Model Explanations under Calibration

Rishabh Jain
jain.rishabh8897@gmail.com
Imperial College London

Pranava Madhyastha
pranava@imperial.ac.uk
Imperial College London

ABSTRACT

Explaining and interpreting the decisions of recommender systems are becoming extremely relevant both, for improving predictive performance, and providing valid explanations to users. While most of the recent interest has focused on providing local explanations, there has been a much lower emphasis on studying the effects of model dynamics and its impact on explanation. In this paper, we perform a focused study on the impact of model interpretability in the context of calibration. Specifically, we address the challenges of both over-confident and under-confident predictions with interpretability using attention distribution. Our results indicate that the means of using attention distributions for interpretability are highly unstable for un-calibrated models. Our empirical analysis on the stability of attention distribution raises questions on the utility of attention for explainability.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

calibration, explanations, attention, recommender systems, deep learning

ACM Reference Format:
Rishabh Jain and Pranava Madhyastha. 2019. Model Explanations under Calibration. In *Paris '19: International Workshop on Explainable Recommendation and Search at ACM-SIGIR.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recommendation systems are used for item filtering based on user preferences in a variety of areas including movies, news, books, social recommendations and products in general. Some commonly used approaches to recommendation systems include Collaborative filtering, Content-based filtering and hybrid systems. There has been an increased interest in the community in utilizing deep learning based models for recommender systems [Zhang et al. 2019]. These models can alleviate several limitations of traditional models including complex non-linear transformations, interactions with different types and modalities of inputs. Deep learning based models have been particularly shown to be flexible and are known to incorporate additional data when training and can learn from large amounts of auxiliary information, which is usually available to recommendation systems. As deep models are modular than other rigid algorithms they are easily adaptable and extendable.

The significance of explaining automated recommendations is widely acknowledged [Herlocker et al. 2000; Tintarev and Masthoff 2007]. Explanations build user trust, improve their experience, and also give them the opportunity to fix incorrect representations or recommendations. For these reasons, there has been extensive research on ways to explain different types of recommendation systems. We refer the reader to Zhang and Chen [2018] for a detailed survey on explainable systems.

Deep learning based recommendation systems have opened up one way of explaining neural models' outputs in the context of recommendations [Zhang and Chen 2018] — by using attention distributions. In this context, neural attention mechanisms have gained significant focus, as they have been shown to not only help the model perform better, but also provide explanations by highlighting the input features that play a significant part in computing the model's output [Goossens et al. 1999; Xue et al. 2019]. However, it is has been recently indicated that attention may not always provide a reliable form of explanation, especially in the domain of natural language processing [Jain et al. 2019].

One of the emerging problems with the modern neural network models (especially deep neural networks) is their poor calibration [Guo et al. 2017]. Over-confident or under-confident predictions can make a model unreliable, especially in sensitive scenarios like health care (disease detection), autonomous driving among others [Guo et al. 2017].

In this paper, we focus on a form of recommendation system that aims to answer *why* a certain recommendation has been made. Especially, we investigate the reliability of attention distributions in deep neural attention based recommendation systems.

2 BACKGROUND AND TOOLS

In this paper, we investigate the utility of attention with a state-of-the-art deep neural network based model with attention [Xue et al. 2019]. In this section, we succinctly describe the necessary background and the tools under consideration.

2.1 Attention Distribution

Attention mechanisms, in neural networks, are known to provide the functionality for the model to focus on certain parts of the inputs or features. An attention mechanism in recommendation systems is usually over u, a user representation, with the set of item specific representations $\{v_i\} \in \mathcal{V}$ where $\mathcal{V}$ is the domain of all item representations. A compatibility function maps u and $\{v_i\}$ to a scalar distribution, which is then typically converted into a probability distribution using a softmax operator. This usually results in a distribution where some items get more probability mass than others, indicating their influence in the decision made by the system. In this paper, we focus on such attention distributions

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
EARS'19, July 25, 2019, Paris, France
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

and are interested in their reliability. We are especially interested in understanding the behaviour of the models when the models are mis-calibrated.

Explanation using Attention. In neural recommendation systems (and neural networks in general), attention is increasingly being used, not just to improve the model's performance but also as a means to explain the model's predictions [Gilpin et al. 2018; Wang et al. 2018; Xue et al. 2019]. The attention maps (heat-maps) are used to indicate which input features to the model were majorly responsible for the model's predictions. In Figure 1 (from a movie recommendation system from Xue et al. [2019]), for instance, for target item #1525, the attention-network assigns the maximum weight to the item #1254 (one of the previously interacted items of the target user). This information can be used to generate a human-readable explanation like "You are recommended to watch #1525 because you watched #1254".

More recently, there has been research on the reliability of attention-maps based explanations [Jain and Wallace 2019] and if they can be used to explain a model. In this paper, we work on this line of research in the context of recommendation systems and their calibration(2.2).

2.2 Model Calibration

Classification models used as part of any decision process need to be both accurate in their predictions, and should also indicate when they are probably incorrect. Model calibration is the degree to which a model's predicted probability correlates with its true correctness likelihood. Calibration measures this property of a model. For example, if a perfectly calibrated model gives 100 different predictions, each with 80% confidence (probability), 80 of the predictions should be classified correctly.

We use the concept of calibration to plot reliability diagrams [Hamill 1997]. A reliability diagram can be defined as the accuracy of the model as a function of its confidence. Reliability diagrams help us visualize a model's calibration. A reliability plot which falls below the identity function suggests that the model is over-confident of its predictions (blue plot in Figure 3) since it means that the ground truth likelihood (accuracy) is less than the model's confidence in its predictions. On the other hand, it is considered under-confident if the reliability plot is above the identity function. For a perfectly calibrated model, the reliability plot is the identity function.

2.3 Attention Permutation

One of the experiments we perform to check the reliability of attention based explanations is permuting the weights randomly and recording the effects of the permutations on the output of the model (inspired from Jain and Wallace [2019]).

Since the particular weights assigned to the input features are used as the basis for the explanations, permuting these weights randomly should cause the model's prediction to change by a substantial margin. In case the predictions remain unchanged it indicates that the attention necessarily doesn't contribute to the predictions. This can be concerning especially when using attention as grounds for explanations.

2.4 Model Stability

In our study, we refer to model stability as the consistency of model predictions and internal parameters with different runs of the model by only changing random seeds. [Jiang 2003, 2007]. The seed values are responsible for regulating the training dynamics (weight initialization, training batch generation, among others). This way, we get to measure the impact of these random processes on the output of the model (and the attention weights).

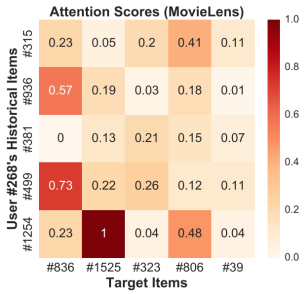

Figure 1: Attention map showing weights assigned to input features

3 EXPERIMENTAL SETUP

In the following sections we describe our experiments and observations.

3.1 DeepICF with attention

The DeepICF model uses a deep neural network to learn latent low-dimensional embeddings of users and items that capture implicit and explicit user interactions. It uses a pair-wise interaction layer, which consists of an element-wise product (also called the Hadamard product[Wikipedia contributors 2019a]) of the target item's latent vector with each of the historical items' vectors. The model then follows this with the pooling layer whose output is a vector of fixed size, to facilitate the deep interaction of layers. This is done via *attention based pooling*. The output of the pooling layer is a vector which condenses the second-order interaction between historical items and the target item (we refer the reader to Xue et al. [2019] for a detailed explanation of the model). Finally, the higher order interactions are captured with a multi-layer perceptron. The output of the model is a sigmoid on the final layer's weighted sum.

Modifications: We replaced sigmoid function with a softmax with two outputs over the two classes and trained the model with cross-entropy loss.

3.2 Dataset, Evaluation and Hyperparameters

We train, evaluate the model and perform our experiments on the MovieLens[1] dataset. This dataset has been commonly used to evaluate collaborative filtering algorithms. The dataset contains one million ratings where each user has at least 20 ratings and use the standard splits. In our study, we retain the standard procedure used in DeepICF where the original dataset is transformed such that

[1]https://grouplens.org/datasets/movielens/1m/

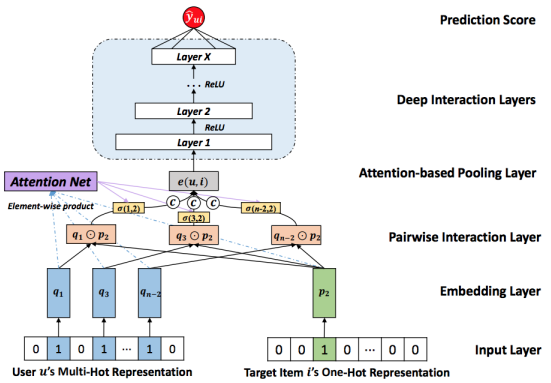

Figure 2: Deep ICF with Attention (from Xue et al. [2019])

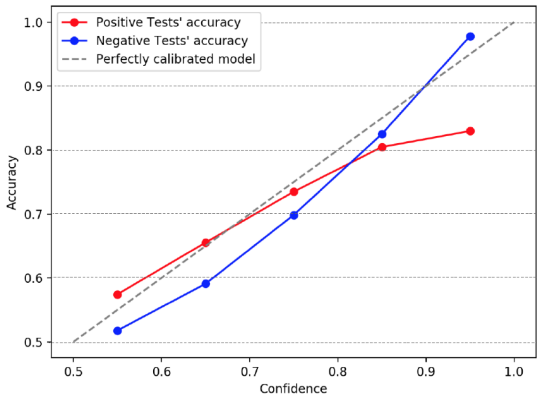

Figure 3: Calibration plot (Reliability diagram) of the DeepICF model.

each user item entry is marked as 1: when there is some interaction between the user and item and -1: when there is no interaction the between the target user and item.

Evaluation: For evaluation purposes, the standard metrics used are HitRatio (HR@10) and the NDCG@10(Normalized Discounted Cumulative Gain [He et al. 2015]) as the main metrics. We further use the binary labelling accuracy to investigate the model performance per class, where the classes are defined as: -1 when there is no interaction between the user and items and 1 when there is an explicit interaction (user ratings for the item).

Hyperparameter Settings: For training purposes, we use the same hyper-parameters as mentioned in the paper [Xue et al. 2019]. We use the original DeepICF implementation[2].

4 RESULTS

Table 1 compares the performance of the softmax output model with the original DeepICF model and the state-of-the-art Neural Matrix Factorization model[He et al. 2017]. We observe that the performance of our model is highly competitive and performs as well as the DeepICF with pretraining. In the following sections, we will investigate the reliability of models and the attention distribution in the models.

4.1 Calibration

We plot the reliability diagram for the DeepICF model by bucketing the model predictions based on their confidence and calculating the accuracy for each of the buckets[3].

We see in Figure 3, for positive test cases, the DeepICF (with attention) model seemingly tends to be over-confident as the confidence increases, where the model tends to be extremely confident about predicting the positive class without being as accurate. This can be problematic especially when dealing with real-world production systems. We also notice that the model is seemingly over-confident in predicting the negative class. This could be because of the imbalance in the dataset where the dataset is extremely skewed towards the negative class. We also note that the test-set has a very high degree of imbalance in the number of positive and negative test cases in our test set (1 positive sample for every 99 negative tests). This is impacted in Figure 3 as it shows the curves for positive and negative test samples separately.

4.2 Attention Permutation

What is the effect of over-confidence over attention? In order to test the reliability of explanations generated from attention, we permute the attention weights randomly and notice the effect of the permutation on the output of the model (as described in Section 2.3).

Specifically, in DeepICF, as shown in Figure 2, the attention based pooling layer assigns a weight for each of the user and item interaction, where the magnitude of the weights indicate the importance of the interaction. In this experiment, we randomly *shuffle* these weights amongst the items and record the difference in the output prediction score (originally classified interaction label). We randomly shuffle the weights 100 times (as performed in Jain and Wallace [2019] for each test case, and average the absolute variations in the output predictions.

We plot the average variations in false negatives (right axis) against the confidence of the predicted output for the positive test cases in Figure 4. We focus on positive test cases as it is the most salient label to measure the model. The plot also contains the reliability diagram for the model (left axis). We note that the perturbations especially have barely any effect on the mis-calibrated cases. In both false positives and false negatives (these increase with mis-calibration), we notice similar trends where the effect of permuting the attention weights decreases as the confidence in the predicted label increases. **Thus, showing that model explanations generated from the attention distribution become less reliable with over-confident predictions.**

4.3 Fixing the effect of Class-imbalance

As the training split of the dataset is heavily imbalanced: 4 negative labels (no interactions) for every positive label, we use a simple class-weighting heuristic, to cope with this imbalance in the training set and modify our cross-entropy loss. The new loss is calculated by assigning weights to the losses from the test cases such that the loss contribution from both the classes (positive and negative

[2]https://github.com/linzh92/DeepICF
[3]Refer to the Appendix for further details on Calibration plots (Reliability diagrams)

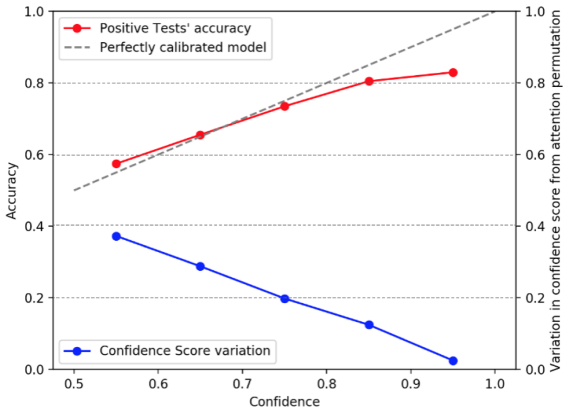

Figure 4: Figure showing the effect of attention permutation (right axis) on the prediction score of wrongly (negatively) classified positive test cases (false negatives).

interactions) is balanced[King and Zeng 2001]. We retrain the model with the new loss function and were able to achieve similar HitRatio values to the original model as shown in Table 1. We analysed the effect of attention permutation (Section 4.2) on this model. Figure 5 compares the new model to the previous model's results. We notice that the new model is considerably more sensitive to attention permutation, compared to the original one. **This suggests that attention based explanations from the class-balanced loss model are more reliable than the original model.**

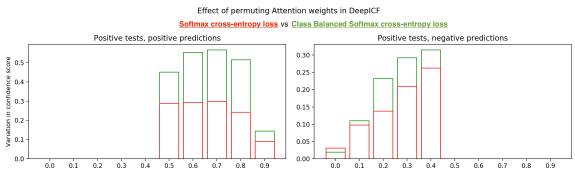

Figure 5: Effect of permuting attention weights.

4.4 Stability of DeepICF

We now consider the effect of random seeds and on initialization of model parameters and in general the model performance. We notice in Table 1 — the standard deviation is generally very low suggesting that the performance of the model is seemingly stable and it seems to have small deviation.

Attention score stability: What is the effect of random seeds on attention distribution? As we are interested in the reliability of attention *explanations*, we focus on the stability of attention scores in DeepICF. We perform the same experiment by running the same model but with 10 different random seeds and record the *top 10%* of the most attentive items (user-item interactions which get the highest attention weight assigned) for every particular test case for each model. Then we compare if these top 10 percent most attentive items for a particular test case are consistent for different runs of the models with different random seedst. We calculate the similarity between two sets of items by computing the `Jaccard Index` [Wikipedia contributors 2019b] of the sets. We calculate the Jaccard Index for every possible pair of sets of top attentive items and average over them. Figure 6 shows that the average `Jaccard Index` for positive predictions with high confidence is around 0.5 (where max `Jaccard Index` is 1, implying completely stable attention scores). **This highlights that the attention explanations from two identical models, trained with different seeds for the same input can vary, severely highlighting the unreliability of such explanations.**

Model type	Hit Ratio@10 (%)	NDCG@10 (%)
DeepICF*	68.81	41.13
DeepICF*+Pretrain	70.84	43.80
NeuMF*+Pretrain	70.70	42.60
DeepICF (ours)	70.41(±0.24)	43.00 (±0.34)
DeepICF+cls-wt	68.61	41.14

Table 1: Performance Comparison for DeepICF and NeuMF[He et al. 2017]. * indicates scores directly from the corresponding papers. The standard deviation (±) is obtained with 10 runs of the model with different random seeds.

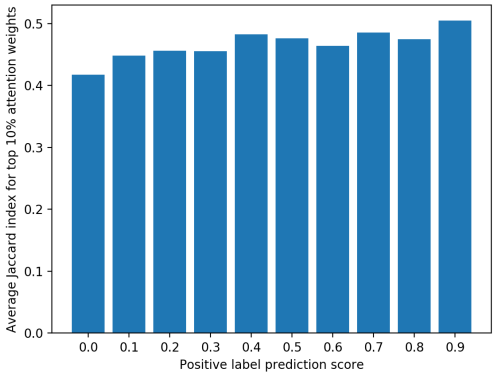

Figure 6: Attention score stability.

5 CONCLUSION

In this paper, we have explored the importance of model dynamics and its relation to explanation using attention. Concretely, we observe that attention may not be reliable when the selected model is especially mis-calibrated. We have explored one possible way of stabilizing the model by accounting for the class imbalance. Significantly, we noticed that using an inverse-class weighted cross-entropy formulation can help improve the stability of attention distribution. Further, we observe that over different runs of models with different random seeds, the models seem to obtain different attention distributions. We posit that our work is extremely relevant to the community and can orient towards an important discussion on the reliability of using attention as an explanation.

REFERENCES

Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining Explanations: An Overview of Interpretability of Machine Learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 80–89.

Michel Goossens, S. P. Rahtz, Ross Moore, and Robert S. Sutor. 1999. *The Latex Web Companion: Integrating TEX, HTML, and XML* (1st ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1321–1330.

Thomas M Hamill. 1997. Reliability diagrams for multicategory probabilistic forecasts. *Weather and forecasting* 12, 4 (1997), 736–741.

Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 1661–1670.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.

Jonathan L Herlocker, Joseph A Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. ACM, 241–250.

Sarthak Jain, Ramin Mohammadi, and Byron C. Wallace. 2019. An Analysis of Attention over Clinical Notes for Predictive Tasks. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*. Association for Computational Linguistics, Minneapolis, Minnesota, USA, 15–21. https://www.aclweb.org/anthology/W19-1902

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. *Annual Conference of the North American Chapter of the Association for Computational Linguistics* (2019).

Yulei Jiang. 2003. Uncertainty in the output of artificial neural networks. *IEEE transactions on medical imaging* 22, 7 (2003), 913–921.

Yulei Jiang. 2007. Uncertainty in the output of artificial neural networks. In *2007 International Joint Conference on Neural Networks*. IEEE, 2551–2556.

Gary King and Langche Zeng. 2001. Logistic regression in rare events data. *Political analysis* 9, 2 (2001), 137–163.

Nava Tintarev and Judith Masthoff. 2007. A survey of explanations in recommender systems. In *2007 IEEE 23rd international conference on data engineering workshop*. IEEE, 801–810.

Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. 2018. Tem: Tree-enhanced embedding model for explainable recommendation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1543–1552.

Wikipedia contributors. 2019a. Hadamard product (matrices) — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Hadamard_product_(matrices)&oldid=895450816 [Online; accessed 19-May-2019].

Wikipedia contributors. 2019b. Jaccard index — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Jaccard_index&oldid=890384326 [Online; accessed 18-May-2019].

Feng Xue, Xiangnan He, Xiang Wang, Jiandong Xu, Kai Liu, and Richang Hong. 2019. Deep Item-based Collaborative Filtering for Top-N Recommendation. *ACM Transactions on Information Systems (TOIS)* 37, 3 (2019), 33.

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 5.

Yongfeng Zhang and Xu Chen. 2018. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192* (2018).

APPENDIX

Hyperparameter Settings: For training purposes, we use the same model settings for DeepICF as mentioned in the paper [Xue et al. 2019] (or the Github implementation[4]), our port of the code is made available at: https://github.com/wakeuprj/DeepICF. Hyperparameters for replication studies are:

- Embedding size: 16
- Multi-Layer-Perceptron layers: 64, 32, 16
- Alpha (α): 0
- Beta (β): 0.8
- Learning Rate: 0.01
- Pretrain: False

Calibration plots: We plot the calibration curve for the positive and the negative test cases separately. We do this to focus more on the positive interactions than the negative ones (especially because the negative interactions are randomly sampled out of all the remaining un-interacted items).

For instance, for a case with target label t and prediction for the positive label as p (note that since we are using softmax activation, the prediction of the negative label would be $1 - p$). If the target label is positive and $p > 0.5$, the test case is classified as *correct* while if it is negative, it is classified as *incorrect*. Similarly, if the target label is negative and $p < 0.5$, the test is classified as *correct*, otherwise *incorrect*. When $p < 0.5$, $1 - p$ is used as the value to be bucketed. Note that there is a subtle but crucial difference between this approach and the general approach mentioned in [Hamill 1997].

[4] https://github.com/linzh92/DeepICF