Improving the Performance-Compatibility Tradeoff with Personalized Objective Functions (Student Abstract)

Jonathan Martinez
Ben-Gurion University, Israel
martijon@post.bgu.ac.il

Abstract

AI-systems that model and interact with their users can update their models over time to reflect new information and changes in the environment. Although these updates may improve the overall performance of the AI-system, they may actually hurt the performance with respect to individual users. Prior work has studied the tradeoff between improving the system's performance following an update and the compatibility of the updated system with prior user experience. The more the model is forced to be compatible with a prior version, the higher loss in performance it will incur. This paper shows that that by personalizing the loss function to specific users, it is possible to increase the prediction performance of the AI-system while sacrificing less compatibility for these users following an update to improve the system's performance. Our approach updates the sample weights to reflect their contribution to the compatibility of the model for a particular user following the update. We construct a portfolio of different models that vary in how they personalize the loss function for a target user and then select the best model to use based on a validation set. We use a model selection algorithm to choose the best model from the portfolio for each user given a set of features that reflect the users' characteristics and performance of the different models on a training set. We apply this approach to three supervised learning tasks commonly used in the human-computer decision-making literature. We show that using our approach leads to significant improvements in the performance-compatibility tradeoff over the non-personalized approach of Bansal et al., achieving up to 300% improvement for certain users.

Introduction

As the user interacts with an AI-system, two processes occur. First, the user develops a mental model of the system's capabilities based on the quality of its predictions/recommendations. Second, the system collects more data and is able to update its prediction model. While updating the system's model can improve its performance, it can also change the way the system makes predictions in a way that does not agree with the user's expectations, based on its past interactions with the system. Thus while the update improves the overall system performance, it may exhibit a poor compatibility with the user's expectations (Bansal et al. 2019b), possibly causing the user to lose trust in the system and start ignoring its predictions/recommendations.

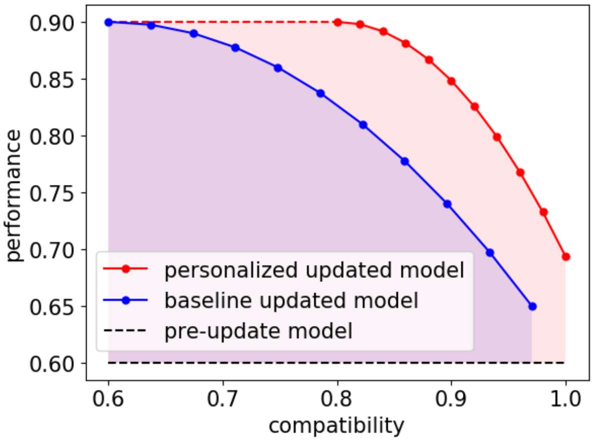

Figure 1: Synthetic example of performance-compatibility tradeoff curves. The x-axis represents the model's compatibility score and the y-axis its performance.

Prior Work

Let h_1 be the model prior to the update and h_2 the model following the update, such that h_1 is trained only with a subset of the training set of h_2. A *newly introduced error* is an error that the updated model h_2 makes that the pre-update model h_1 didn't make. The compatibility of an update to a classifier measures the amount of newly introduced errors that are introduced by the updated model h_2 on some dataset D (Bansal et al. 2019b):

$$C(h_1, h_2, D) = \frac{\sum_{i=1}^{|D|} \mathbb{1}[h_1(x_i) = h_2(x_i) = y_i]}{\sum_{i=1}^{|D|} \mathbb{1}[h_1(x_i) = y_i]} \quad (1)$$

Bansal et al. (2019b) propose a way of modifying a loss function L (e.g., Cross-Entropy Loss) such that the amount of penalty given for the predicted label of an instance depends on whether it corresponds to a newly introduced error:

$$\begin{aligned} L_c(x) = &(1-\lambda) \cdot L(x) \\ &+ \lambda \cdot L(x) \cdot \mathbb{1}[h_1(x) = y] \end{aligned} \quad (2)$$

model	W_k			
	w_0	w_1	w_2	w_3
m_0	1	0	1	0
m_1	0	1	0	1
m_2	0	1	1	0
m_3	0	1	1	1
m_4	1	0	0	1
m_5	1	1	0	1
m_6	1	0	1	1
m_7	1	1	1	0
m_8	1	1	1	1

Table 1: The portfolio of models M.

Where $L(x)$ is the penalty given to a an updated model for the label it predicts for a sample x. As the value of the parameter λ increases, so does the penalty for newly introduced errors. This simultaneously increases the compatibility score of the updated model h_2 and tends to decrease the model's performance as it is forced to make predictions that are more similar to those made by the pre-update model h_1.

Personalizing Objective Functions

Our hypothesis is that personalizing the objective function for particular users is likely to improve the performance-compatibility tradeoff provided to that user following the model update. Let $D^c \subseteq D$ be the subset of samples for which h_1 is correct. Errors that h_2 makes on samples in D^c decrease its compatibility score. Given a dataset D and a pre-update model h_1 used to determine D^c, the weight of each sample $x \in D$ can be assigned by the following function:

$$w_c(x, D, \lambda) = (1 - \lambda) \cdot \mathbb{1}[x \in D] + \lambda \cdot \mathbb{1}[x \in D^c] \quad (3)$$

Weighting samples by Equation 3 and training a model with a regular loss function is equivalent to training the model with the modified loss function from Equation 2. Now, we extend this weight function to *personalize* the objective function for particular users. Let $D_i \subseteq D$ be the set of samples corresponding to the history of interaction between user i and the AI-system, and $D_i^c \subseteq D_i$ be the subset of those samples such that the pre-update hypothesis h_1 is correct. The objective function is personalized to a target user by assigning a weight to each sample in the dataset according to the following. Let $W = (w_0, w_1, w_2, w_3)$ be a set of four weights such that each weight captures the impact that each set of samples D, D_i, D^c and D_i^c (respectively) will have on the updated model's compatibility in respect to a target user i. Each combination of weights W represents a different approach or degree of personalization. Given a dataset D, a target user i, a pre-update model h_1 used to determine D^c and D_i^c and a vector W we assign the weight of each sample $x \in D$ by the following function:

$$\begin{aligned} & w(x, i, D, W, \lambda) = \\ & (1 - \lambda) \cdot (w_0 \cdot \mathbb{1}[x \in D] + w_1 \cdot \mathbb{1}[x \in D_i]) \quad (4) \\ & + \lambda \cdot (w_2 \cdot \mathbb{1}[x \in D^c] + w_3 \cdot \mathbb{1}[x \in D_i^c]) \end{aligned}$$

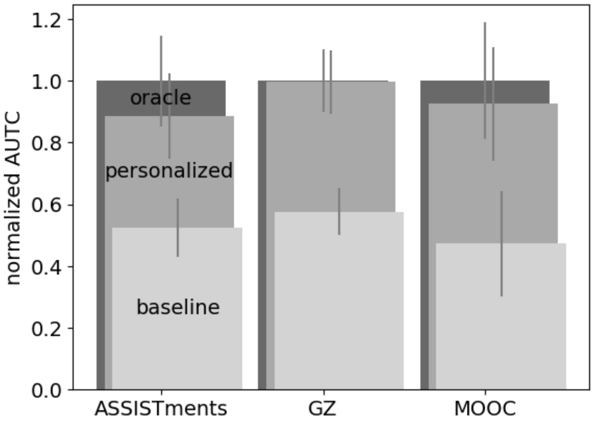

Figure 2: Performance of the baseline, personalized and oracle models on the different datasets.

In this work we consider only binary weights (0 or 1), resulting in the family of models shown in Table 1. Each model represents a different approach to personalization. In particular, the model m_0 does not differentiate between users and is equivalent to Equation 2. This is the model of Bansal et al. (2019a). In contrast, the objective function of m_1 represents an extreme approach to personalization where only the samples that belong to the user's history ($x \in D_i$) are considered.

Empirical Results

Fig. 2 shows results averaged over a set of around 100 users on three commonly used datasets that register the interactions of real-world users. The height of each bar indicates the area under the tradeoff curve (Fig. 1) of each model normalized to that of an oracle model that always chooses the model with the highest performance on each user's test set. As indicated by the standard deviation, the personalized model's performance is not statistically different than the oracle's, meaning that it approximates the optimal performance. However, the difference between the personalized model and the baseline is statistically significant (obtained a p-value below 0.05 on a paired t-test between the two models, on all three datasets).

References

Bansal, G.; Nushi, B.; Kamar, E.; Lasecki, W. S.; Weld, D. S.; and Horvitz, E. 2019a. Beyond Accuracy: The Role of Mental Models in Human-AI Team Performance. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, 2–11.

Bansal, G.; Nushi, B.; Kamar, E.; Weld, D. S.; Lasecki, W. S.; and Horvitz, E. 2019b. Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 2429–2437.