Movie Recommendation based on Collaborative Topic Modeling

Abhishek Bhowmick
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
abhowmi1@andrew.cmu.edu

Udbhav Prasad
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
udbhavp@andrew.cmu.edu

Satwik Kottur
Department of Electrical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
skottur@andrew.cmu.edu

Abstract

Traditional collaborative filtering relies on reviews provided by viewers in the movie watching community to make recommendations to the user. In this project, we attempt to combine this approach with probabilistic topic modeling techniques to make recommendations that consist not only of movies that are popular in the community, but also those that are similar in content to movies that the user has enjoyed in the past.

1 Introduction

Recommender systems are an important technology for TV/movie streaming services like Netflix, HBO, audio/music streaming sites like Spotify, Pandora, news article feeds like Pulse, online retailers such as Amazon, Walmart etc. Indeed, any service provider or content management system that has large quantities of information (or the ability to extract such information) such as usage patterns, browsing and click history, natural text descriptions etc can and should make use of recommendation methods to help find items of interest. Among various information sources, data in the form of natural text is a particularly rich and expressive source of information, however it is highly unstructured in general. Topic models are used to extract latent structures from large volumes of unlabeled text, that can be used for analysis of contents and in turn, aid end goals such as making recommendations. In particular, textual information such as movie plot summaries can be very helpful to improve the prediction performance of traditional movie recommendation based on collaborative filtering. In the remaining part of this report, we limit ourselves to the study of how topic modeling of large text corpora can help in the task of movie recommendation, however most of the discussion/analysis can be applied to other domains.

1.1 Collaborative Filtering and its shortcoming

Traditional collaborative filtering makes use of interactions between users and items. They may be broadly classified into two categories - neighbourhood methods and latent factor models. Neighbourhood models explicitly capture relationships between items (or users) and predict a user's liking for a particular item based on ratings of neighbouring items by the same user. The other approach, latent factor models, directly characterize both users and items by latent factors. We focus on factor based models as they are more accurate than neighbourhood based methods. However, all collaborative filtering methods suffer from the 'cold start' problem, that is they are unable to recommend movies in the absence of rating patterns. In fact, in the domain of music, it has been observed [**?**] that the distribution of available rating information for music artists has a very long tail, meaning that most of the music items have little rating data available. We believe the same is true of movies as well and hence, would like to be able to recommend movies that are in this long tail.

1.2 Content Based Recommendation

Content-based recommendation addresses the 'cold start' problem associated with collaborative filtering, where certain items do not have any rating information and hence the corresponding item vectors consist of all zeroes (we use zeroes to represent missing ratings in the rating matrix). One approach is to use topic modeling on movie plot summaries to identify latent themes/topics. We can learn topic representations for each item (a vector of topic proportions) and add them to the item vectors in the latent-factor model. Such topic representations of movie items are also useful outside the domain of movie recommendation. Interpretability of topics may help in explaining recommendations to users, effective content programming and ad targeting based on user profiles [**?**].

2 Problem Definition

Briefly, the problem we are trying to solve is predict how highly a user will rate certain movies based on all users' rating histories and plot summaries for all movies. Making use of these predicted ratings, we come up with movie recommendations for a user. The problem can be formalized as follows [**?**] :

We are given a list of users $U = \{u_1, u_2 ... u_m\}$ and a list of items $V = \{v_1, v_2 ... v_n\}$, where each user u_i has a list of items I_{u_i} which he/she has given ratings for. For a given user $u_a \in U$, we need to solve the following two tasks:

> **Prediction**: Estimate the predicted rating P_{a_j} of an item $v_j \notin V_{u_a}$. The prediction task can be further split into two types: *in-matrix prediciton* and *out-of-matrix prediction*. *In-matrix prediction* is the problem of predicting ratings for items that have already been rated by atleast twenty other users, whereas *out-of-matrix prediction* makes predictions about those items that have very few or no ratings (less than 20).
>
> **Recommendation**: Return a list of N items $I_r \in I$ & $I_r \cap I_a = \phi$, that the user will like most. This is simply a problem of returning the items with highest predicted rating values.

Specifically, we are interested in observing how incorporation of item topic representations increases the prediction accuracy of factor models. We would also like to analyse the interpretability of the latent topics that are captured by the topic model, however this will be just be a qualitative analysis.

3 Proposed Method

We use Probabilistic Matrix Factorization (PMF) for collaborative filtering on movie ratings and Latent Dirichlet Allocation (LDA) for topic modeling of the corpus of movie plot summaries. We then combine the latent factor model learned through PMF and the topic model learned through LDA into a single collaborative topic regression model (CTR). A CTR model essentially uses the latent topic space (latent variables) to explain observed ratings and observed documents (observed variables), thus incorporating content information into a collaborative filtering framework [**?**].

3.1 Probabilistic Matrix Factorization

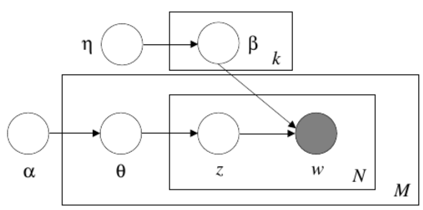
3.2 Latent Dirichlet Allocation

For a collection of text documents, a topic modeling algorithm extracts a set of 'topics', where each 'topic' is a distribution over words that occur in the documents. Words belonging to a topic are biased around a single theme. The topic model that we use for representation of documents is Latent Dirichlet Allocation (LDA), which is a generative probabilistic graphical model for collections of discrete data [**?**]. Each document in the corpus is modeled as a finite mixture over a set of underlying topics.

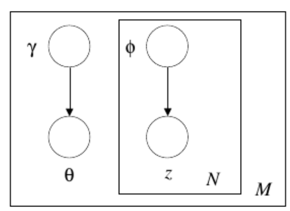
LDA has the underlying assumptions that the words in a document and documents in a corpus are exchangeable - i.e., the specific ordering of words and documents can be neglected. Now, De Finetti's representation theorem states that a collection of infinitely exhangeable random variables are conditionally independent and identically distributed, if they are conditioned on a random parameter that is drawn from some probability distribution. Now, the generative process of the LDA model is:

initialize vocabulary from corpus, the size of which is V;
for *each of the K topics k* **do**
    Choose $\beta_{k,1:V} \sim$ Exhangeable Dirichlet(η) // Draw topic distributions;
end
for *each of the M documents* **w** *in corpus D* **do**
    Choose $\theta \sim$ Dirichlet(α);
    for *each of the N words w_n in document* **w** **do**
        Choose a topic $z_n \sim$ Multinomial(θ);
        Choose word w_n from $p(w_n|z_n, \beta_{z_n})$, a multinomial probability conditioned on topic z_n
    end
end

Algorithm 1: Generative process for LDA



(a) True posterior distribution



(b) Variational posterior distribution

Figure 1: Graphical models of LDA, before and after variational approximation

Figure 1a shows the graphical model representation of LDA, which is a three-level hierarchical model. We assume that the number of topic vectors k is fixed and the vocabulary size of the corpus to be modeled is V. Now, the word probabilities are parametrized by a k X V random matrix β, each row of which represents the distribution of topics over words in the vocabulary. Each row of β is independently drawn from an exchangeable Dirichlet distribution with parameter η. Also, α is a k-dimensional vector which is a parameter for the Dirichlet random variable θ. Given the parameters α and β (which itself is a random matrix parametrized by η), the joint distribution of the topic mixture θ, the set of N topics **z** and the set of N words **w** is:

$$p(\theta, \mathbf{w}, \mathbf{z}|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta) \tag{1}$$

We make use of the representation theorem which states that the set of topics **z** are independent conditioned on θ which is a random parameter of a multinomial distribution. Next, we describe two important tasks for LDA, namely inference and estimation:

Inference: The inference task is to compute the posterior distribution of the hidden variables given the observed variable **w** (the document), assuming we know the parameters α and β. (Note that we treat β as a fixed parameter for the following discussion)

$$p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)} \tag{2}$$

Computing this distribution is intractible and hence we use variational approximate inference, as described by Blei et al. [**?**]. Simple modifications to the LDA graphical model such as dropping edges between θ, **z** and **w** and adding variational parameters lead us to the variational model as inFigure 1b, which has the following variational distribution:

$$q(\theta, \mathbf{z}|\gamma, \phi) = q(\theta|\gamma) \prod_{n=1}^{N} q(z_n|\phi_n) \tag{3}$$

The optimal values of the variational parameters (γ^*, ϕ^*) are obtained by minimizing the Kullback-Leibler (KL) divergence between the variational and true posterior distribution. By placing a Dirichlet prior on β, we get a separable variational distribution, which yields the same expressions for γ^*, ϕ^* and introduces a new variational parameter λ which has a similar expression as γ.

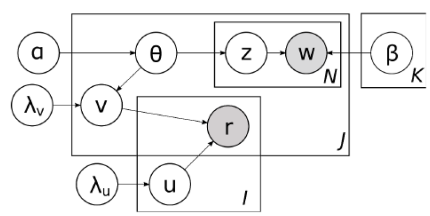
Estimation: We wish to find parameters α and η that maximize the log-likelihood of observed data, however computing the likelihood is intractable. So, we use a variational EM procedure in which we alternatingly maximize a lower bound on the log-likelihood of data with respect to variational parameters γ, ϕ and λ. This is the E-step. We then maximize this lower bound with respect to the parameters α and η, which comprises the M-step. The updates for both the parameters α and η are obtained using an efficient Newton-Raphson method in which Hessian is inverted in linear time.

3.3 Collaborative Topic Regression

The Collaborative Topic Regression (CTR) model combines a topic model such as LDA with traditional collaborative filtering [**?**]. This is done by combining both the latent topic vector and observed ratings to describe the item latent vector in the factor model of Section **??**. The graphical model and generative process of CTR are given below:

for *each user i* **do**
  Draw user latent vector $u_i \sim \mathcal{N}(0, \sigma^2\lambda_u^{-1}I_K)$;
end
for *each item j* **do**
  Draw topic proportions $\theta_j \sim$ Dirichlet(α);
  Draw item latent offset $\epsilon_j \sim (N)(0, \sigma^2\lambda_v^{-1}I_K)$;
  Set item latent vector as $v_j = \epsilon_j + \theta_j$;
  for *each word w_{jn}* **do**
    Draw a topic assignment $z_{jn} \sim$ Multinomial(θ);
    Draw word $w_{jn} \sim$ Multinomial($\beta_{z_{jn}}$);
  end
end
for *each user-item pair (i,j)* **do**
  Draw the rating $r_{ij} \sim \mathcal{N}(u_i^T v_j, \sigma^2 I)$;
end

(a) Generative process of CTR



(b) Graphical model of CTR

Figure 2: Generative process and model of CTR

Parameter Estimation: First, we train our LDA implementation on a separate training corpus and learn the model parameters α and β. Now, the log likelihood of the data is given by:

$$\mathcal{L} = \frac{-\lambda_u}{2}\sum_i u_i^T u_i - \frac{\lambda_v}{2}\sum_j (v_j - \theta_j)^T(v_j - \theta_j) + \sum_j \sum_n log(\sum_k \theta_{jk}\beta_{k,w_{jn}}) - \sum_{i,j} \frac{1}{2\sigma^2}(r_{ij} - u_i^T v_j)^2 \tag{4}$$

Computing the full posterior of u_i, v_j and θ_j given parameter β is intractable. Our approach is to maximize the likelihood function by co-ordinate ascent, iteratively optimizing the collaborative filterings u_i, v_j and topic proportions θ_j. Given topic estimate θ_j, setting the gradient of the log likelihood with respect to u_i and v_j equal to zero gives us closed form update rules for the collaborative filtering variables. Similarly, in the next phase, we can optimize the topic estimate θ_j given u_i and v_j. However, similar to the approach taken by Blei et al. [**?**], we simply set

θ_j equal to the topic estimate obtained from our LDA implementation, to save computation time without significant performance loss.[1]

Prediction : We then use the learned parameters for prediction of movie ratings. For in-matrix prediction, we use the following approximation:

$$r_{ij}^* \approx (u_i^*)^T(\theta_j^* + \epsilon_j^*) = (u_i^*)^T v_j^* \tag{5}$$

For, out-of-matrix prediction, where the movie has no ratings available, we use the following approximation:

$$r_{ij}^* \approx (u_i^*)^T(\theta_j^*) \tag{6}$$

4 Experiments

First level headings are lower case (except for first word and proper nouns), flush left, bold and in point size 12. One line space before the first level heading and 1/2 line space after the first level heading.

4.1 Headings: second level

Second level headings are lower case (except for first word and proper nouns), flush left, bold and in point size 10. One line space before the second level heading and 1/2 line space after the second level heading.

4.1.1 Headings: third level

Third level headings are lower case (except for first word and proper nouns), flush left, bold and in point size 10. One line space before the third level heading and 1/2 line space after the third level heading.

5 Conclusions

6 Citations, figures, tables, references

These instructions apply to everyone, regardless of the formatter being used.

6.1 Citations within the text

Citations within the text should be numbered consecutively. The corresponding number is to appear enclosed in square brackets, such as [1] or [2]-[5]. The corresponding references are to be listed in the same order at the end of the paper, in the **References** section. (Note: the standard BIBTEX style `unsrt` produces this.) As to the format of the references themselves, any style is acceptable as long as it is used consistently.

As submission is double blind, refer to your own published work in the third person. That is, use "In the previous work of Jones et al. [4]", not "In our previous work [4]". If you cite your other papers that are not widely available (e.g. a journal paper under review), use anonymous author names in the citation, e.g. an author of the form "A. Anonymous".

6.2 Footnotes

Indicate footnotes with a number[2] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).[3]

---

[1] We use the movies for which we have both ratings and plot summaries to learn the optimal parameters $u_{i=1:I}^*$, $v_{j=1:J}^*$, $\theta_{1:J}^*$, β^*. We also use include these movies in the training corpus of our LDA.

[2] Sample of the first footnote

[3] Sample of the second footnote

Table 1: Sample table title

PART	DESCRIPTION
Dendrite	Input terminal
Axon	Output terminal
Soma	Cell body (contains cell nucleus)

6.3 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction; art work should not be hand-drawn. The figure number and caption always appear after the figure. Place one line space before the figure caption, and one line space after the figure. The figure caption is lower case (except for first word and proper nouns); figures are numbered consecutively.

Make sure the figure caption does not get separated from the figure. Leave sufficient space to avoid splitting the figure and figure caption.

You may use color figures. However, it is best for the figure captions and the paper body to make sense if the paper is printed either in black/white or in color.

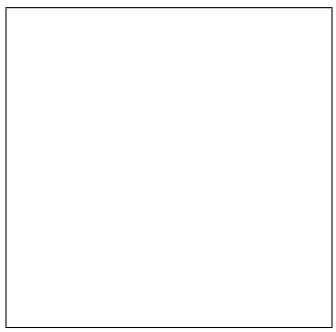


Figure 3: Sample figure caption.

6.4 Tables

All tables must be centered, neat, clean and legible. Do not use hand-drawn tables. The table number and title always appear before the table. See Table 1.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

7 Final instructions

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the **References** section; see below). Please note that pages should be numbered.

8 Preparing PostScript or PDF files

Please prepare PostScript or PDF files with paper size “US Letter”, and not, for example, “A4”. The -t letter option on dvips will produce US Letter files.

Fonts were the main cause of problems in the past years. Your PDF file must only contain Type 1 or Embedded TrueType fonts. Here are a few instructions to achieve this.

- You can check which fonts a PDF files uses. In Acrobat Reader, select the menu Files>Document Properties>Fonts and select Show All Fonts. You can also use the program `pdffonts` which comes with `xpdf` and is available out-of-the-box on most Linux machines.
- The IEEE has recommendations for generating PDF files whose fonts are also acceptable for NIPS. Please see `http://www.emfield.org/icuwb2010/downloads/IEEE-PDF-SpecV32.pdf`
- LaTeX users:
  - Consider directly generating PDF files using `pdflatex` (especially if you are a MiKTeX user). PDF figures must be substituted for EPS figures, however.
  - Otherwise, please generate your PostScript and PDF files with the following commands:

    ```
    dvips mypaper.dvi -t letter -Ppdf -G0 -o mypaper.ps
    ps2pdf mypaper.ps mypaper.pdf
    ```

    Check that the PDF files only contains Type 1 fonts.
  - xfig "patterned" shapes are implemented with bitmap fonts. Use "solid" shapes instead.
  - The `\bbold` package almost always uses bitmap fonts. You can try the equivalent AMS Fonts with command

    ```
    \usepackage[psamsfonts]{amssymb}
    ```

    or use the following workaround for reals, natural and complex:

    ```
    \newcommand{\RR}{I\!\!R} %real numbers
    \newcommand{\Nat}{I\!\!N} %natural numbers
    \newcommand{\CC}{I\!\!\!\!C} %complex numbers
    ```
  - Sometimes the problematic fonts are used in figures included in LaTeX files. The ghostscript program `eps2eps` is the simplest way to clean such figures. For black and white figures, slightly better results can be achieved with program `potrace`.
- MSWord and Windows users (via PDF file):
  - Install the Microsoft Save as PDF Office 2007 Add-in from `http://www.microsoft.com/downloads/details.aspx?displaylang=en&familyid=4d951911-3e7e-4ae6-b059-a2e79ed87041`
  - Select "Save or Publish to PDF" from the Office or File menu
- MSWord and Mac OS X users (via PDF file):
  - From the print menu, click the PDF drop-down box, and select "Save as PDF..."
- MSWord and Windows users (via PS file):
  - To create a new printer on your computer, install the AdobePS printer driver and the Adobe Distiller PPD file from `http://www.adobe.com/support/downloads/detail.jsp?ftpID=204` *Note:* You must reboot your PC after installing the AdobePS driver for it to take effect.
  - To produce the ps file, select "Print" from the MS app, choose the installed AdobePS printer, click on "Properties", click on "Advanced."
  - Set "TrueType Font" to be "Download as Softfont"
  - Open the "PostScript Options" folder
  - Select "PostScript Output Option" to be "Optimize for Portability"
  - Select "TrueType Font Download Option" to be "Outline"
  - Select "Send PostScript Error Handler" to be "No"
  - Click "OK" three times, print your file.
  - Now, use Adobe Acrobat Distiller or ps2pdf to create a PDF file from the PS file. In Acrobat, check the option "Embed all fonts" if applicable.

If your file contains Type 3 fonts or non embedded TrueType fonts, we will ask you to fix it.

8.1 Margins in LaTeX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the graphicx package. Always specify the figure width as a multiple of the line width as in the example below using .eps graphics

```
\usepackage[dvips]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.eps}
```

or

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

for .pdf graphics. See section 4.4 in the graphics bundle documentation (`http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.ps`)

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

References follow the acknowledgments. Use unnumbered third level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to 'small' (9-point) when listing the references. **Remember that this year you can use a ninth page as long as it contains *only* cited references.**

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D. S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609-616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer-Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.