Collaborative Neighborhoods

Diego Represas, David Dindi
diegorep,ddindi@stanford.edu

December 13, 2014

Abstract

Finding relevant research publications is a growing problem for researchers across all fields. Online platforms such as Mendeley and CiteULike have attempted to address this need by providing researchers the ability to share relevant articles with one another. These platforms have further sought to extend their capabilities by recommending articles to users based on the user's past interactions and preferences. The models that underlie these methods, however, are unable to provide recommendations on an item for which no prior interactions have been observed. To solve this issue, we propose Collaborative Neighborhoods (CN). CN combines elements of Collaborative Topic Regression [0] and Nearest Neighbor Models [1] to provide meaningful recommendations in both the presence and absence of past user interactions with items. We assess the performance of CN on dataset of 129,531 articles sourced from PubMed, and demonstrate that our models provides more accurate recommendations that extant recommendation frameworks.

1 Introduction

Researchers today are inundated with information; 2 million peer-reviewed articles are published every year [2]. The difficulty in finding relevant articles amidst this abundance of information has prompted citation management platforms like Mendeley and CiteULike to implement recommendation systems to aid researchers in the search [13]. These systems recommend articles based on implicit user preferences learned from a user's past article associations (likes, shares or additions to one's personal archive.)

Several collaborative filtering recommendation models have been proposed to aide in the implicit learning of user preferences. *Collaborative Topic Regression* (CTR) [0] has been the most promising model thus far. CTR applies topic modeling to augment the item feature vector used in traditional collaborative filtering. The limitation of CTR, however, is its representation of users by latent features alone. In the absence of information on past user-article associations, these user latent features cannot be accurately predicted. Consequently, CTR performs poorly on users that have fewer article associations.

We address CTRs shortcomings with CN. CN not only applies topic modeling to augment the latent feature representation of items, but does so as well for the latent feature representation of users. We begin by representing users and items by their associated topic distributions. We then proceed by learning latent variables that offset these distributions using past observations of user-article interactions. These offsets capture hidden interests that an author may have in fields that are outside their main area of research. Given that CN attributes both explicit and implicit content features to every user, it is capable both of understanding hidden preferences , and providing recommendations to users for whom there is insufficient information to learn implicit preferences.

2 Background

A basic approach to recommending text items has been to do so based on content similarity. Such methods employ probabilistic models and similarity scoring to define an article's content [3,4,5,6], or matrix factorization methods such as content-based Collaborative Filtering (CBCF) to provide recommendations to users [7,8,9]. For our particular problem, an article j's content can be thought off as a topic distribution vector, $\theta_j \in \mathbb{R}^K$, across Ktopics. When a new item is introduced, a similarity function (e.g. cosine similarity) is used to determine the k items that are most similar to the new item. The new item is then recommended to users who in the past have rated any of the k items favourably

Despite their intuitive appeal, similarity and neighborhood models (such as k-Nearest Neighbors) are inadequate for providing recommendations in research literature. This inadequacy stems from the fact that content similarity alone is not sufficient to determine whether or not an author would cite an article. A neighborhood model, for instance, would recommend an article that is of little intrinsic value to a user, solely because the article contains topics that the user has referenced in the past. This dependence on explicit features prevents neighbourhood models from capturing hidden preferences of a user.

Some of the most popular market recommendation systems, including the one used by Mendeley, are based on Collaborative Filtering (CF) methods using *latent factor models* [10,11,12,13,14]. In these models, the rating that a user j attributes to item i can be predicted through the rating function $\hat{r}_{i,j} = u_i^T v_j$, where $u_i \in \mathbb{R}^K$ is the latent factor vector for user i and $v_j \in \mathbb{R}^K$. An effective approach for implicit feedback datasets is to translate the continuous-value rating into the implicit space by setting the preference variable as follows [1, 12]:

$$\hat{p}_{i,j} = \begin{cases} 1 & \hat{r}_{i,j} > 0 \\ 0 & \hat{r}_{i,j} = 0 \end{cases} \qquad (1)$$

Because not all values of $\hat{r}_{i,j} > 0$ are equally likely to predict a user-item interaction, a confidence variable $c_{i,j} = 1 + \alpha f(r_{i,j})$ is introduced to measure the confidence of observing $p_{i,j} = 1$. The constant α is a learning rate constant and $f(r_{i,j})$ is an empirical function that depends on the dataset. The latent factor vectors u_i and v_j are then computed by minimizing the objective function:

$$\min_{u,v} \sum_{i,j} c_{i,j}(p_{i,j} - u_i^T v_j)^2 + \lambda \left(\sum_i \|u_i\|^2 + \sum_j \|v_j\|^2 \right) \qquad (2)$$

The following update rules for both latent vectors are then derived from the objective function:

$$\begin{aligned} u_i &\leftarrow (VC_iV^T + \lambda I)^{-1} VC_iP_i \\ v_j &\leftarrow (UC_jU^T + \lambda I)^{-1} UC_jP_j \end{aligned} \qquad (3)$$

Where U, V are the user and item latent factor matrices, C is the confidence variable matrix and P is the preference variable matrix. Updates are thus performed through an alternating least-squares model.

Recommender systems based on latent factor models have been shown to provide better recommendations than neighborhood methods [15,12]. However, because CF relies on observing prior user-item interactions to provide recommendations, it is unable to recommend articles that have not been previously cited or

”liked” by researchers.

To address the shortcomings of CF, David M. Blei and Chong Wang recently developed *Collaborative Topic Regression* (CTR) [0]. Much like CF, CTR assigns latent features to every user and item based on prior user-item interactions. However, CTR differs from CF in its usage of Latent Dirichlet Allocation (LDA) to construct topic distributions (drawn from $\boldsymbol{\beta} = \beta_{1:K}$ unique topics) for every item. Rather than describing v_j as an exclusively latent factor-based vector, Blei et al. choose to describe the item vector as $v_j = \theta_j + \epsilon_j$, where $\theta_j \in \mathbb{R}^K$ represents the LDA derived topic distribution of item j, and ϵ_j is a latent variable that captures fluctuations away from the topic distribution. These latent offsets augment the content similarity-based approach through the flexibility that they afford the model to understand hidden features about items and users. Since regularization for the topic-enhanced items must inherently be different than that for pure-latent users, separate parameters λ_u and λ_v are applied to user and item feature vectors respectively. The resulting log-likelihood function is intractable; a type of Expectation Maxmization algorithm is therefore used to arrive upon the optimal parameters.

$$\mathcal{L} = -\tfrac{\lambda_u}{2}\textstyle\sum_i u_i^T u_i - \tfrac{\lambda_v}{2}\sum_j (v_j - \theta_j)^T (v_j - \theta_j) \tag{4}$$

$$+\textstyle\sum_j \sum_n log(\sum_k \theta_{jk} \beta k, w_{jn}) - \sum_{i,j} \tfrac{c_{ij}}{2}(r_{ij} - u_i^T v_j)^2$$

The derived update rules for u and v then become:

$$\begin{aligned} u_i &\leftarrow (VC_iV^T + \lambda_u I)^{-1} VC_iP_i \\ v_j &\leftarrow (UC_jU^T + \lambda_v I)^{-1}(UC_jP_j + \lambda_v \theta_j) \end{aligned} \tag{5}$$

Given u and v, the topic proportions are learned by variational inference; a family of distributions on the latent variables is generated: $q(\theta, \mathbf{z} \mid \gamma, \phi) = q(\theta \mid \gamma) \prod_n q(z_n \mid \phi_n)$ and Jensen's inequality is applied to find the tight lower bound for the log likelihood function below.

$$\mathcal{L} \geq -\tfrac{\lambda_v}{2}(v_j - \theta_j)^T (v_j - \theta_j) \tag{6}$$

$$+\textstyle\sum_n \sum_k \phi jnk(log\theta jk\beta k, w_{jn} - log\phi jkn)$$

The free variational parameters upon which the distributions are generated are optimized by minimizing the Kullback Leibler (KL) divergence between the variational distribution and the true posterior. The resulting update equations, for determining θ and β are given below

$$\phi ni \propto \beta iw_n exp\{E_q[log\theta_i \mid \gamma]\} \tag{7}$$

$$\gamma_i = \alpha_i + \sum_n \phi_i \tag{8}$$

The predicted rating variable $\hat{r}_{i,j}$ for in-matrix predictions is then computed by:

$$\hat{r}_{i,j} = u_i^T v_j = u_i^T(\theta_j + \epsilon_j) \tag{9}$$

When an item is new and in-matrix prediction is not possible, the rating variable is computed ignoring the latent offset:

$$\hat{r}_{i,j} = u_i^T \theta_j \tag{10}$$

Consequently, users are still represented by latent factors but items are represented by a combination of their topical distribution and a latent factor offset; the latter aiming to capture variables conducive to a user-item interaction that are not related to the item's topic. CTR was shown to perform better as a recommender system than Collaborative Filtering using both latent factor and content-based methods.

3 Collaborative Neighbors

Given the performance improvement CTR saw by introducing topic modeling for the items, we developed an algorithm that also introduced topic modeling for the user vectors to increase prediction accuracy. In this section we describe the resulting algorithm, Collaborative Neighbors.

As in CTR, our users are I researchers and our items are J scientific articles. The preference variable $p_{i,j} \in 0, 1$ indicates whether or not user i cited article j. The preference variable is computed from the predicted ratings as in eq. (1) and the predicted rating variable remains $\hat{r}_{i,j} = u_i^T v_j$.

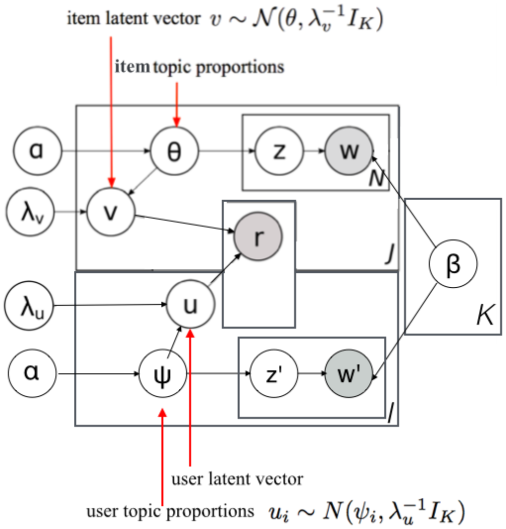

item latent vector $v \sim \mathcal{N}(\theta, \lambda_v^{-1} I_K)$
item topic proportions
α
θ
z
w
N
λv
v
r
J
β
K
λu
u
α
ψ
z'
w'
I
user latent vector
user topic proportions $u_i \sim N(\psi_i, \lambda_u^{-1} I_K)$

Figure 1. The graphical model for the CNR model

We now introduce a way to represent topic proportions for both users and items. We denote θ_j as the topic distribution for item j, where each θ_j is drawn from $\boldsymbol{\beta} = \beta_{1:K}$ unique topics. Conversely, we introduce ψ_i as the topic distribution for user i, each ϕ_i drawn from $\boldsymbol{\beta'} = \beta_{1:L}$. In our experiment, we chose ψ_i to be the average topic space of user i but there are several other strategies one could use to represent the user. Since Matrix Factorization requires for $u, v \in \mathbb{R}^K$ (the user and item factor matrices have to align), both θ and ψ have to be drawn from the same topic space or, alternatively, topic models of the same magnitude ($|\boldsymbol{\beta'}| = |\boldsymbol{\beta}|$). Since finding the optimal value of ψ, θ, u and v given $\boldsymbol{\beta}, \boldsymbol{\beta'}$ becomes intractable, our algorithm will only be tested in the case where $\boldsymbol{\beta} = \boldsymbol{\beta'}$ so we can assume $\psi \approx \theta$and use the same EM-style algorithm as in CTR to optimize for the topic distributions. The most important difference is that our update rules for u_i, v_j now become:

$$\begin{aligned} u_i &\leftarrow (VC_iV^T + \lambda_u I)^{-1}(VC_iP_i + \lambda_u \psi_i) \\ v_j &\leftarrow (UC_jU^T + \lambda_v I)^{-1}(UC_jP_j + \lambda_v \theta_j) \end{aligned} \tag{11}$$

Our algorithm then additionally has to incorporate topic modeling for the users in the same way as CTR does it for the items.

4 Experimental Study

Dataset: A total of 1.2 millions articles were retrieved from the PubMed open access dataset. We parsed the title, abstract, authors, keywords and citations associated with each article. We removed articles missing any one of the fields of interest to obtain a dataset of 129,531 scientific articles.

Articles: For each article, we concatenated its title, abstract and keywords into a document. We then removed all english stop-words and built a vocabulary consisting of the X words that appeared in our corpus more than once. Next, we used a Hierarchical Dirichlet Process (HDP) model on our dataset to determine the number of K topics contained within, and subsequently ran LDA to determine the K-vector topic distribution for each article.

Users: 1.5 million unique authors were identified in the initial open access dataset. User-article interactions were generated by mapping users to all the citations that they had across all their publications. We removed self citations (where an author cites their past work) and filtered out users that had cited fewer than 10 papers in our dataset. Our final user item matrix consisted of 26,152 users by 129,531 thousand articles, with an average of 16.9 interactions (citations) per user and 12.5 publications per user. Lastly, we assigned a topic distribution to each user by taking the average topic distribution of all papers they had co/authored. Because of computational constraints *(our tests routinely took up to 36 hours on Stanford's Barley machines due to alsqr)* , all tests were run on a random fraction of the original dataset consisting of 35,341 articles.

4.1 Evaluation

Cross-Validation: The training and testing sets for all tests were split as follows. From the original $M \times N$ matrix containing all user interactions, a total of $m = M/20$ user row indices and $n = N/20$ item column indices were randomly selected. Interactions belonging to these m randomly selected users were separated into a different $m \times N - n$ matrix for user out-of matrix cross validation. Interactions belonging to the n randomly selected items were also separated, this time into a $M - m \times n$ matrix for user out-of matrix cross validation. Interactions in both the m selected rows and n selected columns were discarded. The remaining bulk of the interactions were Assigned to a $M - m \times N - n$ matrix. Of these interactions, 10% were randomly selected and placed in a separate $M - m \times N - n$ for in-matrix prediction cross-validating. The other 90% were used for training the models. The user and item topic matrices were split and distributed according to where their representative rows/columns were assigned during the shuffle. We saw no significant changes in our training and testing recall values when implementing multiple folds of this matrix shuffle split.

In-matrix predictions: Following Blei and Wang's prior work, we established our measures of accuracy to be standard precision and recall within the first 1-200 highest-scored predictions. For in-matrix recommendations, recall consisted of the number of articles recommended to a user that were also in the testing set over the total number of articles in that user's testing set.

Out-of-matrix predictions: For out-of-matrix recommendations, recall consisted of the number of articles recommended to a user that were also in the out-of-matrix set over the total number of articles in that user's out-of-matrix set.

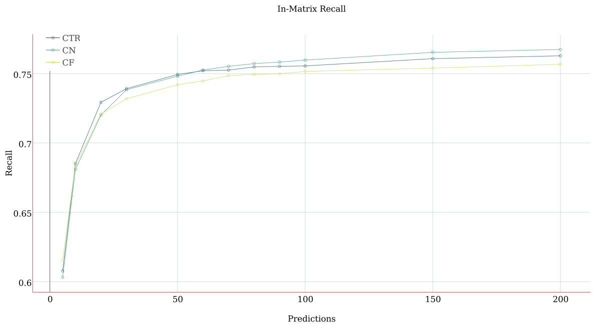
4.2 Settings

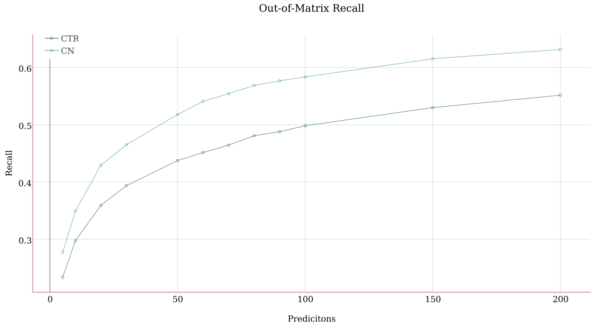
Blei and Wang established that the independence of the topic distributions θ_j relative to user vectors allowed for the optimal topic distributions to be found before beginning to optimize for u_i and $v_j = \epsilon_j + \theta_j$. Consequently, we used HDP to find an optimal topic number of $K = 200$ and also set the dimensions of our latent vectors equal to K. We set $\alpha = 1$ and iterated for values of $\lambda_v, \lambda_u \in \{.01, .1, 1, 10, 100\}$ to find that $\lambda_v = 10$,$\lambda_u = .01$ gave optimal results for CTR using 25 training epochs. Values of $\lambda_v = 10$, $\lambda_u = .1$ gave optimal results for CN. As prior literature had already shown CTR outperforms CF, we simply used $K = 200, \lambda_u = \lambda_v = 0.01$ for our CTR iterations to provide a point of reference without optimizing the parameters.

4.3 Results:

Comparisons: Our algorithm performed better than Collaborative Filtering and comparable to Collaborative Topic Regression when performing in-matrix recommendations. This is consistent with prior research on the performance of CTR. In our particular dataset, we found that recall using CN was almost identical to that measured using CTR during the first predictions and consistently higher than CTR in the ranges after 20 predictions. This could mean that our algorithm generally performs better in-matrix recommendations than CTR but it could also be attributed to the more direct topical connections of our dataset.

For item out-of-matrix predictions, our algorithm demonstrated having considerably higher recall figures compared to CTR, indicating the value of our algorithm is strongest when performing cold-start recommendations. We attempted to perform user out-of-matrix predictions multiple times with no more success than random recommendations. We attribute this failure to the strategy we used to establish the user topic space as an averaging of a user's publications is likely to be similar to too many publications without much specificity.


In-Matrix Recall


Out-of-Matrix Recall

Examining user Profiles: We can explicitly analyze user profiles from the LDA representation of topics that were generated by averaging the topic distribution across all their publications. With CN, we are able to extend the analysis further by analyzing the magnitude of each latent offset from the authors topic distribution. These offsets represent large deviations of an author's preferences, from their core topics of research. For instance, a financial engineer might apply electrical engineering signal processing techniques to filter time series data in his/her research. More generally, these offsets capture passive interests that can allow recommender systems to make more informed judgments on what to recommend. In our dataset, we observe the topic associated with words such as south, variation, india, asia, geographic and madagascar exhibit the largest offsets.

Examining Article Characteristics: Similarly CN allows us to understand which topics attract a broad range of interest from researchers from a diversity of fields. We accomplish this analysis by calculating the average magnitude of each item latent feature offset. Topics with a large offsetting magnitudes on the item side, tend to be topics that have been cited by users in a variety of fields. In our dataset we observe topics associated with words such as world, skin, biodiversity and communities to have the largest offsets.

5 Conclusions

We demonstrate in this study the ability of Collaborative Neighbourhoods of predicting the citations made by researchers in medical literature. We obtain results that are superior to traditional Collaborative Filtering and Collaborative Topic Regression in in-matrix and out-of-matrix predictions. Furthermore, we demonstrate the ability our model to develop a semantic understanding of an author's preferences, as well as to identify the types of articles that enjoy the most reception from a variety of fields. Our method can be generally applied to any user-item recommendation problem where there is sufficient information about each user. Future work will focus on eliciting further meaning from the latent features derived, and augmenting the ability of CN to recommend items to users whose prior interactions have not been observed.

References

[0] Wang, Chong and David M. Blei. ”Collaborative Topic Modeling for Recommending Scientific Articles.” Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: ACM, 2011, 448-456.

[1] Hu, Y., Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets.” Data Mining, 2008. ICDM’08. Eighth IEEE International Conference on. 2009

[2] ”STM International Association of Scientific, Technical and Medical Publishers.” STM. N.p., n.d. Web. 13 Dec. 2014.

[3] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. Journal of Machine Learning Research, 3:9931022, January 2003.

[4] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D. Blei. Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, Advances in Neural Information Processing Systems 22, pages 288296, 2009.

[5] S. M. Gerrish and D. M. Blei. Predicting legislative roll calls from text. In Proceedings of the 28th Annual International Conference on Machine Learning, ICML 11, 2011.

[6] D. Agarwal and B.-C. Chen. flda: matrix factorization through latent Dirichlet allocation. In Proceedings of the third ACM international conference on Web search and data mining, WSDM 10, pages 91100, New York, NY, USA, 2010. ACM.

[7] Content-Based Recommendation Systems Michael J. Pazzani, Daniel Billsus

[8] Burke, R.: Hybrid Web Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web: Methods and Strategies of Web Personalization. LNCS, vol. 4321, pp. 377408. Springer, Heidelberg (2007)

[9] Mooney, R. J., and Roy, L. 2000. Content-based book recommending using learning for text categorization. In Proceedings of the Fifth ACM Conference on Digital Libraries, 195204.

[10] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In Proceedings of the 25th International Conference on Machine learning, pages 880887. ACM, 2008.

[11] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. Advances in Neural Information Processing Systems, 20:12571264, 2008.

[12] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. IEEE Computer, 42(8):3037, 2009.

[13] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1928, New York, NY, USA, 2009. ACM.

[14] K. Yu, J. Lafferty, S. Zhu, and Y. Gong. Large-scale collaborative prediction using a nonparametric random effects model. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 11851192, New York, NY, USA, 2009. ACM.

[15] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information