Agnostic Explanations (LIME) [4], has been proposed to faithfully explain any classification or regression prediction by local approximation. Particularly for image models, LIME can present patches in an image to provide qualitative understanding of the relationship between an instance and its prediction label (detailed application and anticipated result see in Section 3. Evaluation.)

Finally, we are inspired by the work of G.Surma, who created the Image Generator (DCGAN) [6] with Generative Adversarial Networks (GAN) [2]. Following the idea of G. Surma, our goal is to train a GAN that can automatically generate the most popular book covers in various genres.

2. Motivation

There is a saying, "Do not judge a book by its cover." This might be true, but what we are interested in is its literal meaning - despite the content, will a book attract more readers simply by a well-designed cover? As a reader, which book will catch your attention the most among hundred of others in a bookstore? For specific genres and topics, which book cover should be used to obtain the highest popularity score (or to attract readers the most). For example, for the book "Robinson Crusoe", should we use a portrait of Robinson, or should we use a photo of the land surrounded by the sea? Our project should find out the answer.

In addition, if there is a relationship between book covers and their popularity, we would like to explore more about the design and the potential patterns they may have. In this way, we can offer suggestions to the publisher in terms of the design of book covers. This will be regarded as a way to boost book sales and more importantly, attract more readers to read the book.

Lastly, the outcome of the model can be used as a tool to analyze which kind of images and patterns will catch human eyes. In addition to the design of the book cover, the general idea can also be implemented into other fields, such as the design of the film poster and the advertised photo of a particular brand.

3. Evaluation

Since we will categorize the popularity of a book to 5 different classes, we plan to predict which category the book will be in based on its cover from the Convolutional Neural Network (CNN). The outcome of the model will be evaluated based on the accuracy of the test set. In other words, a successful model will produce an acceptable test accuracy.

In addition, we plan to use Local Interpretive Model-Agnostic Explanations (LIME) to extract specific parts of the picture that play an essential role in determining the popularity of the book [4]. Particularly, a successful LIME

output will tell us which specific parts of the book cover are mostly deterministic to the image label and thus be the key contribution to book's popularity.

Lastly, we plan to implement Generative Adversarial Nets (GAN) to generate a book cover that is likely to have the highest popularity score [2]. We hope to generate the "hottest book cover" for different genres. For example, we hope to generate a "hottest book cover" for fictions and a "hottest book cover" for detective stories. In order to evaluate this method, we will look into pictures generated and explore realistic interpretations.

4. Resources

4.1. Dataset

Fortunately enough, Goodreads provides a list that keeps track of the best books ever, voted by the general Goodreads community [3]. A project on Kaggle further scrapes the details of each book from Goodreads and thus compiles them to a data set with 54301 books (`https://www.kaggle.com/meetnaren/goodreads-best-books`) [5]. The dataset contains general information that help identify a book, such as book titles, authors, descriptions, and genres. Moreover, it also includes useful details that are beneficial for training, such as book ratings, rating counts, and review counts. And finally, each book record comes with an image of the book's cover. However, the year of publication is missing from the Kaggle dataset, a field that is essential for normalizing review counts by year. Hence, we plan to write a script to scrape the year of publication for each book on the list from Goodreads.

4.2. Computing Resources

Python will be our primary programming language used in this project. For web scraping, we plan to use a powerful Python library named Beautiful Soup. For neural network training, we plan to use PyTorch. Neural nets will be trained either on a local desktop with a single GPU (NVIDIA's GeForce GTX 1080) or on Google Cloud. In addition to Python, we plan to use shell scripts to automate some processes so as to provide a better way for others to reproduce our results.

5. Contributions

Yien Xu plans to scrape Goodreads to get the year of publication for every book on the list. Boyang Wei plans to implement the CNN that predicts the popularity of each book. Jiongyi Cao plans to experiment on LIME to interpret CNN that Boyang trains. Yien Xu will train a GAN to generate a new book cover that demonstrates high popularity. Each group member will write up their methodology sections in the final report respectively. The rest of the write up (e.g., introduction, research on related works) will be distributed evenly among group members.

References

[1] W. Cao, V. Mirjalili, and S. Raschka. Consistent rank logits for ordinal regression with convolutional neural networks. *CoRR*, abs/1901.07884, 2019.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[3] Goodreads. Best books ever (55627 books).

[4] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[5] N. Santhanam. Goodreads' best books ever, Dec 2018.

[6] G. Surma. Image generator - drawing cartoons with generative adversarial networks, Feb 2019.