
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Daeë, Pedram; Peltola, Tomi; Vehtari, Aki; Kaski, Samuel
User Modelling for Avoiding Overfitting in Interactive Knowledge Elicitation for Prediction

Published in:
IUI 2018 - Proceedings of the 23rd International Conference on Intelligent User Interfaces

DOI:
[10.1145/3172944.3172989](https://doi.org/10.1145/3172944.3172989)

Published: 08/03/2018

Document Version
Early version, also known as pre-print

Please cite the original version:
Daeë, P., Peltola, T., Vehtari, A., & Kaski, S. (2018). User Modelling for Avoiding Overfitting in Interactive Knowledge Elicitation for Prediction. In IUI 2018 - Proceedings of the 23rd International Conference on Intelligent User Interfaces (pp. 305-310). Association for Computing Machinery (ACM).
<https://doi.org/10.1145/3172944.3172989>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

User Modelling for Avoiding Overfitting in Interactive Knowledge Elicitation for Prediction *

Pedram Daei[†], Tomi Peltola[†], Aki Vehtari, and Samuel Kaski

Helsinki Institute for Information Technology HIIT,
Department of Computer Science, Aalto University
`firstname.lastname@aalto.fi`

[†]Authors contributed equally.

Abstract

In human-in-the-loop machine learning, the user provides information beyond that in the training data. Many algorithms and user interfaces have been designed to optimize and facilitate this human-machine interaction; however, fewer studies have addressed the potential defects the designs can cause. Effective interaction often requires exposing the user to the training data or its statistics. The design of the system is then critical, as this can lead to double use of data and overfitting, if the user reinforces noisy patterns in the data. We propose a user modelling methodology, by assuming simple rational behaviour, to correct the problem. We show, in a user study with 48 participants, that the method improves predictive performance in a sparse linear regression sentiment analysis task, where graded user knowledge on feature relevance is elicited. We believe that the key idea of inferring user knowledge with probabilistic user models has general applicability in guarding against overfitting and improving interactive machine learning.

1 INTRODUCTION

Interactive and human-in-the-loop machine learning (and the related field of visual analytics) exploits the complementary knowledge and skills of humans and machines to improve performance over automatic training-data-based machine learning and to extend the reach of machine learning systems [1–3, 8, 13, 17]. However, the study of how to optimally combine the strengths of humans and machines is still in its early phase, and many possible issues arising from the interaction have not been thoroughly considered yet.

Overfitting, that is, the model fitting to idiosyncrasies in the training data and hence not generalizing to new data, is a thoroughly studied topic in automatic machine learning. Subtle issues with regard to overfitting can arise when introducing a human into the loop. Yet the risk of overfitting seems little discussed in the interactive machine learning literature, although many methods combine user interaction with training-data-based machine learning models. For example, many methods visualize statistics of training data or machine output directly for the user [1, 10, 11, 13, 14, 20, 22] or use the training data to select informative queries to present for

*This is the pre-print version. The paper is published in the proceedings of *IUI 2018* conference. Definitive version DOI: <https://doi.org/10.1145/3172944.3172989>.

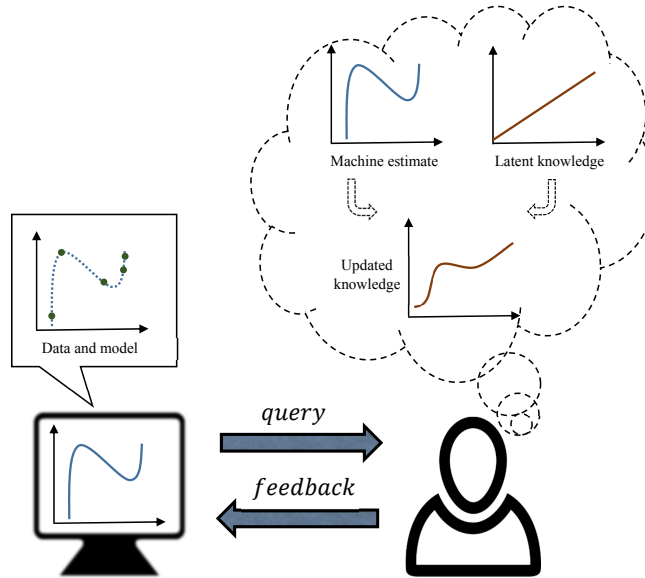


Figure 1: A schematic example of overfitting in human-in-the-loop machine learning. The machine and user are collaborating to improve a regression problem. The machine fits a regression model to the training data (green dots) and employs it to interact with the user (here by visualizing the trained model). Through the interaction, the user learns aspects of the training data and considers them to update her latent knowledge and to form her feedback. This creates a dependency between the feedback and training data that needs to be accounted for in the model to avoid double use of data and overfitting.

the user (e.g., active learning) [3, 18, 19]. Some methods use *validation* datasets in addition to the training set to evaluate the performance. This can help to alleviate overfitting, but if the user can interact with the model based on the results on the validation, the validation set is effectively only another training dataset. A parallel line of research into controlling for overfitting has spawned in adaptive data analysis [4, 16], which has a related but different goal from interactive machine learning, of exploring the data for interesting hypotheses.

In interactive machine learning, we note that, in particular, the following three steps can lead to overfitting and hence decrease in the performance of the model, as they violate the assumption of independence of the feedback and the training data: (1) showing the training data or some of its statistics to the user, (2) querying the user for feedback, and (3) inputting the feedback back to the model as independent data (a common assumption in machine learning models). Figure 1 illustrates the induced user behaviour producing a dependency between the feedback and training data. Overfitting can also happen if the user controls some preferences (e.g., cost function weighting or regularization) in the model based on the training or validation data, since the effected improvement in the fit to the training or validation data will not necessarily generalize. The more freedom the user has (or, in this sense, the richer the feedback is), the more problematic this can be.

Rather than tying the hands of the user, improved performance can be attained by accounting for the risk of overfitting in the design of the interactive machine learning system. We propose a user modelling approach in probabilistic models to infer the user’s knowledge that is complementary

to the training data. We assume that the user behaves rationally, in a simple Bayesian sense [6], in combining her latent knowledge with the information that the machine reveals of the training data. Given the observed user feedback, we then invert the process to infer the latent user knowledge and use it to update the model. We illustrate the approach in a proof-of-concept user study in a sparse linear regression prediction task, where graded feature relevance knowledge is elicited from the users.

2 METHOD

2.1 Overview

We consider human–machine interaction in probabilistic models. In particular, we consider a situation, where the machine estimates a probabilistic model from training data, and then asks the user for feedback for the learned model. The feedback is included as further data in the model. This procedure can be iterated, although in the specific implementation here, we only consider a single round.

The setup is as follows. The machine has a probabilistic model for a prediction task and a set of training data that it uses to fit the parameters of the model. The machine assumes that the user has knowledge about some aspect of the task and elicits user knowledge by displaying the result of learning from the training data and asking for feedback. This kind of interaction is common in interactive machine learning. If the system naively uses this user feedback to update the model, for example, including it as an observation that is assumed independent of the training data, it risks double use of the data and overfitting, because such assumption would clearly be invalid. On the other hand, building a feedback model that would adequately describe the dependency of the feedback on the training data can be difficult.

We instead propose to infer the latent (unobserved) user knowledge, representing information that the user has beyond that of the training data, from the observed user feedback. This inferred knowledge can then be used to update the machine’s model without double use of the training data. To make this feasible, we assume that the user behaves rationally, using the Bayes theorem, in integrating the information sources (her knowledge beyond training data and the information the machine provided). We then invert the process to infer the latent user knowledge.

2.2 General Mathematical Formulation

Let the observation model of a training dataset \mathcal{D} be $p(\mathcal{D} | \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a vector containing the model parameters, and $p(\boldsymbol{\theta})$ be their prior distribution. Given the model and the training dataset, the machine computes the posterior distribution $p(\boldsymbol{\theta} | \mathcal{D})$ of the parameters using the Bayes theorem. We assume the user has knowledge about some parameter or statistic ϕ , which is an element in $\boldsymbol{\theta}$ or, more generally, a function of $\boldsymbol{\theta}$. The machine provides the user with information on ϕ , for example, its posterior distribution $p(\phi | \mathcal{D})$ and asks the user to provide feedback on it.

Let f be the latent (unobserved) user knowledge. Our goal is to infer from the observed feedback a latent feedback likelihood function $p(f | \phi)$ that can be used to update the model to $p(\boldsymbol{\theta} | \mathcal{D}, f)$. By the assumption of a rational user, the observed feedback is based on the posterior distribution

$$p(\phi | \mathcal{D}, f) = \frac{p(f | \phi)p(\phi | \mathcal{D})}{p(f | \mathcal{D})}, \quad (1)$$

where $p(f | \mathcal{D}) = \int p(f | \phi)p(\phi | \mathcal{D})d\phi$ is the normalization constant. The technical details on how to invert this to learn $p(f | \phi)$ are case-specific. In the next section, we will show how to do this in eliciting feature relevance for sparse linear regression.

2.3 Feature Relevance Elicitation in Sparse Linear Regression

We apply the approach to infer user knowledge on feature relevance in sparse linear regression. We use a probabilistic sparse linear regression model described in [3], which formulates a linear model to predict the target variable y given a vector of features \mathbf{x} and uses a spike-and-slab prior to model whether features are included or excluded from the regression:

$$\begin{aligned} y_i &\sim \text{N}(\mathbf{x}_i^\top \mathbf{w}, \sigma^2), \\ \sigma^{-2} &\sim \text{Gamma}(\alpha_\sigma, \beta_\sigma), \\ w_j &\sim \gamma_j \text{N}(0, \tau^2) + (1 - \gamma_j) \delta_0, \\ \gamma_j &\sim \text{Bernoulli}(\rho), \end{aligned}$$

where $i = 1, \dots, N$ runs over N training samples $(y_i, \mathbf{x}_i) \in \mathcal{D}$, $j = 1, \dots, M$ runs over M features, σ^2 is a residual variance parameter, the w_j are regression weights, and the γ_j are binary variables indicating whether feature j is included in the regression ($\gamma_j = 1$: w_j is a priori normally distributed with variance τ^2) or excluded ($\gamma_j = 0$: $w_j = 0$ via the point mass δ_0). The parameter ρ is the prior expected proportion of included variables. The probabilistic parameters of the model are $\boldsymbol{\theta} = (\mathbf{w}, \sigma^2, \boldsymbol{\gamma})$; here, α_σ , β_σ , τ^2 , and ρ are fixed hyperparameters.

To elicit feedback on feature relevance, we show the marginal posterior probability $p(\gamma_j = 1 \mid \mathcal{D})$ to the user and ask her to provide as feedback her estimate of the probability of the feature being relevant for the prediction. To infer the latent user knowledge, we assume that the user’s feedback is the posterior probability

$$p(\gamma_j = 1 \mid \mathcal{D}, f_j) = \frac{p(f_j \mid \gamma_j = 1)p(\gamma_j = 1 \mid \mathcal{D})}{Z}, \quad (2)$$

where Z is the normalization constant and the latent feedback likelihood is

$$p(f_j \mid \gamma_j) = A_{f_j} \gamma_j + B_{f_j} (1 - \gamma_j),$$

where A_{f_j} is the likelihood for the latent f_j when $\gamma_j = 1$ and B_{f_j} when $\gamma_j = 0$. Without loss of generality (for using the likelihoods for updating the model later), we can set $A_{f_j} + B_{f_j} = 1$, with $A_{f_j} \in (0, 1)$ and $B_{f_j} \in (0, 1)$.

We infer A_{f_j} (and, consequently, $B_{f_j} = 1 - A_{f_j}$) by solving from the Bayes theorem in Equation 2:

$$A_{f_j} \propto \frac{p(\gamma_j = 1 \mid \mathcal{D}, f_j)}{p(\gamma_j = 1 \mid \mathcal{D})},$$

where the numerator is the observed feedback given by the user and the denominator is the machine’s posterior probability that was shown to the user.

Given a set of observed feedbacks \mathcal{F} from the user, we update the model to $p(\boldsymbol{\theta} \mid \mathcal{D}, \mathcal{F})$, where $\boldsymbol{\theta}$ denotes all parameters of the model, by updating $p(\boldsymbol{\theta} \mid \mathcal{D})$ using the Bayes theorem and the inferred likelihood functions $p(f_j \mid \gamma_j)$ for each feature j with observed feedback.

Computation of the posterior distribution is intractable. Expectation propagation is used to approximate it (see [3]).

3 EXPERIMENT AND RESULTS

3.1 Sentiment Analysis Task

We considered the problem of rating prediction from user reviews on Amazon kitchen products. The task and data were previously studied in [3, 7]. The data consist of review texts, represented

<i>Condition</i>	Test MSE \pm STD		
	<i>No feedback</i>	<i>User feedback</i>	<i>User model</i>
Baseline	1.835	1.749 \pm 0.050	NA
IE	1.835	1.744 \pm 0.045	1.705 \pm 0.038

Table 1: Mean and standard deviation (STD) of MSE on test data for the two systems in different conditions.

as bag-of-words with 824 distinct unigram and bigram keywords, and their corresponding 1–5 star ratings. The machine learning system aims to predict the ratings of new reviews (test data), given some training data and external expert knowledge. To make the prediction challenging, the data set was randomly partitioned in 500 training data and 4649 test data (the number of training data is smaller than the number of dimensions).

The goal of the experiment was introduced to the participants as eliciting domain knowledge from people to help in designing better predictors of product ratings. However, the real research question was to investigate the efficiency of the user model and user interaction in different scenarios. We listed 70 keywords from the reviews and asked the participants to judge the probability of relevance of each keyword in predicting the product rating. The participants could provide feedback by adjusting the probability value between 0 (not-relevant at all) to 1 (absolutely relevant) by moving a slider. The feedback was only recorded after the slider was moved. The participants could skip giving feedback to keywords that they were very uncertain about. The task description provided example keywords *must buy*, *disaster*, and *is* and explained that the first two provide useful information about the product rating, and therefore, they are relevant while the latter is uninformative for rating prediction.

Two systems were implemented, a baseline system where the initial position of sliders were set to the default value 0, and an interactive elicitation (IE) system where the initial positions were set by the machine to the posterior inclusion probabilities $p(\gamma_j = 1 \mid \mathcal{D})$ based on the training data. The participants were informed about the initialization method. Both systems use the prediction model and the feedback likelihoods introduced in the previous section with the difference that IE can infer the latent user knowledge based on the proposed user model while the baseline directly applies the user feedback in the model. Following [3], the model hyperparameters were set as $\alpha_\sigma = 1$, $\beta_\sigma = 1$, $\rho = 0.3$, and $\tau^2 = 0.01$. Mean squared error (MSE) on test data is used as the performance measure. The code, data, and experiment forms can be found in <https://github.com/HIIT/human-overfitting-in-IML>.

3.2 Results

48 university students and researchers participated in the user study, 3 were excluded since they either left more than 2/3 of the questions unanswered or they finished the study in less than 3 minutes (it was unrealistic to go through the form in less than 3 minutes). 23 participants used the baseline and the remaining 22 the IE system.

Table 1 shows the average test MSE for participants of the two systems before and after receiving feedback and also after the correction done by the proposed user model in IE. The feedback improved the predictive performance in both systems (p-value= 3×10^{-8} in baseline and p-value= 5×10^{-9} in IE without user model, using paired-sample *t*-test). This shows, as analogously claimed in [3], that the participants, on average, have the necessary knowledge to improve the prediction. More prominently, the predictive performance further improved in the IE system after inferring the latent user knowledge using the user model (p-value= 7×10^{-7} in

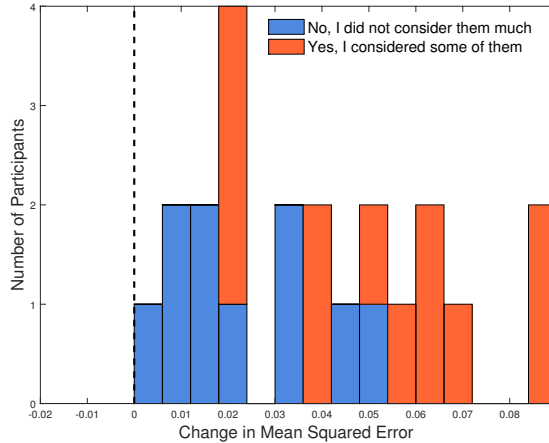


Figure 2: Stacked histogram of MSE change (the difference between MSE after directly using the feedbacks and MSE after the correction done by the propose user model) for 22 participants in IE system. The participants are grouped based on their answer to the question on whether they found the machine estimates useful or not. User modelling has improved MSE values for all participants.

paired-sample *t*-test between directly using the feedbacks and after employing the user model). Moreover, the user model improved the predictions for each individual user (Figure 2). In a post-questionnaire, we asked the participants of the IE system about the usefulness of the machine estimates: 12 participants answered that they considered them when giving some of the answers and the remaining 10 responded that they did not consider machine estimates that much. On average, the predictions of the former group improved more with the user model (Figure 2).

The feedback the participants gave to the keywords differed between the two systems in several aspects. Table 2 lists average feedback values and machine estimates for 10 keywords with the lowest nominal p-values (two-sample *t*-test without assuming equal variances between the two systems), showing the keywords that were the most different between the systems. The average correlation to machine estimates for users in IE system was 0.46 (0.33 for the baseline system) and the average variance of feedbacks on keywords was 0.043 (0.060 for the baseline system). These support the hypothesis that the participants considered the machine estimates in the IE system. In the IE system, the average correlation to machine estimates after the correction done by the user model declined to -0.017 which suggests that the user model was successful in reducing the dependency to the training data.

4 DISCUSSION AND CONCLUSION

We described a user modelling methodology in probabilistic models for disentangling latent user knowledge from observed user feedback that was given in response to machine revealing information from the training data. We used this to guard against double use of the training data and overfitting in interactive machine learning. The proof-of-concept user study in interactive knowledge elicitation of feature relevance information in sparse linear regression showed the potential of the approach for improving prediction performance.

<i>Keyword</i>	<i>Machine</i>	<i>IE</i>	<i>Baseline</i>	<i>P-value</i>
best	0.89	0.90	0.80	0.014
great	1.00	0.95	0.83	0.031
disappointed	0.99	0.96	0.85	0.038
heavy	0.20	0.36	0.52	0.038
buy this	0.70	0.77	0.63	0.053
steel	0.19	0.12	0.24	0.073
line	0.34	0.15	0.06	0.088
don't	0.47	0.54	0.38	0.090
recommend	0.16	0.74	0.84	0.102
good	0.26	0.71	0.81	0.115

Table 2: Difference between user feedback in the baseline and IE systems (without user model).

Our approach is based on a simple rationality assumption of the user. It is, however, unreasonable to assume that users would in general behave completely rationally. In particular, the amount information and the way it is presented to the user are important factors that should be considered in designing interactive machine learning systems and user models. In our experiment, the interaction was based on reporting probability values through sliders. Such probability elicitation can be prone to the anchoring effect [5, 21]. Arguably, similar psychological mechanisms will appear in many interactive machine learning applications, since the machine often needs to guide the user for efficient interaction. This paper demonstrates that as long as the user modelling is able to capture the main sources of defects in the interaction, it would be expected to improve the results.

We believe that, as the field of interactive machine learning matures, more consideration will be put into the possible defects in the human–machine interaction. In particular, better user modelling will allow the user to behave naturally in providing feedback to the machine, while the machine (that is, the design of the machine learning system and models) will account for the inevitable human factors and biases [2, 5, 9, 12, 15, 17, 21] to optimally combine the complementary knowledge and skills of the user and the machine. Our work is a step towards this with regard to overfitting in the interaction.

5 ACKNOWLEDGMENTS

This work was financially supported by the Academy of Finland (Finnish Center of Excellence in Computational Inference Research COIN; Grants 295503, 294238, 292334, and 284642), Re:Know funded by TEKES. We acknowledge the computational resources provided by the Aalto Science-IT Project. We thank Marta Soare for collaboration and helpful comments in early stage of the project.

References

- [1] Hodayun Afrabandpey, Tomi Peltola, and Samuel Kaski. 2017. Interactive Prior Elicitation of Feature Similarities for Small Sample Size Prediction. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP '17)*. ACM, New York, NY, USA, 265–269. DOI:<http://dx.doi.org/10.1145/3079628.3079698>

- [2] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35, 4 (2014), 105–120.
- [3] Pedram Daei, Tomi Peltola, Marta Soare, and Samuel Kaski. 2017. Knowledge elicitation via sequential probabilistic inference for high-dimensional prediction. *Machine Learning* 106, 9 (2017), 1599–1620. DOI:<http://dx.doi.org/10.1007/s10994-017-5651-7>
- [4] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. 2015. The reusable holdout: Preserving validity in adaptive data analysis. *Science* 349, 6248 (2015), 636–638.
- [5] Paul H Garthwaite, Joseph B Kadane, and Anthony O’Hagan. 2005. Statistical Methods for Eliciting Probability Distributions. *J. Amer. Statist. Assoc.* 100, 470 (2005), 680–701. DOI: <http://dx.doi.org/10.1198/016214505000000105>
- [6] Samuel J Gershman, Eric J Horvitz, and Joshua B Tenenbaum. 2015. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science* 349, 6245 (2015), 273–278.
- [7] José Miguel Hernández-Lobato, Daniel Hernández-Lobato, and Alberto Suárez. 2015. Expectation propagation in linear regression models with spike-and-slab priors. *Machine Learning* 99, 3 (01 Jun 2015), 437–487. DOI:<http://dx.doi.org/10.1007/s10994-014-5475-7>
- [8] Andreas Holzinger. 2016. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics* 3, 2 (01 Jun 2016), 119–131. DOI: <http://dx.doi.org/10.1007/s40708-016-0042-6>
- [9] Tzu-Kuo Huang, Lihong Li, Ara Vartanian, Saleema Amershi, and Xiaojin Zhu. 2016. Active Learning with Oracle Epiphany. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 2820–2828. <http://papers.nips.cc/paper/6155-active-learning-with-oracle-epiphany.pdf>
- [10] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. 2010. Interactive Optimization for Steering Machine Classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’10)*. ACM, New York, NY, USA, 1343–1352. DOI: <http://dx.doi.org/10.1145/1753326.1753529>
- [11] Josua Krause, Adam Perer, and Enrico Bertini. 2014. INFUSE: interactive feature selection for predictive modeling of high dimensional data. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 1614–1623.
- [12] Todd Kulesza, Saleema Amershi, Rich Caruana, Danyel Fisher, and Denis Charles. 2014. Structured Labeling for Facilitating Concept Evolution in Machine Learning. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI ’14)*. ACM, New York, NY, USA, 3075–3084. DOI:<http://dx.doi.org/10.1145/2556288.2557238>
- [13] Luana Micalef, Iris Sundin, Pekka Marttinen, Muhammad Ammad-ud din, Tomi Peltola, Marta Soare, Giulio Jacucci, and Samuel Kaski. 2017. Interactive Elicitation of Knowledge on Feature Relevance Improves Predictions in Small Data Sets. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI ’17)*. ACM, New York, NY, USA, 547–552. DOI:<http://dx.doi.org/10.1145/3025171.3025181>

- [14] Thomas Mühlbacher and Harald Piringer. 2013. A partition-based framework for building and validating regression models. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 1962–1971.
- [15] Edward Newell and Derek Ruths. 2016. How One Microtask Affects Another. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3155–3166. DOI:<http://dx.doi.org/10.1145/2858036.2858490>
- [16] Daniel Russo and James Zou. 2016. Controlling bias in adaptive data analysis using information theory. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, Vol. PMLR 51. 1232–1240.
- [17] Dominik Sacha, Michael Sedlmair, Leishi Zhang, John A Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C North, and Daniel A Keim. 2017. What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing* 268 (2017), 164–175.
- [18] Burr Settles. 2010. *Active learning literature survey*. Computer sciences technical report 1648. University of Wisconsin, Madison.
- [19] Iris Sundin, Tomi Peltola, Muntasir Mamun Majumder, Pedram Daei, Marta Soare, Homayun Afrabandpey, Caroline Heckman, Samuel Kaski, and Pekka Marttinen. 2017. Improving drug sensitivity predictions in precision medicine through active expert knowledge elicitation. *arXiv preprint arXiv:1705.03290* (2017).
- [20] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S. Tan. 2009. EnsembleMatrix: Interactive Visualization to Support Machine Learning with Multiple Classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1283–1292. DOI:<http://dx.doi.org/10.1145/1518701.1518895>
- [21] Amos Tversky and Daniel Kahneman. 1974. Judgment under Uncertainty: Heuristics and Biases. *Science* 185, 4157 (1974), 1124–1131. DOI:<http://dx.doi.org/10.1126/science.185.4157.1124>
- [22] Stef Van Den Elzen and Jarke J van Wijk. 2011. BaobabView: Interactive construction and analysis of decision trees. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST)*. IEEE, 151–160.