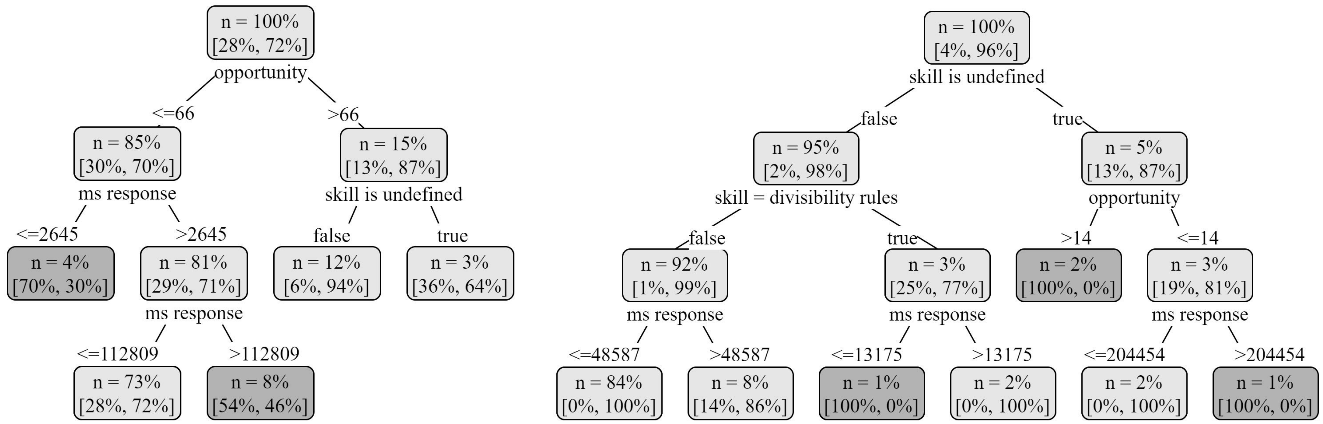

(a) Baseline (compatibility = 88%, AUC = 54%)

(b) Personalized (compatibility = 94%, AUC = 62%)

Figure 5: Baseline and personalized decision trees (up to a depth of 3) for a target user in the ASSISTments dataset, produced by the CART algorithm. Each node includes the percentage of training samples at the respective node and the distribution over the target class weighted according to the sample weights (the shade indicates the majority class). The performance and compatibility measures in the user's test set are indicated.

sification task is to predict whether a student will solve a math problem correctly or not. The baseline model and the personalized model for the target user are shown in Figure 5. The model that provided the best tradeoffs in average on this user's validation sets was m_5, which also happened to achieve a significantly better performance-compatibility tradeoff than m_0 (the baseline model) on the user's test set.

A significant difference is that the personalized model uses the fact that the student lacks knowledge regarding some skills ("divisibility rules" and "least common multiple") as the student tends to choose the wrong answer in questions involving these skills, while the baseline model does not consider any particular skill.

An important trend that is exhibited in the dataset is that if a student's response time is less than $\sim$3000 milliseconds the answer is very likely to be incorrect. According to an ASSISTments domain expert, a possible reason for this is that a short response time indicates that the student has guessed the answer. The baseline model capitalizes on this trend, significantly basing its decisions on this threshold. On the other hand, the personalized model did not split on this variable, possibly reflecting the fact that it's less common for the target user to guess an answer.

The "opportunity" variable indicates how many questions involving the same skill the student has answered in the past. According to domain experts, high opportunity values along with incorrect answers reflect students who are "gaming" the system and answering arbitrarily, since the number of opportunities stacks up with each answer given to questions involving the same skill regardless of its correctness. The personalized model predicts the answer will be incorrect given an opportunity count greater than 14 on questions where the skill involved is not defined, hinting that this student may be a "gamer" when it comes to this kind of questions. The baseline model does not check for this trend, possibly indicating that most students are not "gamers".

Next we analyze a use case for the MOOC dataset, where the classification task is to predict whether a forum post by a student reflects a high level of confusion or not. Please refer to the supplementary material to see the corresponding decision trees. The personalized model significantly outperformed the baseline model in terms of AUC (97% vs 68%) when being 100% compatible with the pre-update model. In this case the best performing model in the user's validation sets was m_3, a model that gives a relatively low importance to non-target users.

The most important difference between the baseline model and the personalized model is that the personalized model considers whether the forum post contains many words of the type "insight" and "cause", which are words like "think", "know" and "because". The fact that a post includes many of these words indicates that the student may be explaining something rather than posting a question that expresses confusion. The personalized model benefited from making this distinction, which it found useful apparently because the target student commonly gives explanations to other students.

Another difference between the two models is that the the personalized model's decision tree branches according to the amount of words in the category "tentative", which includes words like "maybe" and "perhaps". In the cases where the student included many of these words, the forum post was likely to indicate confusion. The baseline tree doesn't branch by this variable, possibly indicating that this student hypothesizes answers to his/her own questions more often than the average user.

Lastly, the personalized model's tree checks for whether the post includes many words from the "friend" category, which are words like "buddy" and "neighbor". It found that the post is very likely not showing confusion if it contains many of these words, which may indicate that this student often engages in social conversations.

Related Work

This paper builds on the recent work of **?** that introduces the idea of the compatibility score of an update (Equation 1) and proposes a method for increasing this score by employing a customized loss function (Equation 2) where an additional weighted penalty is given for newly introduced errors (mistakes that the model prior to the update didn't make). They showed that forcing the update to be more compatible generally decreases its performance, i.e. a performance-compatibility tradeoff. We expand this method by adding the notion of personalization towards target users with the goal of improving the performance-compatibility tradeoff provided by the update. It could be interesting to explore the theoretical similarities between these two methods, since model ensemble enjoys a vast theoretical framework (**?**).

Choosing the best model for each user is related to research on methods for choosing the best expert (**?**), but in our work we simply consider the quality of the performance-compatibility tradeoffs (in terms of AUTC) provided by the various models on a validation set to determine this. Further implementation of the ideas proposed in that research may improve the reliability of this selection.

Several other works relate to the personalization of AI-models to users but do not address the personalization of updates to these systems, let alone the notion of an update's compatibility with the prior model. For instance, for the ASSISTment dataset mentioned in previous sections, work was performed on individualizing student models (**??**) and on clustering the students (**??**) with the goal of improving the accuracy of the predictions.

Much work has been done in the field of human-AI interactions. The compatibility of an update to an AI-system is closely related to the 14th Guideline for Human-AI Interactions from Amershi et al.'s work (**?**) described as *"Update and adapt cautiously: Limit disruptive changes when updating and adapting the AI-systems behaviors"*. In our case, this means making sure that the predictions made by the updated model conform to the user's expectations that developed prior to the update. It is related also to the 5th step in an article from Google Design (**?**) that states the importance of making sure that the AI-system and the user's model evolve in tandem. For more references to related work on human-AI interaction and the field of AI-advised human decision making refer to the related work section in the paper of **?**.

Conclusion

The compatibility of an update to an AI-system with the system prior to the update is important for the adequate functioning of human-AI teams (**??**). Previous work addressed the problem of increasing compatibility by developing a loss function that delivers an additional penalty for newly introduced errors (mistakes that the model prior to the update didnt make), and showed that there's a tradeoff between the compatibility and performance of the updated model (**?**). We extended this approach by personalizing the model's objective function to target users with the goal of producing improving this tradeoff. We also proposed a framework for selecting the best way of performing this personalization.

The experimental results showed that our personalization approach can yield significantly better performance-compatibility tradeoffs than the baseline non-personalized model. We then analyzed two use cases where the personalization exceptionally outperformed the baseline and showed that the personalized classifier model differed fundamentally from the baseline model.

Our approach is limited in the sense that it assumes that the user's future interactions with the system will resemble the ones observed so far. In future work we will address this limitation, and explore ways of modifying the objective function beyond simply assigning weights to the dataset samples possibly by employing program synthesis or inverse reinforcement learning methods. We believe that informing users about the performance-compatibility tradeoff of the models that are used to interact with them can contribute on making AI-systems more transparent.

Acknowledgements

Thanks very much to Avi Segal and Nicholas Hoernle for helpful comments. This research was partially supported by Israeli Science Foundation (ISF) Grant No. 773/16 and by Canada's CIFAR AI Chairs program.

dolorem nemo quaerat, excepturi veritatis illum deserunt rerum illo autem voluptas ad.Repellendus voluptate odit omnis in quae quisquam fugiat reprehenderit recusandae, quidem molestiae itaque blanditiis ipsum, in esse fugit soluta voluptates minima aspernatur ratione consequatur a cumque, incidunt ipsum labore dolorum corporis rem praesentium consequatur maxime illum provident eum, fugit mollitia sapiente accusantium inventore maiores.Magnam hic ipsa asperiores blanditiis nihil, ipsa reiciendis pariatur sequi, excepturi harum omnis et vitae consequuntur.Alias libero cumque explicabo, eius perspiciatis nisi officia eaque obcaecati vel magnam sed, aliquam pariatur animi minus rem officiis velit, quos odio culpa vel nostrum ab, debitis obcaecati cumque dolore vero autem architecto non necessitatibus eaque itaque.Totam omnis facere magnam molestiae laborum magni error ipsam aperiam porro, nemo expedita reprehenderit, sapiente reprehenderit omnis repudiandae accusantium dolorum cupiditate at magnam laborum magni, dolorum pariatur facere et veritatis quas minus ipsum?Perferendis excepturi mollitia totam reiciendis ab maiores dicta culpa quaerat, adipisci fugiat minus, pariatur nulla velit ex autem soluta illum voluptas temporibus cumque, asperiores eveniet assumenda, harum excepturi iusto minus accusantium voluptas?Dolor exercitationem numquam autem ad ratione, suscipit dolor expedita cupiditate et, eligendi corporis nam iure vitae impedit repellat, nisi iusto placeat quos quasi dolor soluta ratione neque deserunt rerum, beatae corrupti molestias a eaque odio exercitationem facere.