Book recommendation using clustering algorithms

Emily Behlmann

Definition

Project Overview

The goal of this project is to create an application that recommends books to an individual based on the person's taste and the book ratings of fellow readers. The project relies on the goodbooks-10k dataset, which was published under a Creative Commons license by Zygunt Zajac on the FastML website in 2017. The dataset includes about 6 million ratings of 10,000 books with the highest volume of ratings (Zajac). The project involves grouping readers based on their book tastes so I can recommend to individual readers books they haven't read that their fellow group members have rated highly.

The idea for this project came from the fact that, like many readers, I have an account on Goodreads, a website for tracking your reading, reviewing books and connecting with fellow readers. The site, which says that it has 75 million members, provides book recommendations that it states are based on 20 billion data points (Chandler). However, I've never been very interested in the books Goodreads has recommended for me. This made me wonder if I could create a better recommendation engine.

Problem Statement

To build a book recommendation engine, I used unsupervised learning algorithms to discover clusters of readers who have someone similar tastes. The process involved the following steps:

- Analyze the dataset to better understand it.
- Create a sparse matrix that includes a column for each book and a row for each reader.

- Preform data preprocessing, including dropping books and readers with the least ratings to reduce sparseness and applying truncated SVD to reduce dimensionality. In addition, separate out a test set of readers.
- Attempt k-means clustering and Gaussian mixture modeling with various k values. Select and apply the algorithm and k value that results in the best silhouette score.
- Build a function to recommend to an individual reader books his or her fellow cluster members rated highly that the individual had not yet read.
- Analyze the results. Evaluate whether readers give higher ratings to the favorite books of their cluster than they do to books chosen for them at random.

My goal was to maximize the rating score readers give to the books selected for them.

Metrics

I did not know ahead of time the "ground truth," or the number of reader clusters that existed in my dataset. Therefore, calculating the silhouette coefficient on various models and numbers of clusters was an appropriate metric for evaluation. The silhouette coefficient helps to illustrate how well points fit within their clusters based on two scores (Scikit-Learn Silhouette Score):

1. The mean distance between a sample and other points within the same cluster, and
2. The mean distance between a sample and all points in the next nearest cluster.

I evaluated multiple clustering algorithms and multiple k values to choose the one that results in the highest silhouette coefficient.

In addition, I attempted to create a metric to evaluate my application's performance in rating books. For each reader in a test set, I calculated a benchmark by picking at random 10 books the person had rated and averaging the individual's ratings for those books. Then after clustering, I found the 10 books the reader had read that his or her cluster rated the highest and calculated the reader's average score

for those books. Finally, I compared the average reader's rating for the recommended books to the reader's average rating for the randomly chosen books, anticipating that the person would give higher ratings to the cluster's favorite books.

Analysis

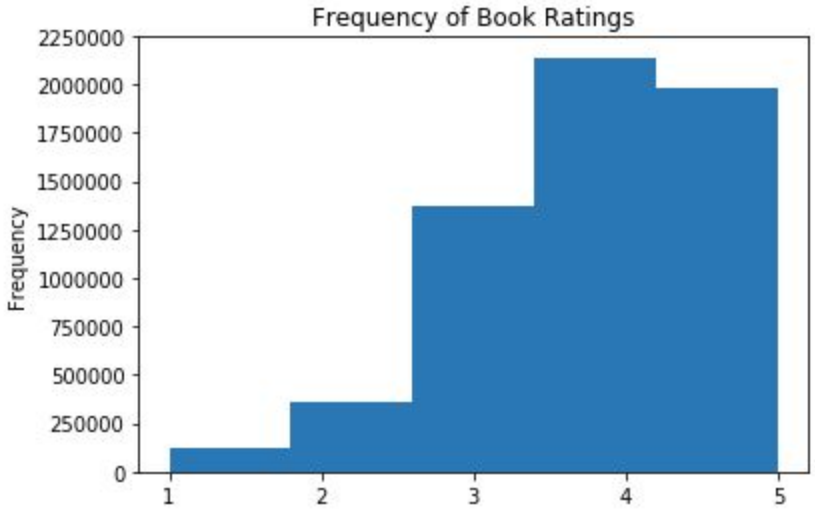
Data Exploration

My study is based on the goodbooks-10k dataset, which includes the 10,000 books that received the most ratings on a book-related website (Zajac). Overall, these books received nearly 540 million ratings. However, I focused my attention on the approximately 6 million ratings, about 1.1 percent of the total, that are associated in the dataset with 53,424 individual users. The dataset does not include user associations for the remaining 98.9 percent of ratings.

Ratings are based on a 5-point scale, though readers appeared not to take advantage of the full scale. From among the 6 million ratings associated with users, the mean rating is 3.92 and the median is 4.0. The book with the highest average rating, at 4.82, is “The Complete Calvin and Hobbes” by Bill Watterson. The book with the lowest average rating, at 2.47, is “One Night at the Call Center” by Chetan Bhagat. Among the 53,424 users in the dataset, the average user completed about 112 ratings of the books in the dataset. Many users completed 100 or more ratings, which meant I could drop users with fewer than 100 reviews and still have a significant amount of data to process.

Exploratory Visualization

To analyze the frequency of various ratings, I created a histogram. The negative-skewed histogram reinforces an idea that was already apparent when I calculated the mean (3.92) and median (4.0 ratings): Many users do not take full advantage of the 5-point scale to rate their books. Instead, many ratings are concentrated at the upper end of the scale.



The skewed nature of the ratings could have a variety of causes. Some readers may already know what they like well enough that they select books they end up enjoying. Alternatively, readers may feel uncomfortable giving negative reviews. Finally, they may have given relatively high scores to books they only somewhat enjoyed, and now they don't have much higher to go. Regardless, it's possible this phenomenon will have a negative impact on my ability to cluster readers, because it's hard to discern a significant difference between a book a reader loved and one he or thought was mediocre.

Algorithms and Techniques

This project depends on the ability to associate users into groups that have similar tastes. Because these groups were unknown to me — the datasest is unlabeled — it was necessary to use unsupervised learning techniques to discover them. I decided to try two clustering algorithms within this domain, then select the algorithm and hyperparameters that provided the best silhouette coefficient:

- K-means clustering – This method is relatively fast and easy to apply, so it's a good default (Seif). It involves randomly placing cluster center points, then associating each data point with its closest center. Then, the center points are adjusted based on the mean of all the vectors in the group (Seif). This process is repeated until the clusters are relatively stable.

- Gaussian mixture modeling – This method is more suitable than k-means for non-circular clusters (Seif). Because I don't know the shape of my clusters, it's an appropriate alternative to k-means. Gaussian mixture modeling is “a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters” (Gaussian Mixture Models). It determines the probability that each data point belongs to a particular cluster.

Benchmark

The objective of the book recommendation engine is to recommend books a person hasn't read. However, evaluating the quality of these recommendations is difficult without the opportunity to wait for readers to read the book and return with feedback. Therefore, I had to take a different approach when comparing the quality of my recommendations to those of a benchmark model.

The simplest way to provide books recommendations would be to select books at random. This is the idea that informed my benchmark. To calculate my benchmark metric, I first separated out 20 percent of users and their book ratings into a test set. For each user in the test set, I randomly selected 10 books the user had rated, and I calculated the reader’s mean score for those books. Then, I averaged all these users’ means together to find an overall mean of about 3.89.

My goal was to build a recommendation engine that could recommend 10 books to test users that, on average, the users would rate at better than the 3.89 benchmark.

Methodology

Data preprocessing

My initial dataset included two separate dataframes — one containing users’ ratings and one containing books. The first step in the data preprocessing process was to combine these two dataframes

into a single matrix of ratings, in which each row represented a user and each column represented a book. The matrix was very sparse, because each reader has only rated a small fraction of the books in the dataset. To help reduce the sparseness and make the data easier to process, I dropped books that received fewer than 300 ratings. Then, I dropped users who had given fewer than 100 ratings to the remaining books. I ended up with a matrix with 4,167 books (about 41% of the original) and 21,093 users (about 39% of the original).

At this point in the process, my dataset included 4,167 features, which would have made it difficult to process. I expected that many of the features may have correlations with one another — many books may fall in the same genre, for example — so I applied dimensionality-reducing techniques to identify a smaller number of composite features. Specifically, I used truncated SVD, which is a method of reducing dimensionality in sparse datasets (Scikit-Learn Truncated SVD). I determined using the explained variance ratio that if I reduced my data to 200 features, this would explain about 45 percent of the variance in the data. This allowed me to substantially reduce the number of features in the dataset while still retaining nearly half of the variance.

Finally, I separated 20 percent of users and their book ratings into a test set. For these users, I calculated my benchmark of a 3.89 mean rating on 10 randomly selected books per user.

Implementation

After preprocessing, I had a training data set that included 200 composite features and 16,874 users ready for clustering. I used the Scikit-Learn Python library to apply k-means clustering and Gaussian mixture modeling to the data. For each algorithm, I tested numerous different cluster counts (known as a k value in k-means clustering an a component count in Gaussian mixture modeling), then calculated and recorded the silhouette coefficient.

I received the highest silhouette coefficient, of about 0.0447, using k-means clustering and a k value of 7. Therefore, I chose to fit my data to this model. Once complication arose during this process: In the dataframe I used for training, the indices had been reset during truncated SVD preprocessing, which meant there was no way to associate a record in the training set with a user ID. I resolved this by saving a copy of the indices before truncated SVD, which represented user IDs, and applying these indices to the cluster predictions. In this way, I was able to identify the assigned cluster number for each user profile.

Finally, I created a book recommender function based on my clustering model. To make a recommendation, the function fits a reader into a cluster based on his or her existing book ratings. Then, it proceeds down a list of books to which the cluster has given the highest mean rating until it finds one the reader has not yet read, and it recommends that book.

Refinement

The primary method I used for refining my results was the technique described above of testing various cluster counts and recording the silhouette coefficient of each. For each algorithm (k-means clustering and Gaussian mixture modeling) I attempted the following values: 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 25, 50, 100. For k-means clustering, the lowest silhouette coefficient was a score of 0.0103 for a k value of 100, while the highest was a score of 0.0447 for a k value of 7. For Gaussian mixture modeling, the lowest silhouette coefficient was a score of 0.0069, which occurred for both 13 and 14 components. The highest was a score of 0.0309 with 2 components. Therefore, I chose to cluster my users using k-means clustering with a k value of 7.

Results

Model Evaluation and Validation

It is difficult to say with certainty that a clustering model is "good" or that book recommendations are of high quality. However, there are a few ways to assess this model and how well it performed at identifying clusters of readers.

First is to analyze the silhouette coefficient. Although I chose the model and k value that maximized the silhouette coefficient, the resulting score was still quite low — 0.0447. According to Scikit-Learn, which is the library I used to determine silhouette coefficients, the value can range from a best of 1 to a worst of -1. Therefore, any positive number should indicate that, at least to some degree, points are fit within their appropriate clusters. However, the Scikit-Learn documentation also notes that values close to 0 indicate overlapping clusters (Scikit-Learn Silhouette Score). Given that my model earned a silhouette score of less than 0.1, it seems likely that there is overlap among my clusters, which could lead to misclassified data points.

Another, more subjective means of evaluating the model is to study the individual clusters it identified. I found seven clusters:

Cluster 0

title	
Guess How Much I Love You	4.833333
Harry Potter Boxset (Harry Potter, #1-7)	4.811966
The Boys in the Boat: Nine Americans and Their Epic Quest for Gold at the 1936 Berlin Olympics	4.800000
Harry Potter and the Deathly Hallows (Harry Potter, #7)	4.762994
A Court of Mist and Fury (A Court of Thorns and Roses, #2)	4.751773
The Harry Potter Collection 1-4 (Harry Potter, #1-4)	4.750000
Words of Radiance (The Stormlight Archive, #2)	4.736842
Schindler's List	4.727273
Acheron (Dark-Hunter #14)	4.718750
Talking to Dragons (Enchanted Forest Chronicles, #4)	4.714286

This cluster is at least somewhat focused on children's books ("Guess How Much I Love You", "Harry Potter") and books related to history ("The Boys in the Boat", "Schindler's List").

Cluster 1

title	
It's a Magical World: A Calvin and Hobbes Collection	4.823529
The Harry Potter Collection 1-4 (Harry Potter, #1-4)	4.820513
The Kindly Ones (The Sandman #9)	4.785714
Fool's Fate (Tawny Man, #3)	4.785714
Brief Lives (The Sandman #7)	4.785714
Fables and Reflections (The Sandman #6)	4.774194
Harry Potter Boxset (Harry Potter, #1-7)	4.771044
Nothing to Envy: Ordinary Lives in North Korea	4.769231
Worlds' End (The Sandman #8)	4.766667
A Game of You (The Sandman #5)	4.741935

dtype: float64

This cluster appears to be somewhat focused on comics/graphic novels, including "Calvin and Hobbes" and the Sandman series.

Cluster 2

title	
The Complete Calvin and Hobbes	4.925000
The Calvin and Hobbes Tenth Anniversary Book	4.868421
Saga, Vol. 2 (Saga, #2)	4.800000
It's a Magical World: A Calvin and Hobbes Collection	4.739130
Saga, Vol. 1 (Saga, #1)	4.714286
Words of Radiance (The Stormlight Archive, #2)	4.707657
Calvin and Hobbes	4.706522
The Essential Calvin and Hobbes: A Calvin and Hobbes Treasury	4.688312
The Kindly Ones (The Sandman #9)	4.670455
The Stand: Soul Survivors	4.642857

dtype: float64

Members of this cluster appear to be big fans of Calvin and Hobbes, though these books are also popular across the entire dataset.

Cluster 3

title	
It's a Magical World: A Calvin and Hobbes Collection	4.868421
The Complete Calvin and Hobbes	4.790323
Saga, Vol. 3 (Saga, #3)	4.785714
Saga, Vol. 2 (Saga, #2)	4.769231
The Calvin and Hobbes Tenth Anniversary Book	4.750000
A Game of Thrones: The Graphic Novel, Vol. 1	4.733333
Shadowfever (Fever, #5)	4.727273
The Constitution of the United States of America	4.725490
The Essential Calvin and Hobbes: A Calvin and Hobbes Treasury	4.721429
The Wake (The Sandman #10)	4.696429

dtype: float64

Similar to people in other clusters, members of this cluster like Calvin and Hobbes and Sandman books. They also like Saga, another graphic novel series.

Cluster 4

title	
The Hitchhiker's Guide to the Galaxy: A Trilogy in Four Parts	4.833333
The Complete Calvin and Hobbes	4.833333
The Case for Christ	4.800000
Dreamfever (Fever, #4)	4.800000
Faefever (Fever, #3)	4.800000
Worlds' End (The Sandman #8)	4.777778
The Wake (The Sandman #10)	4.769231
The Kindly Ones (The Sandman #9)	4.769231
The Far Side Gallery	4.764706
Brief Lives (The Sandman #7)	4.750000

Readers in cluster 4 appear to be interested in fantasy books such as "The Hitchhiker's Guide to the Galaxy" and the Fever series.

Cluster 5

title	
The Book of Mormon: Another Testament of Jesus Christ	5.000000
Holy Bible: New International Version	4.857143
Words of Radiance (The Stormlight Archive, #2)	4.800000
Fruits Basket, Vol. 1	4.800000
The Cat in the Hat and Other Dr. Seuss Favorites	4.789474
Complete Poems, 1904-1962	4.750000
The Paper Bag Princess	4.750000
Saga, Vol. 2 (Saga, #2)	4.750000
The Harry Potter Collection 1-4 (Harry Potter, #1-4)	4.733333
The Complete Poems of Emily Dickinson	4.714286

Several of the favorite books of cluster 5 are religious, including the "Book of Mormon" and the "Holy Bible". Members of this cluster also appear to like poetry.

Cluster 6

title	
The Complete Calvin and Hobbes	4.818182
Bloodfever (Fever, #2)	4.800000
The Calvin and Hobbes Tenth Anniversary Book	4.730769
It's a Magical World: A Calvin and Hobbes Collection	4.729730
Beyond the Highland Mist (Highlander, #1)	4.727273
The Complete Novels	4.714286
The Harry Potter Collection 1-4 (Harry Potter, #1-4)	4.708333
The Secret (Highlands' Lairds #1)	4.666667
A Voice in the Wind (Mark of the Lion, #1)	4.655172
Going Postal (Discworld, #33; Moist von Lipwig, #1)	4.650000

Members of Cluster 6 like "Calvin and Hobbes" and "Harry Potter" as well, but they also appear to be interested in romance books such as "Beyond the Highland Mist" and "The Secret."

A subjective review of the clusters' favorite books leads to similar conclusions to the silhouette coefficient analysis. In both cases, the model shows some success; the silhouette score is greater than 0 and there appear to be some patterns within the favorite books of a cluster, such as a preference for a particular genre. However, both analyses also indicate an overlap between clusters. For example, books that are highly rated across the entire dataset, such as "The Complete Calvin and Hobbes" and the Harry Potter series, recur as favorite books among several clusters.

A final means of evaluation would be to provide book recommendations and to ask readers to evaluate the quality of the recommendations. I created a function to recommend books to readers within my test dataset, and this function recommended books that are highly rated in general. However, I was unable during this project to share these recommendations with real readers and solicit their feedback.

Based on the analysis above, I would conclude that clustering was somewhat effective, but without further evaluation, it's difficult to trust this model to provide consistently high-quality book recommendations customized to each cluster. It's very likely that clusters overlap, which means it's unlikely I've succeeded in identifying distinct and clearly defined reader profiles. Further, I've been unable to incorporate feedback from real readers to improve the model.

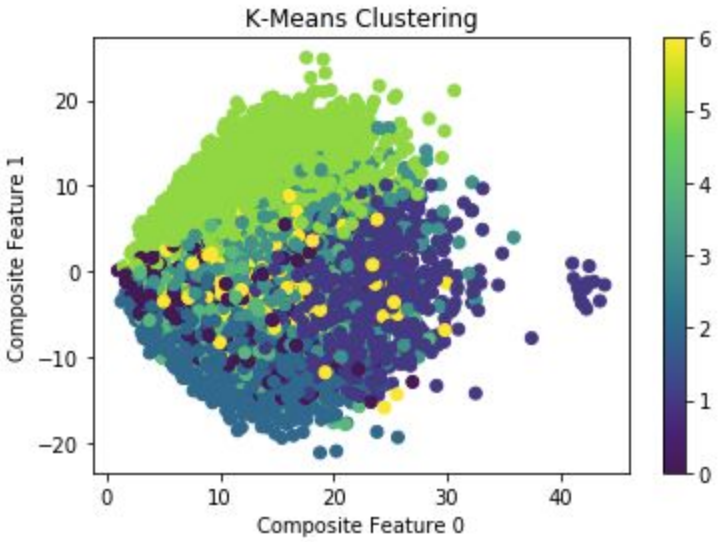
Justification

My clustering model appears to do a better job of recommending books than my benchmark model. In the benchmark, for each reader in a test set, I "recommended" 10 random books the individual had already read and obtained the person's mean rating for those books. Then, I averaged all these users' means together to get an overall mean of about 3.89 on a 5-point scale. To obtain a similar metric for my model, I clustered each user in the test set, then "recommended" the 10 favorite books of the cluster that the person had already read. Again, I found the mean of the user's ratings for these books, then averaged all the means together. The resulting overall mean was 4.36, which, in the context of a 5-star rating scale, is nearly a half star better. Thus, my model outperforms recommending books at random.

However, I can't say for certain that the clustering itself is what caused my model to outperform the benchmark. My function recommends books that are highly rated by a reader's cluster, but in many cases, such as with "The Complete Calvin and Hobbes," these books are also highly rated across the entire dataset. Therefore, it's likely that the simple act of recommending relatively highly rated books — regardless of cluster — will lead to better results than recommending books at random.

Conclusion

Free-Form Visualization



The dataset I used for clustering included 200 separate composite features, making it difficult to visualize. Therefore, I selected the 2 first — and most significant — features highlight in a scatterplot visualization. In the scatterplot, a few distinct clusters are visible — especially cluster 5, depicted in light green. This indicates that the clustering model was effective in grouping users based on their values for composite features 0 and 1. However, the scatterplot also provides further evidence for my previous conclusion that there is significant overlap between clusters. For example, cluster 6, depicted in yellow, appears to capture points over a wide area that might also reasonably fit within other groups.

Reflection

This project involved analyzing a large set of user book rating data, preprocessing it to reduce sparseness and dimensionality, applying clustering algorithms with various hyperparameters to find the model that maximized the silhouette coefficient, and finally, using the model to cluster readers and

recommend books for them. I believe that overall, the steps I took to preprocess the data and create a clustering model were appropriate for the problem at hand. However, there were some challenges that made it difficult to create a truly effective recommendation engine.

Many weaknesses were in the data itself. For example, each reader may interpret the 5-star rating scale differently. Some readers might consider 3 stars to be a solid rating for a book they liked, while others might believe 3 stars is a relatively poor rating. The ratings are not factual information about the quality of books, and they may be inconsistent. Another interesting characteristic about the dataset is the fact that readers tend to concentrate their ratings in the upper half of the scale. This means there's little separation in scores between books readers love and merely like. Also challenging is the fact that a few popular books, such as "The Complete Calvin and Hobbes," received high ratings from a large number of people across clusters, so they tended to dominate recommendations.

Other challenges arose because of the nature of the process. I was working with a sparse dataset, which is difficult to cluster, so I decided to reduce sparseness and dimensionality. However, it was difficult to know how far to go with this preprocessing work and whether I was sacrificing too much of the data in performing dimensionality reduction. Another concern is the fact that to truly evaluate a recommendation engine requires asking real readers to assess its recommendations — something I was unable to do during the course of the project.

Improvement

While I believe my book recommendation function is a good start based on sound machine learning concepts, there are many ways it could be improved. Some possibilities include:

- Remove from the dataset some the most popular and highly rated books. Books such as "The Complete Calvin and Hobbes" were so popular that they were among the favorites for multiple

clusters. By identifying and removing books like this, I might be able to discover more distinct clusters.

- Experiment with different numbers of composite features to determine if it's possible to reduce cluster overlap.
- Seek out a different dataset that has more variation in book ratings, such as one where users rate books on a 10-point scale or where users are allowed to give a book half stars. More granularity in ratings might lead to more distinct clusters.

Perhaps the best way to improve the model, however, would be to use it in the real world by making recommendations to actual readers, then asking those readers for feedback. If I could incorporate readers' feedback into the model, I could, over time, produce better and better recommendations.

Works Cited

Chandler, O. About Goodreads. Retrieved from https://www.goodreads.com

Gaussian Mixture Models. (n.d.). Retrieved from http://scikit-learn.org/stable/modules/mixture.html

Scikit-Learn Silhouette Score. (n.d.). Retrieved from http://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

Scikit-Learn Truncated SVD. (n.d.). Retrieved from http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html

Seif, G. (2018, February 5). The 5 Clustering Algorithms Data Scientists Need to Know. Retrieved from https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68

Zajac, Z. (2017, November 29). Goodbooks-10k: A new dataset for book recommendations. Retrieved from: http://fastml.com/goodbooks-10k