Gaussian Process Regression for Movie Recommendations

Group 7
National University of Singapore

Abstract

In recent years, online streaming and video on-demand services have grown exponentially. The online nature of such services necessarily multiplies the number of products that can be shown to the consumer. Consequently, there is a need for such services to determine a selection which should be recommended. In order to produce accurate predictions, there is a need to consider users' preferences, item profiles and correlations between items and users. In this paper, we attempt to apply Gaussian Process (GP) regression on movie ratings to generate recommendations.

1 Introduction

Today, it is easy to obtain online information pertaining to a movie and its critics through a search on Google, IMDB, Rotten Tomatoes, etc. Hence, modern day movie watchers would often research online about a movie before buying tickets in their local cinema or paying for online streams on movie streaming platforms such as Netflix, Amazon Video, HBO, or Hulu.

The rise of on-demand video services requires an accurate movie recommendation system in order to help users purchase the movies they like. As movie-watching shifts into the online world, consumers are faced with more choices as compared to the traditional way of browsing limited movies available in physical dvd rental stores or the theatre. Within the online ecosystem, services like Netflix are faced with the double-edged sword of being able to provide virtually any movie in existence to the consumer. Letting the consumer conduct an exhaustive search on this massive search space is certainly a computational burden that we would like to avoid.

2 Motivating Application

Moreover, motivating reasons such as word of the mouth, reviews and advertising are insufficient in the context of digitized movies. To cater to the consumer who faces the paradox of choice, there is a need for quality recommender systems which can enable movie watchers to discover movies they love on their own.

Our application firstly exploits the predictive mean and uncertainty provided by the Gaussian process model. In particular, it is important for our application to obtain a ranking of movie that we should recommend. The Gaussian process model is particularly well-suited in this context as compared to traditional regression models. This is because our selection can be made on the basis of the predictive uncertainty that the Gaussian process model usefully provides. For example, we might obtain similarly high ratings for 2 movies, A and B. If the predictive uncertainty given by the Gaussian process model is such that $\sigma_A^2 > \sigma_B^2$, then we are more likely to recommend movie B, or rank it higher on our recommendation list. Additionally, it is possible to use multiple models and weight our predictions inversely proportional to the variance. As a consequence, users will benefit by deriving greater enjoyment.

Moreover, movie preference is highly personal and complex to model. Director, script, cast, score, visual effects and genres are just of some obvious features of a movie. To give a trivial example, an individual might not like mainstream A-listers action movies such as Fast & Furious, Bourne or James Bond, but he may like Marvel action movies because he likes superhero comics. Even then, he might not like the newer Spiderman movies because of the main lead. He might love DC Comic Batman trilogy directed by Christopher Nolan but still be disappointed by Batman v Superman: Dawn of Justice. The usage of Gaussian process models in this context is particularly helpful in enabling us to exploit the correlations between users with similar preferences. Since users with similar tastes are likely to give similar ratings to a specific movie, we can make use of these correlations to recommend movies.

It is clear that modelling human behaviour and preference is complex. Using a parametric model could unintentionally constrain our model. Since the Gaussian process model is a distribution over an infinite number of functions, we are better able to model the full spectrum of highly variable human preferences by exploiting the non-parametric characteristic of Gaussian processes.

Finally, let us suppose that our Gaussian process model can generate relatively accurate recommendations. With increased user satisfaction and hence higher utilisation of the system, we are likely to obtain even more data from new ratings. This will allow us to further exploit the correlations

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

among user preferences, as well as the Bayesian nature of Gaussian process models by updating the model with new data.

In conclusion, to tackle the problem of movie recommendation, we propose a Gaussian process (GP) model. Given a list of movies that an individual has not rated before, a GP model will compute a predictive mean rating that the individual is likely to give and a predictive uncertainty which gives us insight on how accurate our prediction is. These can then be made use of to generate movie recommendations.

3 Technical Approach

A Gaussian process is a collection of random variables, any finite subset of which have a multivariate Gaussian distribution. It is completely specified by a mean function $\mu(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$. For a real process $f(\mathbf{x})$:

$$\begin{aligned} \mu(\mathbf{x}) &= E[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}') &= E[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}')] \end{aligned}$$

The GP can then be denoted as:

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

We assume that our data $\mathcal{D} = \{(x_1, y_1), \dots, (x_i, y_i)\}$ are such that y_i are noisy observations originating from a GP-distributed random function $f(\mathbf{x}_i)$ such that:

$$\begin{aligned} y_i &= f(\mathbf{x}_i) + \epsilon_i \\ \epsilon_i &\sim \mathcal{N}(0, \sigma_n^2) \end{aligned}$$

Given $\mathbf{y} = [y_1 y_2 \dots y_i]^\top$, suppose we have a new input $\mathbf{x}_*$ for which we would like to obtain a prediction for. In other words, we would like to infer $p(f_* \mid \mathbf{y})$. Then, the predictive mean and variance from the GP can be given by:

$$\begin{aligned} E[f_* \mid \mathbf{y}] &= \mu(\mathbf{x}_*) + \mathbf{k}_*^\top(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}(\mathbf{y} - \boldsymbol{\mu}) \\ V[f_* \mid y] &= k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{k}_* \end{aligned}$$

Problem Definition

Traditionally, there are 2 approaches to implementing recommender systems:

1. Content-based systems make recommendations based on item attributes. For example, a user who tends to watch many movies of the horror genre will be recommended a movie from that genre.
2. Collaborative filtering systems analyse the similarities between users and/or items to make recommendations. Users will be recommended items that are preferred by users with similar tastes.

The task is thus rating prediction, given item and user attributes. In particular, suppose we are given a set of movies, $\mathcal{M}$, and a set of users, $\mathcal{U}$. We formulate an input matrix X based on 2 types of models, per-user and per-movie, to predict user ratings $y = [y_1, y_2, \dots, y_n]^\top$, where X is given by

$$X = \begin{bmatrix} x_1(1) & x_2(1) & \cdots & x_d(1) \\ \vdots & & \ddots & \vdots \\ x_1(N) & x_2(N) & \cdots & x_d(N) \end{bmatrix}$$

Model Definition

In this paper, we assume that user ratings follow a Gaussian distribution. We formulate two different types of models:

1. **Per-user:** The per-user model can be viewed as follows – imagine the entire universe of movies, $\mathcal{M}$ – the known ratings by this particular user consist of a subset of movies, $\mathcal{M}_i$. By utilising the item profiles of these $|\mathcal{M}_i|$ movies, the Gaussian process predicts the ratings for the unknown subset of movies, $\mathcal{M}_u$, where $\mathcal{M}_i \cup \mathcal{M}_u = \mathcal{M}$.
2. **Per-movie:** The per-movie can be viewed using a similar analogy, but flipping the roles of users and items. Given the user profiles of all known user ratings for a specific movie, we predict the ratings that the remaining users are likely to give, based on the correlations between the user profiles.

Choosing Model Parameters

The GP function is mainly characterized by its covariance function after normalizing the data to attain a mean of 0. The covariance function produces a covariance matrix which is utilised by the Gaussian process model for inference. Our choice of kernel for both experimental models, per user and per movie, was the radial basis function(RBF) kernel

$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp(-\frac{1}{2}\sum_{i=1}^{d}\frac{(x_i - x_i')^2}{\ell_i^2})$$

where the hyperparameters σ^2 is the variance, ℓ is the lengthscale, d is the dimension of the input vector x_i.

The choice of the rbf kernel was due to its ability to produce similar predictive ratings for two inputs with similar features based on their distance. If two users have contrasting preferences, it is highly unlikely that both will assign the same rating to their opposing's favorite movie.

4 Experimental Setup

Dataset

Our primary dataset is the MovieLens 1M Dataset which is a stable benchmark dataset widely used for evaluating recommender systems. The primary dataset contains around 1,000,000 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens[1] in 2000. Besides ratings, the following attributes were extracted from this dataset:

movie_id movie identification number

title movie title

[1] MovieLens is a movie recommender system created by GroupLens Research.

genres a list of genres that the movie belongs to

We also utilised user demographic information such as age, gender and occupation available from the dataset.

To supplement this dataset, we combined it with the MovieLens+IMDb/Rotten Tomatoes Dataset released by HetRec 2011. Our motivation for doing so was to obtain movie critic ratings from Rotten Tomatoes. As the project progressed, we also extracted the relevant directors and actors for feature engineering purposes. The notable features extracted from this dataset include:

rtAllCriticsRating average critics' rating for the movie

rtAudienceRating average audience rating for the movie

rtCriticsNumRatings number of critics' ratings for the movie

rtAudienceNumRatings number of audience ratings for the movie

Procedure

In our experiment, the ratings and movie data were merged together to produce a dataset with both numerical and categorical features. We built a simple model based solely on numerical features before adding new features as well as experimenting with different kernels. In order to test the performance of our models, the data was also split randomly into train and test sets[2].

The basic model consists of the following features: user age, user gender, movie release year, critics and audience rating from Rotten Tomatoes.

Two advanced models are also proposed to include genres of a movie into the feature set. However, to do so, we have to preprocess the genre information into numerical values.

Feature Engineering

From our dataset, we were only able to obtain 3 demographic features: age, gender and occupation. However, the movie dataset contains a much richer set of features, including title and genres. We were also able to obtain the directors and actors by merging the original dataset with the supplementary dataset from HetRec. To utilise these features so that we could input them into the Gaussian process, we first transform them into numerical features.

One approach to encode genre was to use one-hot encoding. To achieve this, we created new columns for each of the genres. For each movie, the value is 1 if the movie belongs to that particular genre, 0 otherwise. However, since there are 20 movie genres in total, we sought to reduce the number of dimensions for the input matrix.

Through the use of Word2vec, we could transform the remaining text data into useful numerical features and reduce the dimensions of the input. Word2vec enables us to create word embeddings, or vector representations of words with a given corpus. Similar words are constructed such that they are close to each other within the vector space. We thus extracted the textual data from our datasets and processed them into a corpus suitable for training a Word2vec model on. The Word2vec model is then queried for vectors, $x = [x_1, \cdots, x_k]^\top$ representing each film. Additionally, we were able to obtain vectors representing each distinct genre present in the dataset. In training the model, it was found that a value of $k = 8$ produced reasonable results from a test of similarity between selected films and genres. The following table gives some of the cosine similarites between genre vectors most similar to each other:

Genre	Genre most similar to	Similarity
Drama	Crime	0.9132
Thriller	Mystery	0.9834
Animation	Children	0.9621
Sci-fi	Fantasy	0.8729

Another way is to group genres into buckets and compute a probability for the genres of a movie given the buckets. Choosing from the top popular genres, we identify 9 buckets: drama, comedy, crime, action, thriller, horror, fantasy, family and animation. This probabilistic approach allows us to compare the distance between genres and buckets. It also reduces the dimension of the input vector significantly.

5 Experimental Evaluation

As mentioned, our approach involves taking two main perspectives on the data, per-user and per-movie. We thus train a GP model for each user and each movie using the processed data. The Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for the models trained are given as follows:

Type	Features	Kernel	MAE	RMSE
Per-Movie	Basic	RBF	0.7823	0.9785
	Basic	Cosine	0.7823	0.9786
Per-User	Basic	RBF	0.8128	1.019
	Word2vec Genres		0.8210	1.026
	Probabilistic		0.8204	1.027
	Word2vec Movies		0.8278	1.033

6 References

Note: max 6 pages

[2] A 70-30 split was used.