Improving measurements of similarity judgments with machine-learning algorithms

Jeffrey R. Stevens[1], Alexis Polzkill Saltzman[1,2], Tanner Rasmussen[1], & Leen-Kiat Soh[2]

[1] Department of Psychology, Center for Brain, Biology & Behavior, University of Nebraska, Lincoln, Lincoln, NE, USA
[2] Department of Computer Science and Engineering, University of Nebraska, Lincoln, Lincoln, NE, USA

Intertemporal choices involve assessing options with different reward amounts available at different time delays. The similarity approach to intertemporal choice focuses on judging how similar amounts and delays are. Yet we do not fully understand the cognitive process of how these judgments are made. Here, we use machine-learning algorithms to predict similarity judgments to (1) investigate which algorithms best predict these judgments, (2) assess which predictors are most useful in predicting participants' judgments, and (3) determine the minimum number of judgments required to accurately predict future judgments. We applied eight algorithms to similarity judgments for reward amount and time delay made by participants in two data sets. We found that neural network, random forest, and support vector machine algorithms generated the highest out-of-sample accuracy. Though neural networks and support vector machines offer little clarity in terms of a possible process for making similarity judgments, random forest algorithms generate decision trees that can mimic the cognitive computations of human judgment-making. We also found that the numerical difference between amount values or delay values was the most important predictor of these judgments, replicating previous work. Finally, the best performing algorithms such as random forest can make highly accurate predictions of judgments with relatively small sample sizes (~15), which will help minimize the numbers of judgments required to extrapolate to new value pairs. In summary, machine-learning algorithms provide both theoretical improvements to our understanding of the cognitive computations involved in similarity judgments and intertemporal choices as well as practical improvements in designing better ways of collecting data.

Keywords: classification, intertemporal choice, judgment, machine learning, similarity

Introduction

Intertemporal choices are a critical class of decisions that involve choosing between rewards available at different times [1]. We all face these decisions on a daily basis. Would you prefer to buy the latest gadget or put that money away for retirement? Would you prefer to consume a decadent dessert or avoid the calories for a slimmer waistline? Researchers of intertemporal choice typically probe people's preferences by providing a series of choices between smaller amounts of money available after a short or no delay and a larger amount available later (e.g., Would you prefer to receive $10 today or $12 in one week?).

Though *temporal discounting* is the dominant approach to intertemporal choices [2], an alternative heuristic model asserts that *similarity judgments* can account for these choices [3, 4]. For example, if people find the reward amounts to be similar (e.g., $10 vs. $12) but the time delays to be dissimilar (e.g., today vs. one week), they may ignore the similar attribute and choose based on the dissimilar attribute (e.g., choose the immediate option). This approach predicts intertemporal choices well when it can make predictions [5], but it raises the question of what drives similarity judgments.

Previously, we used machine-learning algorithms to assess similarity judgments [6]. Machine learning is a powerful set of tools that "sift through data looking for patterns" [7]. Researchers can input *predictors* to evaluate if machine-learning algorithms can predict *responses* [8]. In our case, we were interested in which features of the amount and delay values predicted people's similarity judgments. We proposed a particular type of machine-learning algorithm [9, 10] as both a potential predictor of choice and a reasonable approximation of the cognitive process that people could use to make the similarity judgments. We found that these decision trees accurately predicted choice (about 86% out-of-sample accuracy) and that the numerical difference between the large and

small amounts and delays (large − small) and the numerical ratio between them (small / large) were the best features for predicting similarity judgments.

The aim of that study was to investigate a decision tree called Classification and Regression Tree or CART [11]. This algorithm was chosen because it was a fairly simple decision tree algorithm that is well-studied and could provide a relatively straightforward cognitive process model of decision making. Yet there are many potential machine-learning algorithms that could be used to classify similarity judgments based on the numerical values of the small and large amounts and delays. One key aim of the current study is to test a range of algorithms on our data to determine which algorithms best predict similarity judgments. In addition to *accuracy* (number of correct predictions / total number of predictions), machine learning uses other performance metrics of classification [12]. *Precision* (or positive predictive value) is the proportion of instances predicted to be positive that are actually positive (number of correct positive predictions / number of positive predictions). *Recall* (or sensitivity, hit rate, true positive rate) is the proportion of actual positives that are correctly classified (number of correct positive predictions / number of positive instances). For our purposes, we can think of "similar" judgments as positive. So precision is the proportion of similar predictions that the algorithms correctly classify as similar, and recall is the proportion of actual similar judgments that the algorithms correctly classify as similar (Table 1).

To calculate these performance metrics, we must have predictors. In Stevens and Soh [6], we mathematically arranged the small and large values to generate 11 predictors that may predict similarity judgments (Table S1). A second aim of the current study is to reassess which predictors are most useful in predicting similarity judgments using the wider range of algorithms. Further, the previous analysis only found the single best predictor for each person by extracting the predictor used as the first node in the decision tree. Here, we assess predictor *importance* [8] for each algorithm that allows this calculation. Therefore, we compute importance measures across a range of algorithms and for each predictor.

Finally, assessing similarity judgments requires asking for pairwise binary judgments of similar or dissimilar from participants. It would be useful to be able to predict an individual's similarity judgments with as few questions as possible. Therefore, our final aim is to evaluate prediction accuracy at different sample sizes to determine the minimum number of questions required to accurately predict similarity judgments using a learning-curve analysis [13]. Further, we assess whether the ordering of the questions influences prediction accuracy. Typically, when assessing the effects of sample size on accuracy, machine-learning analyses randomly select the instances within the training sets. Though this is fine for overall analyses of sample size, our aim requires a different approach. Because we are interested in minimizing the number of questions asked, we must consider the questions in the order in which they were asked in case judgments change over time. Therefore, we assess the effect of sample size on accuracy for questions that are randomly selected as well as those that are selected in the order experienced by the participants.

To address the aims of this study, we reanalyzed the two similarity judgment data sets used in Stevens and Soh [6]. We repeatedly split the data from each individual into a training set and testing set. We fit each algorithm to the training set and then used the fitted model to predict the testing set [14]. We calculated accuracy, precision, and recall on this out-of-sample testing set. With this method, we investigated (1) which algorithms performed best, (2) which predictors best predicted judgments, and (3) how sample size and question order influenced predictive accuracy for similarity judgments.

Methods

Data sets

We tested the different machine-learning algorithms on two data sets used by Stevens and Soh [6] (available at https://doi.org/10.17605/OSF.IO/EW8DC). In both data sets, participants with inattentive choice (e.g., judged 10 vs. 10 to be dissimilar or 1 vs. 90 to be similar), inconsistent choice (in a step-wise increase of large values, switching judgments more than three times), or near uniform choice ($\geq 95\%$ choice for similar or dissimilar) were removed. This eliminated 32 of the 155 participants, leaving 123 for our current analysis.

The first data set was collected from 50 participants (25 males and 25 females) with a mean±SD age of 28.6±3.8 (range 24-42) years recruited from the Adaptive Behavior and Cognition Web Panel at the Max Planck Institute for Human Development in Berlin, Germany in August 2011. Participants received a flat fee of €3 for completing the survey. Web panel participants made similarity judgments between 50 pairs of amount values (e.g., €6 vs. €8) and 49 pairs of delay values (e.g., 6 days vs. 8 days): "Please decide whether the numbers are similar". This research was approved by the Max Planck Institute for Human Development's Ethics Committee.

The second data set was collected from 73 participants (25 males and 48 females) with a mean±SD age of 19.9±1.6 (range 18-26) years recruited from the University of Nebraska-Lincoln Department of Psychology undergraduate participant pool in December 2014. Participants received course credit for their participation. Participants started by making 20 intertemporal choices before rating the similarity of 41 reward amount values and 42 time delay values: "Do you consider receiving [small amount] and [large amount] to be similar or dissimilar?" and "Do you consider waiting [short delay] and [long delay] to be similar or dissimilar?". The intertemporal

Table 1

Confusion matrix for true vs. predicted judgments with accuracy, precision, and recall

Predicted judgment	**True judgment**		
	Judged similar	*Judged dissimilar*	
Predicted similar	True Similar (TS)	False Similar (FS)	Precision = $\frac{TS}{TS+FS}$
Predicted dissimilar	False Dissimilar (FD)	True Dissimilar (TD)	
	Recall = $\frac{TS}{TS+FD}$		Accuracy = $\frac{TS+TD}{TS+FS+FD+TD}$

choices used the same value pairs as the similarity judgments and were included first to expose participants to the range of amount and delay magnitudes and to provide the overall decision context before they made similarity judgments. This research was approved by the University of Nebraska-Lincoln Internal Review Board (IRB Approval # 20130313118EP).

Data analysis

We used R [15] and the R-packages *C50* [16], *caret* [17], *e1071* [18], *foreach* [19], *GGally* [20], *here* [21], *kernlab* [22], *naivebayes* [23], *nnet* [24], *papaja* [25], *patchwork* [26], *randomForest* [27], *rpart* [28], *tidytext* [29], and *tidyverse* [30] for all our analyses (package usage described in the R script found in Supplementary Materials). The manuscript was created using *rmarkdown* [31]. Data, analysis scripts, supplementary tables and figures, and the reproducible research materials are available in Supplementary Materials and at the Open Science Framework (https://doi.org/10.17605/OSF.IO/EDQ39).

Predictors. We adapted predictors used in [6] for our investigation in this paper. In the original study, Stevens and Soh tested 11 predictors: small value, large value, difference, ratio, mean ratio, log ratio, relative difference, disparity ratio, salience, discriminability, and logistic (Table S1). However, we observed that a number of these predictors are very similar functions and thus may suffer from multicollinearity, which can be a problem for some machine-learning algorithms [7]. Therefore, we computed pairwise correlations for all predictors (Figures S1 & S2). Correlation coefficients for ratio, mean ratio, log ratio, relative difference, disparity ratio, salience, and discriminability all exceeded 0.81. Therefore, we removed mean ratio, relative difference, disparity ratio, and salience from the analyses. We kept ratio, log ratio, and discriminability as predictors because ratio was a key predictor in [6] and log ratio and discriminability both have curvilinear relationships with ratio and therefore may provide additional information for classification. Thus, the following analyses include small value, large value, difference, ratio, log ratio, discriminability, and logistic.

Algorithms. We used a set of commonly used algorithms, including tree-based models C5.0 [7, 32] and random forest [33], k-nearest neighbor [34], naive Bayes [35], neural networks [36], and support vector machines [37]. We combined these with those used in [6]: CART [11] and logistic regression.

Accuracy, precision, and recall. All analyses were conducted at the level of the individual participant for each judgment type (amount and delay). We conducted analyses for two different orderings: random and sequential. For random ordering, we first partitioned the data using a stratified random sample based on similarity judgments, so the training and testing sets had comparable distributions of similarity judgments (i.e., approximately the same proportion of "similar" vs. "dissimilar" judgments in both sets). For sequential ordering, we created the training set by drawing the judgments in the order in which each participant made their similarity judgments. Once the training sets were drawn, for both orderings, we generated testing sets by randomly drawing 10 samples from the non-training judgments. This ensured that all testing sets included the same number of judgments, regardless of training set size.

Because one of our research aims involved exploring how sample size influenced algorithm predictive accuracy, we analyzed accuracy over a range of training set sizes. The two data sets included 50 and 43 judgments of each type, and we analyzed training set sizes of 15, 20, 25, and 30 samples for both data sets. For data set 1, this is equivalent to 30%, 40%, 50%, and 60% of the total data, and, for data set 2, this maps to 36%, 48%, 59%, and 71% of the total data.

We fit models on each training set for each algorithm using the `train` function in the *caret* package [17], which uses bootstrapping to resample the data and fit the model repeatedly [7]. We applied each model to the training set and calculated accuracy, precision and recall for the training data (Table S2; Figures S3 and S4). We then used the models to predict the testing data to calculate out-of-sample accuracy, precision, and recall. This process was repeated 100 times for each data set, judgment type, subject, algorithm, and training set size. We then calculated the mean accuracy, precision, and recall over the 100 repetitions (see Table S2 for other performance metrics).

Predictor importance. All algorithms except support vector machines provide a measure of predictor importance. We
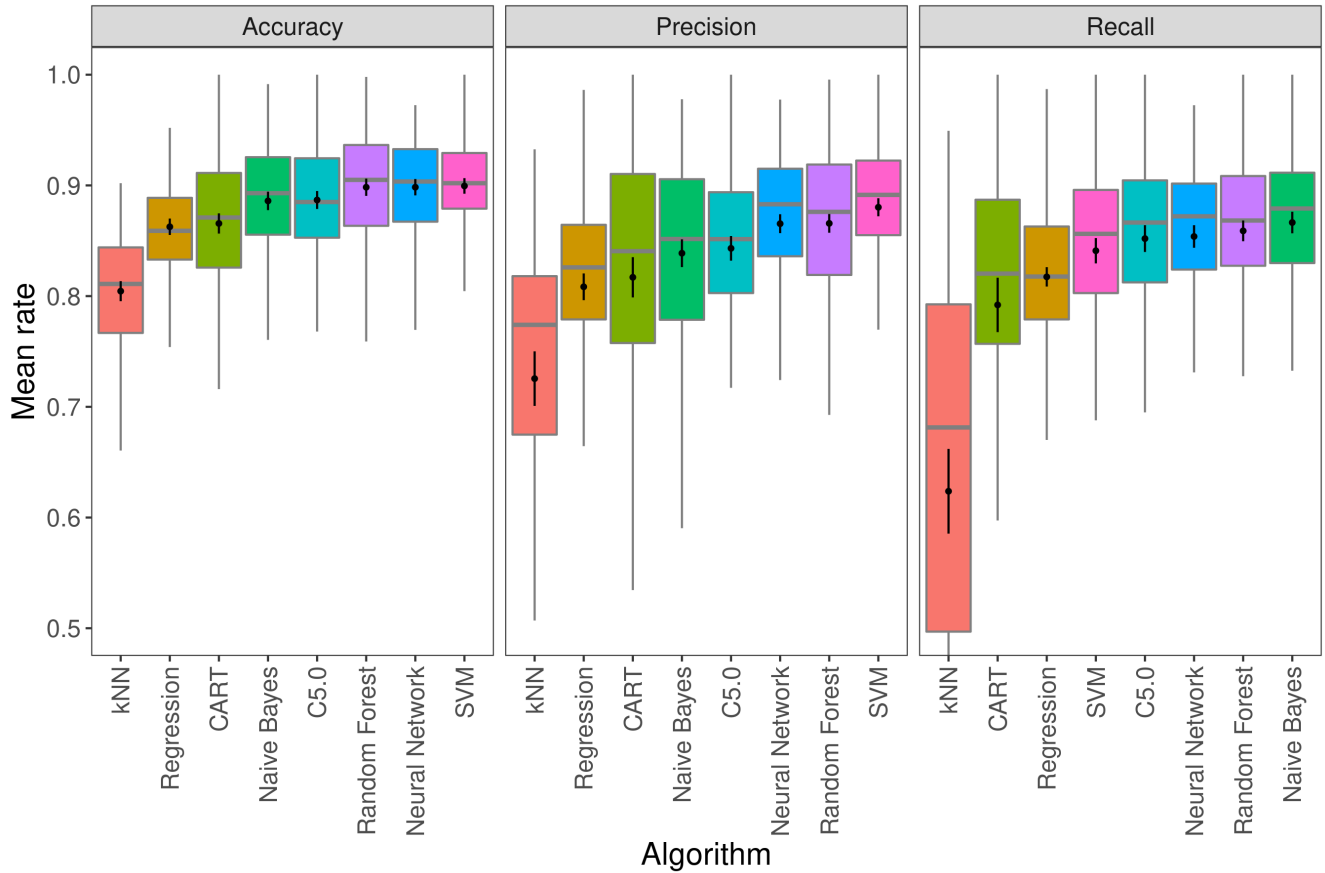
Figure 1. Out-of-sample accuracy, precision, and recall for each algorithm based on random ordering of a sample size of 30 instances and a testing set size of 10 instances. For each performance measure, algorithms are ordered by mean score. Dots represent means, error bars represent within-subjects 95% confidence intervals, boxplot horizontal lines represent medians, boxes represent interquartile range (25-75th percentile), whiskers represent 1.5 × interquartile range. Outliers are not shown. Note the y-axis is truncated at 0.5 to enlarge the presentation of the means and confidence intervals.

calculated predictor importance on the full data set (no training and testing sets) for each participant, data set, judgment type, algorithm (except support vector machine), and predictor using the `varImp` function in the *caret* package [17]. While each model type has a different metric of importance (Table S3), we scaled importance values, with the most important variable importance set to 100.

Results

Algorithm performance

To determine which algorithms best predict similarity judgments, we measured accuracy, precision, and recall on out-of-sample predictions from the aforementioned eight algorithms. We calculated these measures on the largest sample size (30 samples) and with random ordering for each participant. The confusion matrix, accuracy, precision, and recall percentages for training and testing sets and each algorithm are available in Table S2. Figure 1 presents testing set accuracy, precision,

and recall rates for each algorithm summarized over data set and judgment type (training set results available in Figures S3 and S4). For accuracy (number of correct predictions / all predictions), neural network, random forest, and support vector machine algorithms yielded the highest accuracy rates at 90%, with naive Bayes and C5.0 performing slightly worse, followed by CART, logistic regression, and kNN. Precision (correct similar predictions / all similar predictions) shows a comparable ordering. For recall (correct similar predictions / actual similar judgments), naive Bayes tops the list, followed closely by random forest, neural networks, C5.0, and support vector machines. Similar rankings of the algorithms' performance were observed across both data sets and between amount and delay similarity judgments (Figure S5).

Predictor importance

Different algorithms use predictors differently, so the predictors can vary in their contribution to the model performance.
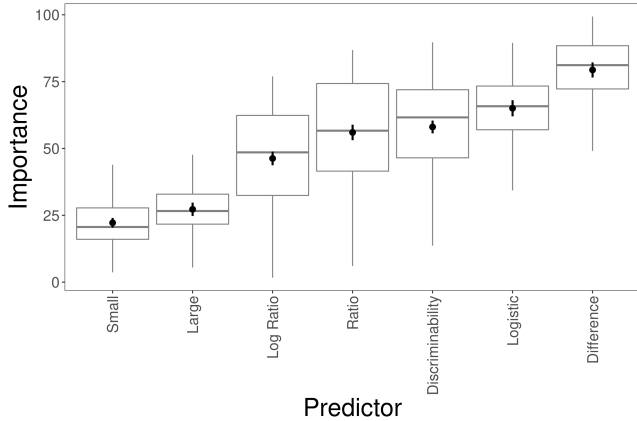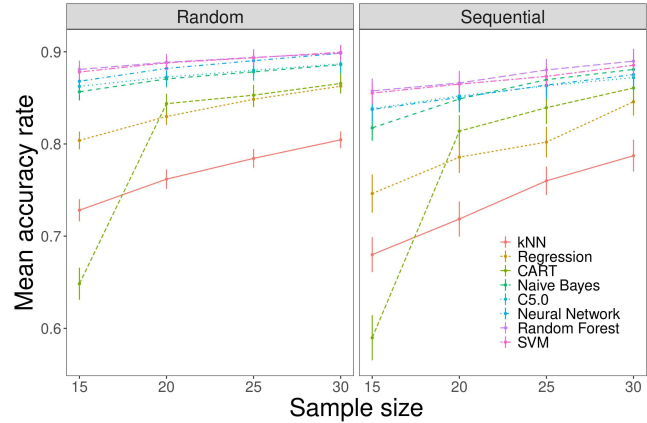
Figure 2. Importance of each predictor for each algorithm. Predictor importance refers to the relative contribution of each predictor to predicting the response. Predictors are ordered by mean importance. Dots represent means, error bars represent within-subjects 95% confidence intervals, boxplot horizontal lines represent medians, boxes represent interquartile range, whiskers represent 1.5 × interquartile range. Outliers are not shown.

Figure 3. Out-of-sample accuracy for each sample size for each algorithm. Sample size refers to number of questions per participant used to train the algorithms. Random refers to a random sample of training questions used to predict a random sample of 10 testing questions. Sequential refers to a sample of training questions drawn in order of presentation to each participant that was used to predict a random sample of 10 testing questions. Dots represent means, and error bars represent between-subjects 95% confidence intervals (within-subject confidence intervals were not used because excessive missing data for small sample sizes caused too many participants to be removed from the calculations).

To assess which predictors were most useful in predicting similarity judgments, we calculated predictor importance for each participant, data set, judgment type, algorithm, and predictor using the full data set. Figure 2 illustrates the importance of each predictor summarized over data set, judgment type, and algorithm. The numerical difference between large and small values was the most important predictor, followed by logistic, ratio and discriminability, log ratio, large value, and small value. Similar rankings of the predictors' performance were observed across both data sets and between amount and delay similarity judgments (Figure S6). While CART, kNN, naive Bayes, and random forest algorithms generate these rankings of predictor importance, C5.0, neural networks, and logistic regression generated different rankings (Figure S7). C5.0 was somewhat similar to the others, logistic regression showed little differentiation between predictors, and neural networks generated completely different rankings than the other algorithms.

Sample size and order

Developing small but predictive sets of judgment questions can allow us to predict judgments of value pairs that participants have not made. To investigate the effect of sample size on algorithm performance, we randomly sampled different training set sizes and repeatedly assessed each algorithm's accuracy in predicting a fixed, out-of-sample testing set. Figure 3 (left panel) shows out-of-sample accuracy for each algorithm at each sample size. Accuracy clearly increases with larger samples, but the rate of increase differs across algo-

rithms. Remarkably, random forest and support vector machines have about 88% accuracy at the smallest sample size of 15 (out of 43-50 judgments). Naive Bayes, C5.0, and neural networks yield only slightly lower accuracy rates of 86%. The remaining algorithms perform substantially worse at the lowest sample size but increase their performances with larger sizes. CART, in particular, performs very poorly at the lowest sample size but dramatically improves its performance at the next size, where it surpasses kNN and logistic regression. These rank orderings of algorithm performance hold across data sets and judgments types, with slightly lower accuracy rates in data set 2 (Figure S8A).

Though most assessments of sample size effects on algorithm performance randomly draw instances from data sets, the order in which participants experience questions can influence their responses. Given that the aim of this analysis is to determine how well small samples can predict judgments more generally, we must account for the sequential order in which participants make judgments. To investigate how well early questions can predict later ones, we fit the algorithms on training sets of various sizes, but, rather than randomly drawing the instances, we selected instances in the order in which participants experienced the questions. Figure 3 (right panel) shows out-of-sample accuracy for each algorithm at each sample size for the sequentially ordered data. The pattern

of results is qualitatively similar to those from the randomly selected data but with lower accuracy rates. Again, random forest and support vector machines top the algorithm rankings with only slightly lower accuracy than the random order (86%). And the algorithm rankings hold across data sets and judgment types (Figure S8B).

Discussion

Our analysis of algorithm performance found comparable levels of performance in accuracy, precision, and recall, but the algorithms differed in their performance across these three measures. Similarly, the different predictors varied in their contributions to algorithm performance, some of which matched previous findings, but others differed. Finally, as is typically the case in machine learning, algorithm performance improved with larger sample sizes, and the algorithms performed better predicting randomly selected samples than samples entered in the order experience by participants.

Algorithm performance

Neural network, random forest, and support vector machine algorithms generated the highest out-of-sample accuracy and precision for both data sets and judgment types. Naive Bayes and C5.0 joined all of these algorithms in showing the highest levels of recall. These analyses illustrate interesting differences across algorithms. First, this analysis contrasts with Stevens and Soh [6] showing better accuracy rates in CART than logistic regression. In the current analysis, CART and logistic regression have comparable levels of accuracy. This improvement in performance for logistic regression is likely due to removal of collinear predictors in the current analysis. Regression models are particularly susceptible to problems associated with multicollinearity [7].

The current analysis suggests that both CART and logistic regression are outperformed by a number of other machine-learning models, including C5.0, naive Bayes, neural networks, random forest, and support vector machines. Therefore, even higher levels of out-of-sample accuracy than those observed by Stevens and Soh [6] can be achieved by testing a wider range of models. A key reason that Stevens and Soh used CART was to test the possibility that decision trees capture the actual cognitive computations of decision making. That is, similarity judgments may actually be made in decision-tree-like ways. Thus, it is important to see that two other tree-based algorithms (C5.0 and random forest) outperform CART. While we do not directly test predictions about the computational process on C5.0 and random forest here, this provides a fruitful area of future research.

Decision trees are not the only class of algorithms that perform well. Neural networks and support vector machines perform as well as random forest. These algorithms, however, are "black box" algorithms in the sense that their process of converting predictors into predictions for the outcomes is not straightforward. Whereas random forest produces decision trees which can, in principle, mimic the cognitive computations of how judgments are made, neural networks produce a series of layers of nodes with weights connecting them [38], and support vector machines calculate multidimensional hyperplanes [39]. Therefore, though neural networks mimic neural computations, these algorithms do not resemble a cognitive process.

The three performance measures were quite consistent across data sets and judgment types. Consistency across data sets indicates robustness of these analyses within the area of similarity judgments. Although there were only two data sets analyzed, the actual similarity value pairs differed between the data sets, and, perhaps more importantly, the study sample population differed with Germans being sampled in data set 1 and Americans in data set 2. Nevertheless, both populations were relatively similar in age and educational level, with the Germans being slightly older. Both participant groups were drawn from predominantly white, educated, industrialized, rich, and democratic (WEIRD) populations [40]. The narrow scope of the questions and the similarity of the study populations make it difficult to generalize our findings beyond similarity judgments in WEIRD populations.

Predictor importance

A key feature of many algorithms is that they can offer a metric of how much each predictor contributed to the predictions. This predictor importance offers insight into which predictors are most useful. Across all algorithms, our analysis showed that the numerical difference predictor contributed the most to predictive performance, followed by logistic, discriminability, and ratio. Stevens and Soh [6] also found difference to be the primary predictor used as the first node in 62-71% of participants' decision trees. In fact, difference was the most important predictor in the current analysis for all algorithms except logistic regression and neural networks. This provides robust evidence that one of the simplest predictors (large value − small value) is also the most important in making similarity judgments.

One key difference between the current analysis and Stevens and Soh [6] is the next most important predictors. Stevens and Soh found that ratio was the second most used primary node predictor for CART (27-33% of participants), with relative difference and logistic following (1-2%). The current analysis showed logistic followed by discriminability and ratio. This is a surprising contradiction of Stevens and Soh's findings because logistic and discriminability are more mathematically complicated combinations of small value and large value compared to ratio (Table S1). Though a simple predictor is the most important predictor, the next most important predictor could be a more complex combination of small and large

values.

The discrepancy with Stevens and Soh [6] could arise because of two reasons. First, the measure of predictor importance in the current analysis is based on different types of metrics across algorithms (Table S3) that are scaled similarly for comparison. Because different algorithms use different metrics, the scaling (apart from the most and least important predictor) may not be comparable across algorithms. Therefore, the predictors of intermediate importance may be compressed or expanded differently across algorithms. Nevertheless, logistic was the second most important predictor across all but two of the algorithms. Second, the set of predictors in the two analyses differed. Stevens and Soh included all 11 predictors, and the current analysis used a limited set of predictors to reduce multicollinearity. The multicollinearity of many of the predictors with ratio could have somehow boosted its performance, whereas without multicollinearity, ratio's contribution could have been reduced. This finding speaks to the importance of feature selection in investigating predictor importance [41].

Finally, for this analysis, we eliminated a number of predictors from our analysis because of their multicollinearity with other predictors. Some of these predictors play key roles in psychological and economic models of intertemporal and risky choice (e.g., relative difference, salience). While these models may yield insights into decision making, it is important to consider that they are highly correlated with simpler predictors of these choices (e.g., ratio). Thus, parsimony requires substantial additional predictive value for these models to be favored over simpler ratio-based models.

Sample size and order

Sample size is a key aspect of algorithm performance [13]. As expected, we found that accuracy increased with sample size of randomly selected data. Some algorithms (notably random forest and support vector machines) showed high out-of-sample accuracy even at the smallest size (15 instances or 30-36% of the total number of instances). Therefore, choosing the appropriate algorithm can result in high out-of-sample accuracy even with small samples.

Analyses of randomly selected data, however, do not capture the potential effects of the order of experiencing questions on participants' judgments. That is, participants may get tired or change their judgment criteria over time. So judgments made early during testing may not match those made later in testing. To explore this, we analyzed the data by entering the instances in the order experienced by participants and examining accuracy across a range of sample sizes. Including the sequentially ordered instances reduced accuracy, but random forest and support vector machines still outperformed other algorithms, especially at small sample sizes.

While other algorithms dropped in accuracy substantially, random forest and support vector machines maintained very high accuracy for the sequentially ordered data. At the smallest sample size, these two algorithms correctly predicted 86% of the judgments. This level of accuracy with such small samples sizes is remarkable and bodes well for being able to collect rather small samples and extrapolate more generally.

In summary, we have evidence that machine-learning algorithms can take as input small amounts of data and make robust out-of-sample predictions. Leveraging these algorithms can influence experimental designs by requiring fewer questions. By reducing numbers of questions, we can minimize the burden on participants, which can either improve data quality by not tiring participants or allow the opportunity to add other experimental procedures when participant time is limited. Either way, employing machine-learning algorithms can enhance experimental design.

Limitations and future directions

This article expands the application of machine learning to similarity judgments compared to Stevens and Soh [6] by investigating more algorithms, more measures of performance, more sophisticated measures of predictor importance, and a more nuanced approach to sample size. However, the tools available in machine learning are many, and they are increasing in number and sophistication. We limited our analysis to eight algorithms, chosen based on suitability for our data and previous frequency of use in the machine-learning literature. Of course, there are other algorithms that we could have tested, some of which might have outperformed our top models. Nevertheless, we used a standard set of models, many of which had equally high performance. It seems unlikely additional models would provide substantial new insights or contradictory information.

A great deal of effort has focused on developing methods to optimize model parameters to improve fit [7]. We took a relatively basic approach to tuning model parameters, primarily using default options in our analysis software. It is possible that more sophisticated parameter tuning could yield different results. However, more sophisticated tuning often comes at the price of longer computation times. We have opted to minimize computation time by using the default tuning methods. Finally, optimizing parameters can result in models overfitting data. We used standard cross-validation techniques to reduce overfitting by both calculating predictive performance measures on out-of-sample data fitted on training data and fitting models to the training data using resampling techniques [7].

In general, machine-learning models perform best with many instances to work with. This allows for large training sets that include representative instances from the population of

possible instances. Though we have a large number of total instances (over 11,000), we conducted the analysis at the level of the participant and judgment type (amount or delay judgment) because we were interested in being able to predict individual participant judgments. This resulted in only 40-50 instances per analysis, which is rather small for machine-learning analyses that use cross-validation. This is apparent with the poor performance of CART at sample sizes of 15 samples but rapid improvement at 20 samples (Figure 3). The other algorithms, however, show a more gradual increase in performance with sample size, suggesting that sample sizes used here are not too small to allow reasonable performance. From a logistical perspective, having participants answer more than 50 questions for each judgment is already rather tiring, and increasing the number of questions could result in poor data quality. So, though more instances could be better for the model performance, the models perform well with these sample sizes, and increasing them could produce more problematic data.

This article has focused on similarity judgments of monetary amounts and time delays because they are the attributes that are relevant to intertemporal choice. But the similarity approach also applies to risky and strategic choice [42–44]. Thus, this approach can be expanded beyond amounts and delays to probabilities of receiving rewards, an attribute of risky choice. Probabilities, however, are bounded, which could result in different algorithms and predictor importance compared to amounts and delays. Though the similarity approach has not been formally applied to multiattribute choice (e.g., choosing an apartment based on rent, size, distance from work, etc.), this is another area to which it could be applied. The scale and boundedness of the attribute values could influence how similarity is assessed, but these methods should be able to apply to most quantitative attributes. Yet research on similarity is not limited to quantitative attributes [45–47], and machine learning has broad application to understanding both quantitative and non-quantitative components of similarity [48, 49].

Conclusions

Machine learning comprises a powerful set of tools to classify outcomes. While some areas of psychology have been fruitfully using machine learning for a while [50, 51], the field has not leveraged these tools fully [14]. Judgment and decision making, in particular, is an area ripe for applying machine learning, and some have taken advantage of these tools [52–54]. Here, we used machine learning to achieve multiple goals. First, we assessed the performance of several algorithms in predicting similarity judgments from participant data. Though evaluating algorithm performance is not typically a psychological question, in our case, we investigated whether decision tree algorithms performed well, since they could offer cogni-

tive process-based models of actual decision making. Indeed, previously, we found that CART outperformed regression we found that the random forest algorithm—one that is based on decision trees—topped the list of best-performing algorithms. We can further probe this algorithm because, not only does it accurately predict similarity judgments, it also may give us a window into the process of classification by generating measures of predictor importance and allowing the extraction of a step-by-step set of rules used to generate the predictions. Testing a broad range of machine-learning algorithms allowed us to pinpoint a highly accurate model that may also approximate the actual judgment process. Future research will integrate these algorithms into the similarity model to predict the intertemporal choices.

Second, our analysis provided the opportunity to examine which predictors were most important in making the judgment classifications. While regression alone can provide information about predictor performance, it is only a single model, and its predictions depend on its assumptions and methods. Our analysis produced predictor importance measures across a range of algorithms, which can provide information about the robustness of importance across models. For instance, we found rather consistent rankings of predictor importance across four very different types of algorithms (Figure 2). But differences across algorithms are interesting as well. For example, while it has above average importance in most algorithms, the predictor discriminability is ranked most important by neural networks. This could inspire further investigations, as assessing predictor importance across a range of algorithms can be useful in drawing inferences about those predictors.

Finally, in addition to answering theoretical questions about models and predictors, machine learning can inform the logistics of data collection. We evaluated algorithm accuracy across a range of training set sizes to see how robust they are to sample size. Moreover, we used samples ordered by how they were experienced by participants to see how predictive different numbers of questions were to judgments more generally. Our analysis showed that some algorithms could predict judgments with quite high accuracy at rather small sample sizes. This finding is useful for designing future studies, where we can trim the number of questions that we ask participants, which can reduce participant fatigue or allow time to ask other questions. Thus, using machine-learning algorithms can help us both understand our data in more depth and design better ways of collecting those data.

Declarations

Funding

Conflicts of interest

The authors declare no known conflicts of interest.

Availability of data and material

All data files and supplementary materials (tables, figures) are available at https://doi.org/10.17605/OSF.IO/EDQ39.

Code availability

All data analysis scripts are available at https://doi.org/10.176 05/OSF.IO/EDQ39.

Author's contribution

The authors made the following contributions. Jeffrey R. Stevens: Conceptualization, Data Curation, Formal Analysis, Funding Acquisition, Investigation, Methodology, Project Administration, Resources, Software, Supervision, Visualization, Writing - Original Draft Preparation, Writing - Review & Editing; Alexis Polzkill Saltzman: Formal Analysis, Methodology, Software, Visualization, Writing - Review & Editing; Tanner Rasmussen: Formal Analysis, Methodology, Software, Visualization, Writing - Review & Editing; Leen-Kiat Soh: Conceptualization, Funding Acquisition, Investigation, Supervision, Writing - Review & Editing

References

1. Read, D. (2004). Intertemporal choice. In D. Koehler & N. Harvey (Eds.), *Blackwell Handbook of Judgment and Decision Making* (pp. 424–443). Oxford, UK: Blackwell.

2. Doyle, J. R. (2013). Survey of time preference, delay discounting models. *Judgment and Decision Making*, *8*(2), 116–135.

3. Leland, J. W. (2002). Similarity judgments and anomalies in intertemporal choice. *Economic Inquiry*, *40*(4), 574–581. https://doi.org/10.1093/ei/40.4.574

4. Rubinstein, A. (2003). "Economics and psychology"? The case of hyperbolic discounting. *International Economic Review*, *44*(4), 1207–1216. https://doi.or g/10.1111/1468-2354.t01-1-00106

5. Stevens, J. R. (2016). Intertemporal similarity: Discounting as a last resort. *Journal of Behavioral Decision Making*, *29*(1), 12–24. https://doi.org/10.1002/bdm.1870

6. Stevens, J. R., & Soh, L.-K. (2018). Predicting similarity judgments in intertemporal choice with machine learning. *Psychonomic Bulletin & Review*, *25*(2), 627–635. https://doi.org/10.3758/s13423-017-1398-1

7. Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. New York: Springer.

8. Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer.

9. Murthy, S. K. (1998). Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, *2*(4), 345–389. https://doi.org/10.1023/A:1009744630224

10. Fürnkranz, J. (2010). Decision tree. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning* (pp. 263–267). Boston, MA: Springer. https://doi.org/10.1007/978-0-387-30164-8_204

11. Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. New York: Chapman and Hall.

12. Ting, K. M. (2010). Confusion matrix. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning* (pp. 209–209). Boston, MA: Springer. https://doi.org/10.1007/978-0-387-30164-8_157

13. Perlich, C., Provost, F., & Simonoff, J. S. (2003). Tree induction versus logistic regression: A learning-curve analysis. *Journal of Machine Learning Research*, *4*, 211–255.

14. Yarkoni, T., & Westfall, J. (2017). Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, *12*(6), 1100–1122. https://doi.org/10.1177/ 1745691617693393

15. R Core Team. (2018). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

16. Kuhn, M., & Quinlan, R. (2020). *C50: C5.0 decision trees and rule-based models*. Retrieved from https://CRAN.R-project.org/package=C50

17. Kuhn, M. (2020). *caret: Classification and regression training*. Retrieved from https://CRAN.R-project.org/package=caret

18. Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., & Leisch, F. (2019). *e1071: Misc functions of the department of statistics, probability theory group (formerly: E1071), TU Wien*. Retrieved from https://CRAN.R-project.org/package=e1071

19. Microsoft, & Weston, S. (2020). *foreach: Provides foreach looping construct*. Retrieved from https://CRAN.R-project.org/package=foreach

20. Schloerke, B., Crowley, J., Cook, D., Briatte, F., Marbach, M., Thoen, E., . . . Larmarange, J. (2020). *GGally: Extension to "ggplot2"*. Retrieved from https://CRAN.R-project.org/package=GGally

21. Müller, K. (2017). *here: A simpler way to find your files*. Retrieved from https://CRAN.R-project.org/package=here

22. Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). Kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, *11*(9), 1–20. Retrieved from http://www.jstatsoft.org/v11/i09/

23. Majka, M. (2019). *naivebayes: High performance implementation of the naive Bayes algorithm in R*. Retrieved from https://CRAN.R-project.org/package=naivebayes

24. Venables, W. N., & Ripley, B. D. (2002). *Modern Applied Statistics with S* (4th ed.). New York: Springer. Retrieved from http://www.stats.ox.ac.uk/pub/MASS4

25. Aust, F., & Barth, M. (2018). *papaja: Create APA manuscripts with R Markdown*. Retrieved from https://github.com/crsh/papaja

26. Pedersen, T. L. (2019). *patchwork: The composer of plots*. Retrieved from https://CRAN.R-project.org/package=patchwork

27. Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, *2*(3), 18–22. Retrieved from https://CRAN.R-project.org/doc/Rnews/

28. Therneau, T., & Atkinson, B. (2019). *rpart: Recursive partitioning and regression trees*. Retrieved from https://CRAN.R-project.org/package=rpart

29. Silge, J., & Robinson, D. (2016). tidytext: Text mining and analysis using tidy data principles in R. *Journal of Open Source Software*, *1*(3). https://doi.org/10.21105/joss.00037

30. Wickham, H. (2017). *tidyverse: Easily install and load the "tidyverse"*. Retrieved from https://CRAN.R-project.org/package=tidyverse

31. Xie, Y., Allaire, J. J., & Grolemund, G. (2018). *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman; Hall/CRC. Retrieved from https://bookdown.org/yihui/rmarkdown

32. Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers Inc.

33. Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

34. Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

35. Maron, M. E. (1961). Automatic indexing: An experimental inquiry. *Journal of the ACM*, *8*(3), 404–417. https://doi.org/10.1145/321075.321084

36. McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

37. Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (pp. 144–152). Pittsburgh, Pennsylvania, USA: Association for Computing Machinery. https://doi.org/10.1145/130385.130401

38. Laine, A. (2003). Neural networks. In A. Ralston, E. D. Reilley, & D. Hemmendinger (Eds.), *Encyclopedia of Computer Science* (pp. 1233–1239). John Wiley and Sons Ltd.

39. Zhang, X. (2010). Support vector machines. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning* (pp. 941–946). Boston, MA: Springer. https://doi.org/10.1007/978-0-387-30164-8_804

40. Henrich, J., Heine, S. J., & Norenzayan, A. (2010). The weirdest people in the world? *Behavioral and Brain Sciences*, *33*(2-3), 61–83. https://doi.org/10.1017/S0140525X0999152X

41. Kuhn, M., & Johnson, K. (2019). *Feature Engineering and Selection: A Practical Approach for Predictive Models*. CRC Press.

42. Rubinstein, A. (1988). Similarity and decision-making under risk (Is there a utility theory resolution to the Allais paradox?). *Journal of Economic Theory*, *46*(1), 145–153. https://doi.org/10.1016/0022-0531(88)90154-8

43. Leland, J. W. (1994). Generalized similarity judgments: An alternative explanation for choice anomalies. *Journal of Risk and Uncertainty*, *9*(2), 151–172.

44. Leland, J. W. (2013). Equilibrium selection, similarity judgments, and the "nothing to gain/nothing to lose" effect. *Journal of Behavioral Decision Making*, *26*(5), 418–428. https://doi.org/10.1002/bdm.1772

45. Tversky, A. (1977). Features of similarity. *Psychological*

Review, *84*(4), 327–352. https://doi.org/10.1037/00 33-295X.84.4.327

46. Shepard, R. (1987). Toward a universal law of generalization for psychological science. *Science*, *237*(4820), 1317–1323. https://doi.org/10.1126/science.3629 243

47. Goldstone, R. L., & Son, J. (2005). Similarity. In K. J. Holyoak & R. Morrison (Eds.), *Cambridge Handbook of Thinking and Reasoning* (pp. 13–36). Cambridge, UK: Cambridge University Press.

48. Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, *6*(1), 37–66. https://doi.org/10.1007/BF00153759

49. Hahn, U., & Chater, N. (1998). Similarity and rules: Distinct? Exhaustive? Empirically distinguishable? *Cognition*, *65*(2-3), 197–230. https://doi.org/10.101 6/S0010-0277(97)00044-9

50. Mooney, R. J. (1993). Integrating theory and data in category learning. In *Categorization by Humans and Machines: Advances in Research and Theory* (pp. 189–218). San Diego, CA, US: Academic Press.

51. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press.

52. Kattan, M. W., Adams, D. A., & Parks, M. S. (1993). A comparison of machine learning with human judgment. *Journal of Management Information Systems*, *9*(4), 37–57. https://doi.org/10.1080/07421222.199 3.11517977

53. Rosenfeld, A., Zuckerman, I., Azaria, A., & Kraus, S. (2012). Combining psychological models with machine learning to better predict people's decisions. *Synthese*, *189*(1), 81–93. https://doi.org/10.1007/s1 1229-012-0182-z

54. Brighton, H., & Gigerenzer, G. (2015). The bias bias. *Journal of Business Research*, *68*(8), 1772–1784. https://doi.org/10.1016/j.jbusres.2015.01.061