International Journal of Future Generation Communication and Networking
Vol. 12, No. 5, (2019), pp. 275- 279

Multi-Task Learning in Collaborative Filtering Recommender Systems

Priyanka Muniyandi [1], Dr. Ananthi Sheshasaayee [2]
[1] Research Scholar, Quaid-E-Millath Government College for Women., [2] Head & Associate Professor, Quaid-E-Millath Government College for Women

Abstract:
Multitask Learning is an approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias. It does this by learning tasks in parallel while using a shared representation; what is learned for each task can help other tasks be learned better. This paper focuses on exploring personalized multi-task learning approaches for collaborative filtering towards the goal of improving the prediction performance of rating prediction systems. These methods first specifically identify a set of users that are closely related to the user under consideration (i.e., active user), and then learn multiple rating prediction models simultaneously, one for the active user and one for each of the related users.A fundamental challenge for collaborative filtering algorithm is data sparsity. In practice, most users do not provide ratings for most items and thus the user-item matrix is very sparse with many ratings left undefined. As a result, the accuracy of recommendation is often quite poor. To address this problem, a number of techniques have been proposed.In this paper, to tackle the data sparsity problem multiple classification problems for all users can be solved at the same time. In the machine learning literature, this approach is known as multi-task learning or transfer learning . The rationale behind transfer learning is that learning multiple classifiers together allows transforming information among them and thus improves the overall accuracy while requires less training data.

Keywords*: Collaborative Filtering, Multi-task Learning, data sparsity.*

1. Introduction

Recommender systems strive to solve the scarcity of information by creating custom recommendations that help people find specific items or documents. With a number of commercial systems deployed, these systems played an important role in e-commerce and information filtering.[1]Two of the most common approaches to building recommender systems are: Content-based filtering and Collaborative Filtering.

Collaborative Filtering: In a given area, collaborative filtering systems collect user feedback in the form of ratings. The systems then use these data to discover users with most similar profiles and use their ratings to predict ratings for new items.

Content-based Filtering: Diversly, content-based filtering correlates content illustrations of items in order to detect the item that is most identical to the items that attracts the user.[2]There is a basic challenge in collaborative filtering technique, which is data sparsity of deficiency. Most of the users are not active enough to render their ratings for many of the items. As a result, the user-item matrix is very insufficient with multiple ratings left undefined. A number of techniques have been projected as a solution to this problem.
Billsus and Pazzani suggested to stuff in the rating matrix with 0, and Singular Value Decomposition is used to scale down data dimensionality[3] Zitnick and Kanade implemented the use of maximum entropy to mititgate the impact of data deficiency. [4] There are other methods such as incorporating both memory-based and model-based techniques [5] or combining content-based and collaborative filtering [6]
In this paper the data sparsity problem is addressed by a machine learning literature called, Multi-task learning also known as transfer learning.

2. Multi-task Learning

Multi-task learning is a transfer learning technique devised to enhance the generalization performance of a given model by influencing the domain-specific knowledge enclosed in the training signals of related tasks. [7] In multi-task learning, multiple related tasks are represented by a common representation ,and then they are learned in parallel, such that information from one task can be transferred to another task and help achieve performance boost.

Multi-task learning works by training tasks in correspondence using a common representations. This can be implemented simply by including additional tasks such as extra outputs to a net as a wide invisible layer for all tasks in backpropagation network. The general thought is that the extra task' error gradient moves the common invisible layer towards the representations that better indicate consistencies of the domain as a whole and these better assist learning the main task.
Task 1 is distressed by Tasks 2-4 only by their impact on the common invisible layer. When the net is used to make predictions for Task 1 the outputs for Tasks 2-4 are neglected. [8]

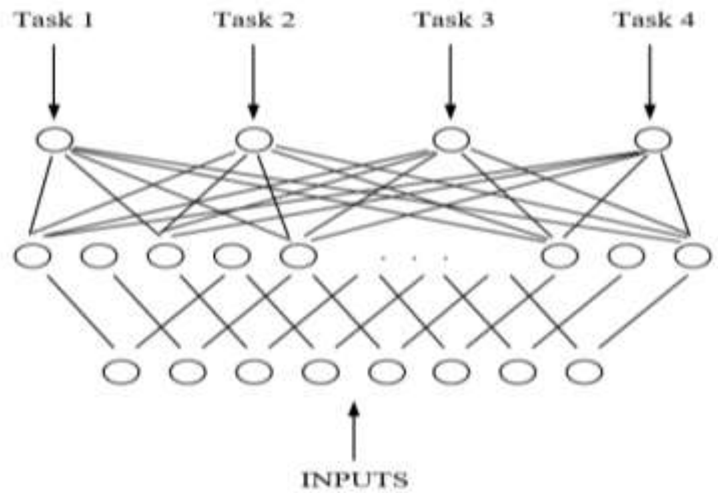

Figure 1. Multi-task Learning with Backpropagation of 4 Related Tasks Defined on the Same Inputs.

3. Multi-task Learning for Collaborative Filtering.

The nature of current multi-task learning systems for collaborative filtering is that they treat the problem of learning the CF-based rating prediction models for the individual users as different learning tasks and they build a single multi-task prediction model that combines all these learning tasks. A direct consequence of that is that from the point of view of a specific user (ui), the resulting multi-task model will be influenced both from the learning tasks that correspond to users that exhibit similar rating patterns with ui (i.e., related tasks) and from the learning tasks that correspond to users that exhibit highly dissimilar and uncorrelated rating patterns with ui (i.e., unrelated tasks). This goes against the transfer learning principle that underlies multi-task learning approaches [9] which are designed to leverage information that can be concurrently learned from a set of related tasks.[10]
The multi-task learning approach can solve the CF-based rating prediction problem by constructing a multi-task model that is specially drafted for each user instead of constructing a individual model for all the users. Precisely, m users, $N_* = \{u_{*1},...,u_{*m}\}$ can be identified originally for each active user u_*, such that the m users are related to u_*, treats their corresponding rating prediction problems as the related learning tasks, and uses them to build a multi-task learning-based rating prediction model for u_*. The

ratings from the users in $\{u_*\} \cup N_*$ can be considered excluding any rating information from users that are unrelated to u_*.
The multi-task model of user u_* is learned using error-insensitive Support Vector Regression (ϵ -SVR) [11]. The input to the model are tuples of the form $((u_i,i_j),r_{i,j})$,where $u_i \in \{u_*\} \cup N_*, i_j \in I_i$,and $r_{i,j}$ is the rating that ui has given to i_j. The user-item tuples(i.e., (u_i,i_j)) represent the instances on which the model is being learned and their corresponding ratings (i.e., ri,j)represent the target values that are estimated by the learned regression function $f(\cdot)$. Once $f(\cdot)$ has been estimated, the prediction from the active user $u*$ on an unrated item $i*$ is determined as $f((u_*,i_*))$.

Based on the previously developed kernel-based approaches for multi-task learning [12]the multi-task kernel function K_{mt} on training instances (i.e., user-item tuples) for support vector regression is defined as

$$K_{mt}((u_i,i_j),(u_{i0},i_{j0})) = K_u(u_i,u_{i0}) \times K_i(i_j,i_{j0}), \qquad [13]$$

where K_u and K_i are kernel functions defined on the users and the items, respectively. When K_u and K_i are valid kernels (i.e., positive semi-definite),K_{mt} is a valid kernel as well [14].

The kernel function K_{mt} compares two tasks (users) $(u_i,i_j),\forall i_j \in I_i$ and$(u_{i0},i_{j0}),\forall i_{j0} \in I_{i0}$ by computing their similarity as the tensor of corresponding user similarity (i.e.,K_u) and item similarity (i.e.,K_i). Incase $u_i = u_{i0}$, the kernel regression minimizes learning errors for only user u_i (i.e,K_u (u_i,u_i) is constant,and thus single-task learning). Incase $i_j = i_{j0}$, the kernel regression transfers user taste on item i_j across user ui and ui0 through K_u (i.e,K_i (i_j,i_j) is constant).

Selection Of Most Similar Users

For a given user u_*, the m most related users $N_* = \{u_{*1},u_{*2},...,u_{*m}\}$ are selected as those users whose historical ratings on co-rated items are the most similar to user $u*$. The historical rating similarity between two users is computed using the modified version of Pearson correlation coefficient employed by user-based approaches [15], which is given by

$$\mathrm{sim}_u(u_i,u_j) = p_u(u_i,u_j) \cdot \frac{\sum_{i_k \in I_c} (r_{i,k} - \bar{r}_{i,\cdot})(r_{j,k} - \bar{r}_{j,\cdot})}{\sqrt{\sum_{i_k \in I_c} (r_{i,k} - \bar{r}_{i,\cdot})^2}\sqrt{\sum_{i_k \in I_c} (r_{j,k} - \bar{r}_{j,\cdot})^2}}, \qquad [16]$$

Where I_c is the set of items that have been co-rated by users u_i and u_j (i.e, $I_c = I_i \cap I_j$), $\bar{r}_{i,\cdot}$ and $\bar{r}_{j,\cdot}$ are the average ratings of users u_i and u_j over all their rated items I_i and I_j, respectively. p_u is a penalty factor that linearly penalizes the similarity value when the number of co-rated items is smaller than a pre-defined small constant C, and p_u is defined as $p_u(u_i,u_j) = \min(|I_c|,C)/C$. Note that since the Pearson correlation coefficient can be negative, only the users whose similarity to the active user u_* is positive are selected.

Conclusion

Multitask learning allows inductive bias to be acquired via the training signals for related additional tasks drawn from the same domain. This paper demonstrates that the benefit of using extra tasks can be substantial. An interesting direction of future research is to explore further the multi-task learning algorithm.

References

1. Zhang, Y., & Yang, Q. (2017). A survey on multi-task learning. arXiv preprint arXiv:1707.08114.

2. Phuong, D., & Phuong, T. M. (2008, July). Collaborative filtering by multi-task learning. In 2008 IEEE International Conference on Research, Innovation and Vision for the Future in Computing and Communication Technologies (pp. 227-232). IEEE.

3. Billsus, D., & Pazzani, M. J. (1998, July). Learning Collaborative Information Filters. In Icml (Vol. 98, pp. 46-54).

4. Zitnick, C. L., & Kanade, T. (2004, July). Maximum entropy for collaborative filtering. In Proceedings of the 20th conference on Uncertainty in artificial intelligence (pp. 636-643). AUAI Press.

5. Xue, G. R., Lin, C., Yang, Q., Xi, W., Zeng, H. J., Yu, Y., & Chen, Z. (2005, August). Scalable collaborative filtering using cluster-based smoothing. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 114-121). ACM.

6. Basilico, J., & Hofmann, T. (2004, July). Unifying collaborative and content-based filtering. In Proceedings of the twenty-first international conference on Machine learning (p. 9). ACM.

7. Richard A. Caruana. Multitask learning: A knowledge-based source of inductive bias. In Proceedings of the Tenth International Conference on Machine Learning, pages 41–48. Morgan Kaufmann, 1993

8. Caruana, R. (1996, July). Algorithms and applications for multitask learning. In ICML (pp. 87-95).

9. Thrun, S., & O'Sullivan, J. (1996, July). Discovering structure in multiple learning tasks: The TC algorithm. In ICML (Vol. 96, pp. 489-497).

10. Abernethy, J., Bach, F., Evgeniou, T., & Vert, J. P. (2009). A new approach to collaborative filtering: Operator estimation with spectral regularization. Journal of Machine Learning Research, 10(Mar), 803-826.

11. Alex J. Smola, Bernhard Sch¨olkopf, and Bernhard Sch¨olkopf. A tutorial on support vector regression. Technical report, Statistics and Computing, 2003.

12. Dumitru Erhan, Pierre-Jean L'Heureux, Shi Yi Yue, and Yoshua Bengio. Collaborative filtering on a family of biological targets. Journal of Chemical Information and Modeling, 46(2):626–635, 2006.

13. Jacob Abernethy, Francis Bach, Theodoros Evgeniou, and Jean-Philippe Vert. A new approach to collaborativefiltering:Operatorestimationwithspectralregularization. J.Mach.Learn.Res.,10: 803–826, 2009.

14. Vapnik, V. (1998). Statistical Learning Theory. New York: John Willey & Sons.

15. Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In SIGIR '99: Proceedings of the 22nd annual internationalACMSIGIRconferenceonResearchanddevelopmentininformationretrieval,pages 230–237, New York, NY, USA, 1999. ACM.

16. G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 17(6):734–749, 2005.