Movie Recommendation Using Naive Bayes Text Classification

Dylan McCardle | Corey Myers | Evan Sheehan

Goal

- Can we make a movie recommendation engine using Naive Bayes text classification?
- What will our classes be?
- How should we filter text?
- Does a recommendation engine based on NB text classification provide interesting, novel, or accurate results?

Naive Bayes Text Classification

- We will use Naive Bayes as our algorithm for determining which movies to recommend
- Naive Bayes fits our problem scope as we need to classify a movie into one of two classes based on a bag of words, that is, the movie’s reviews
- Our two classes will be recommended and not recommended
- Our two vocabularies for these classes will be populated based on the user reviews of movies that the user provides that they liked and disliked

Gathering Data

- Ask the user for *n* movies they like, and *n* movies they dislike.
  - Possibly 3-5 movies.
- IMDB with Selenium WebDriver
  - Allows for ‘realistic’ navigation of website.
  - Gathering data ends up being very slow.
- Get vocab from every review
  - Remove stop words (prepositions, articles, etc.)
  - Remove punctuation
  - Lowercase
  - Stemming (only root words)

Further Data

- Use movie-map.com to populate initial list of similar movies.
  - The results are more anecdotally accurate than IMDB, and it’s easier to parse.
- Populate similar movies list with *n* movies from movie-map.com for each movie, then n^2 total movies, up to a given boundary.
- IMDB scraper will be applied to this list of movies.
  - Still an issue of time. Maybe pre-build a database?
- It is likely that the user has seen many of these new movies before.
  - We can use this for error analysis, asking the user for further input.
  - These new movies will be used to expand the like/dislike vocabulary.
  - Possible third class: neutral/no strong opinion?

Applying the Algorithm

- Once we have gathered the data from the user and constructed our vocabulary, we apply the algorithm on a new movie
- We will be using naive bayes with laplace +1 smoothing and logarithmic scaling so as not to produce decimal multiplication underflow errors
- This will essentially look like the following:

```
p_like = (sum of wordcount in all liked movies)/(sum of total word count in all movies);
p_good = log(p_like);
for each word in movie_vocab{
        find movie_vocab[i];
        numerator = (number of times movie_vocab[i] appears in like_vocab) + 1;
        denominator = like_vocab.size + total number of words;
        add_this = numerator/denominator;
        p_good = p_good + log(add_this);
}
//repeat this exact process for p_dislike and save the resulting p_good and p_bad as a variable that can be linked to the movie title
```

Output

- We will output the top 5 movies that the algorithm finds are recommended to the user, based on naive bayes output scores and how much they skew towards the movies the user likes, compared to how much they skew towards movies the user disliked.
- The top 5 largest ratios of like_score to dislike_score will be displayed

Process Example

- Sample input of 3 liked movies:
  - Avengers: Endgame, The Terminator, The Matrix
- Sample input of 3 dislikes movies:
  - It Chapter 2, The Shining, Doctor Sleep
- Sample slice of vocab:
  - “Action” appeared 134 times in like list; 6 times in dislike
  - “Scary” appeared 70 times in dislike list; 6 times in like list
- Parse movie-map.com to find similar movies
  - The Hobbit, The Amazing Spiderman, Thor: Ragnarok (and *n*-3 less similar results)
- Web parser divides new movie reviews into like and dislike scores
  - Score = comparison to like vocab / comparison to dislike vocab
- Sample output:
  - The Amazing Spiderman, Thor: Ragnarok, The Hobbit (based on scores)

Future additions

- Once we have our framework of algorithm down we can expand to other classifications
    - Prediction of whether a movie is of a certain rating based on reviews
    - Prediction of whether a movie is a certain genre based on reviews
- Increase efficiency in navigating websites and gathering vocab
- Add an error rate
    - Ask the user if they’ve seen other movies and if they liked or disliked them