Using Cognitive Models to Train Big Data Models with Small Data

J. Gregory Trafton
Naval Research Laboratory
Washington, DC
greg.trafton@nrl.navy.mil

Laura M. Hiatt
Naval Research Laboratory
Washington, DC
laura.hiatt@nrl.navy.mil

Benjamin Brumback
Naval Research Laboratory
Washington, DC
benjamin.brumback@nrl.navy.mil

J. Malcolm McCurry
Peraton
Washington, DC
jmccurry@peraton.com

ABSTRACT

Modeling and predicting human behavior pose a difficult challenge for AI and other related fields. Some current techniques (e.g., cognitive architectures) are able to model people's goals and actions from little data, but have poor predictive capabilities. Other methods (e.g., deep networks) have strong predictive capabilities but require large amounts of data to train the model; such abundant empirical data on human performance is not available for many human-based tasks. We show a novel and general method of generating copious synthetic data of human behavior using a cognitive architecture, and then use the data to train a deep network classifier to predict ensuing human actions. We test our approach by predicting human actions on a supervisory control task; the results show that our approach provides superior prediction when compared to training a classifier with only (limited) empirical data.

KEYWORDS

Cognitive models; Agent-based analysis of human interactions; Agents for improving human cooperative activities

ACM Reference Format:
J. Gregory Trafton, Laura M. Hiatt, Benjamin Brumback, and J. Malcolm McCurry. 2020. Using Cognitive Models to Train Big Data Models with Small Data. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

1 INTRODUCTION

We are interested in building predictive models of human behavior. In the types of situations we envision, people perform a series of actions with an overall objective or goal to accomplish. These actions are, in a sense, discrete, such as picking up an object or clicking an icon on a computer monitor. However, these actions are discrete only at an abstract level; as part of clicking an icon on the screen, a person has to to (1) decide what object to click, (2) visually find the object, (3) move the mouse to that location, and (4) finally click on the object. By taking advantage of the thought processes that lead up to an action, we can presumably build models of the human behavior to predict the action before it occurs and provide assistance to the human performing the task.

Modeling processes such as this, as well as of human-level behavior and intelligence in general, is one of the core goals of artificial intelligence (see [16] for a recent survey of current approaches). Modeling techniques cover a wide variety of human behavior, data types, and inference types (such as classifying behavior vs. predicting behavior). One way to delineate the space of models is by distinguishing between those that focus on interpreting the *outward observations* of behavior (such as statistical approaches), and those that focus on understanding the *internal processes* driving behavior (such as computational cognitive models).

The different emphases of these models lead to different data requirements for training and validating them, as well as different insights that they can provide. Computational cognitive models, for example, are typically developed to capture human behavior on a specific task (e.g., controlling a group of UAVs in a supervisory control task). Conceptually, a cognitive model is like a special computer program for a task that is executed within an encompassing architecture that constrains the model's execution to mimic human behavior on that task (e.g., [3, 20, 27]), including the variability of human behavior (e.g., human error). Developing such computational cognitive models typically relies heavily on the architecture itself. Because much data and experimentation goes into developing the architecture over the span of years, validating individual models requires relatively little data on the model's task compared to other statistical AI approaches – an experiment or two of human behavior on the task is not uncommon to be sufficient data to validate a cognitive model [13, 28]. Because of their emphasis on maintaining fidelity to the processes of human cognition, computational cognitive models typically focus on explaining behavior, not predicting it.

Current statistical approaches, in contrast, can provide strong predictive models, but have relatively little process understanding about why people behave the way they do. For this reason, these approaches require very large amounts of data to train effectively, even with current efforts striving to lessen it.

Generally, when considering tasks that involve human behavior, there is very little clean and labeled empirical (e.g., human) data available. For example, when considering supervisory control tasks, the data takes the form of eye fixations (e.g., what a person is looking at on their computer monitor) to provide insights into the

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

thought processes leading up to human actions, and mouse clicks to represent the actions themselves. Eye fixations are noisy indicators of actions that people will take for a number of reasons: people can look at something without fixating on it [9]; people use different perceptual strategies and have different levels of spatial and declarative memory; current eye-tracking sensors are still imperfect both spatially and temporally; etc. Thus, data is expensive to collect and difficult to label, making it extremely challenging to collect enough data to train predictive statistical models on supervisory control tasks. This challenge extends to other domains, as well.

To address this, we introduce a new, general methodology for generating predictive models of human behavior that focuses on combining the relatively low data needs of computational cognitive models with the strong predictive capabilities of statistical models. To this end, we begin by developing cognitive models of different strategies people use to complete tasks, using the little empirical data available. We can then use those cognitive models to generate arbitrarily large amounts of synthetic data of human performance on the tasks.

This synthetic data is a major contribution of this paper, and can capture the variability of human performance on tasks both because we can capture different strategies people use, as well as because of the understanding of human variability that is part of cognitive models. The data is also generated with labels in place, since the cognitive model knows where it is looking, what it is doing, and why.

This large amount of data, in turn, allows us to train robust statistical learning models of human performance on the task that can provide the predictive capabilities required to provide assistance to human partners on these tasks.

We demonstrate and evaluate our methodology on a supervisory control task, where a human operator is responsible for supervising and interacting with several UAVs concurrently. Behavior on the task is recorded via eye fixations and mouse clicks. This is a difficult task where, as we will show, it is extremely challenging to collect and process even relatively small amounts of empirical data. Because of the way in which cognitive models capture the thought processes of human behavior, however, we can use cognitive models of this task to generate data that takes the same form as the human empirical data; namely, eye fixations leading up to a mouse click and action. On this task, our approach results in a classifier that provides superior prediction when compared to a classifier trained with only the limited empirical data.

We next discuss background on different ways of modeling behavior, and the pros and cons of each. We then introduce the specific supervisory control task we will use to demonstrate our approach. Next, we discuss the specific cognitive architecture we employ, cognitive model development, data generation, and training of a statistical model. Finally, we describe how we evaluate our approach, and end with a discussion of the wider implications.

2 BACKGROUND

2.1 Modeling people's internal processes

ACT-R [2, 3] is a computational cognitive architecture that enables models of human behavior to be built with a high degree of fidelity to human cognition. ACT-R focuses on capturing the same types of representations, processes, and strategies that people use. ACT-R has been used to model development, problem solving, memory, robot interaction, decision making, and many more. ACT-R's strength is in providing a process-level description of human experiments, yielding performance that matches the experimental data (e.g., reaction time). ACT-R is described more below; we use it in this paper to model human behavior.

Other architectures and frameworks for capturing human cognition also exist. Soar [20] is a different computational cognitive architecture, but with a focus more on the computational building blocks of intelligence rather than modeling the specific processes and representations of humans. Soar has been used to model categorization, learning, language, both episodic and semantic memory, interactive task learning, and many more. Soar's strength is in creating models that can act intelligently while being computationally efficient and scalable.

GOMS [17] is a framework that models the Goals, Operators, Methods, and Selection Rules of a human performing an computer interface. Each operator has a specific execution time; when the operators are combined via goals, methods, and selection rules, quantitative predictions of how long a specific task will take can be made. GOMS has been used to model CAD design, telephone workstation efficiency, and many more. GOMS's strength is in determining how long a specific interface action (or series of actions) will take, and then comparing alternatives to determine the most time-efficient approach.

There are, of course, many other approaches to modeling people's internal processes and strategies. Mental model theory focuses on how people reason [18]; BPL focuses on concept learning [21]; diffusion models focus on how people remember [26].

2.2 Modeling people's outward actions

Other areas of AI have, in contrast, created strong methods for modeling people's outward actions without regard to how or why people perform those actions. Statistical approaches like hidden Markov models excel at inferring future actions that people will perform (e.g., [24]). Various configurations of neural networks have also been extremely successful at sequence prediction [33, 34] and activity recognition and classification [19]. However, most approaches require substantial amounts of training data to perform adequately. Deep neural networks, for example, typically require approximately 1000 instances per test case in order to develop accurate and effective models. In fact, one of the cottage industries within AI is to construct and publish large datasets that can be used to train deep neural networks. For example, MNIST [23] contains 60,000 handwritten numbers, ImageNet [8] contains 1.5M images, Sentiment140 [11] contains 1.6M tweets, and more datasets are appearing at a regular rate.

Even more critical to the amount of data generally available is the need for the available data to be labeled. This presents an additional challenge that scientists have found several different ways to address. For example, ImageNet used Amazon Mechanical Turk to label images [8]. For MNIST, researchers directed participants to write down specific numbers, so the data and labels were generated concurrently [12]. Sentiment140 did not hand-label sentiments, instead using noisy labels (emoticons) to classify tweets [11].

Together, these two aspects of statistical approaches imply that it is extremely challenging to get sufficient empirical data in order to train effective statistical models of human behavior. In part because of this, there are no large datasets on human behavior like the above standard dataset examples. We further discuss the involved challenges next.

2.3 Learning from empirical human data

In addition to what is outlined above, there are a series of additional challenges to address when training predictive statistical models on empirical human data. These problems can be divided into a *lack of data challenge*, a *labeling challenge*, a *strategy challenge*, and a *balanced data challenge*. Each of these challenges is discussed below. We revisit these challenges in the discussion section and discuss how our approach addresses them.

Lack of data challenge: Human data collection is very expensive for highly skilled or technical tasks. Amazon Mechanical Turk has increased the ability to collect data from relatively unskilled workers [25], but collecting data on how skilled performers execute their tasks can become prohibitively expensive in terms of money, time, or both [4]. Note that there are some existing methods of data augmentation to partially address this problem, but they generally do not apply to human behavioral tasks.

Labeling challenge: Labeling an individual's actions and goals from behavioral observations like where a person is looking is extremely challenging and requires a specialized set of techniques, including knowledge of human thought processes and cognition. Determining an individual's purpose for every observation is particularly difficult in dynamic tasks or environments [32], and considering the noisy aspects of human cognition.

Strategy challenge: In order to achieve a particular objective, people use a variety of strategies [10, 22]. Because of the different strategies available on even very simple tasks, it can be very difficult to interpret how specific actions lead to a person's goals or objectives; as above, this is especially true given the noisiness of human cognition and behavior.

Balanced data challenge: When there is relatively little data, and that data can be highly variable, the prediction classes can be very imbalanced. While this can be a problem with big data, it can be exacerbated when only smaller amounts of data are available and there is less data available to capture the true distribution. Thus, any classification or prediction system will be biased to those more frequent classes, potentially leading to incorrect results [30].

3 TASK DESCRIPTION

In order to address the above challenges for data on human behavior, we turned to a supervisory control task. Specifically, we studied how people performed when interacting with the Research Environment for Supervisory Control of Heterogeneous Unmanned Vehicles (RESCHU) [5] simulator. RESCHU is an interactive system that has a variety of objectives and strategies and that uses complex decision making, problem solving, and reasoning.

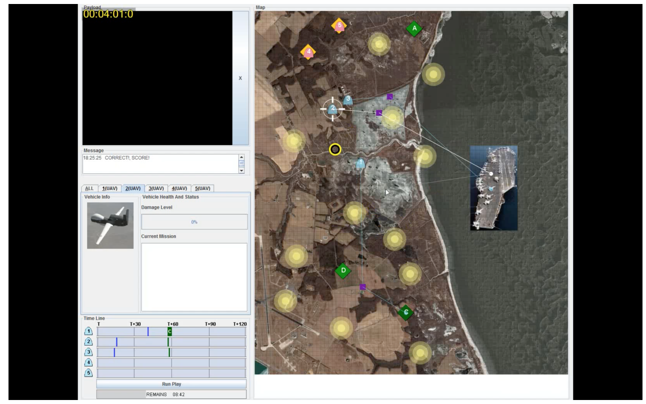

Figure 1: A screenshot of the RESCHU environment simulator used in this experiment.

RESCHU is a heterogenous supervisory control task. The interface of the supervisory control simulation, shown in Figure 1, has three main sections: the map panel, the status panel, and the payload panel. The map panel (Figure 1, right panel) displays UAVs (blue half ovals), targets (red diamonds) towards which UAVs are moving, and threats (yellow circles) which should be avoided by UAVs. The status panel (Figure 1, bottom left panel) shows the status of the UAVs and includes information on vehicle damage, time until the vehicle reaches a waypoint or target, and time remaining in the simulation. The payload panel (Figure 1, top left panel) is used to perform a manual visual acquisition task once the UAV has reached the target (this task is not critical to this work; for those interested, it is more fully described in [6]).

The operator's high level job in the simulation is to monitor UAVs as they proceed to specific target areas in the map panel, and to perform the manual visual acquisition task once the UAV has reached the target. Throughout the session, 5 UAVs moved along straight-line trajectories towards an automatically-assigned target. There were always 5 targets present on the map. At the start of the simulation the UAVs were randomly assigned to different targets; thus, the UAVs might not be directed towards the optimal target. After a target was reached and the visual acquisition payload task was complete, the UAV was randomly assigned to a new currently-unassigned target, which again might not be optimal.

There were also eighteen threat areas. Every four seconds, one of the eighteen threats was randomly selected to change its position, potentially in the path of a UAV. If the UAV passed through a threat, it incurred damage, and would eventually become incapacitated if it were damaged enough.

The simulation was a complex task with multiple events happening in parallel. More than one UAV could be waiting at their respective targets for the operator to perform the payload task, and more than one UAV could be on a path intersecting a threat area at a time.

Operators could change the target of a UAV at any time by clicking on the UAV and then clicking on the UAV's new target. Operators could also drag a UAV's goal point to a different destination. Generally, this occurs when a UAV is heading to a suboptimal target, when a UAV's path to a target brings it through a hazard area, or to keep an area saturated or balanced with UAVs. This action – changing the goal of a UAV to a different target – is thus critical to the success of the operator. By being able to predict what new target a user is going to send a UAV to, the system should additionally be able to facilitate the interaction (e.g., by pre-selecting the object or highlighting the object for easier selection). Therefore, we focused our modeling and prediction on this specific component of the supervisory control task.

The empirical data we consider here was based on ten participants using RESCHU. Participants were provided extensive training on the RESCHU system, through an online tutorial, in-person instruction, and walk-throughs. Participants also had as much time as they wanted to use the entire system until they were well-versed in the intricacies of RESCHU. Participants were all volunteers (no incentives), healthy, with age less than 30 years. Details on the methodology of the study are available in [6].

After a participant was fully trained on RESCHU, they were seated approximately 66 cm from the computer monitor and were calibrated on an SMI eye tracker. Eye tracking data were collected using an SMI RED eye tracker operating at 250 Hz. A fixation was defined using the dispersion method based on a minimum of 15 eye samples within 60 ms and within 50 pixels (approximately 3° of visual angle) of each other, calculated in Euclidian distance. The eye tracker and the RESCHU simulation were synchronized, such that the simulation sent the eye tracker an update of its state each time its state was updated (i.e., every 500 ms). Fixations on specific objects were automatically identified after all data collection was completed. Fixation labels were manually checked in order to verify that the eye-tracker was performing within tolerance. The simulation also logged all mouse clicks, indicating (when appropriate) what object was clicked on at different times.

All instances of changing a UAV's target were manually extracted from the simulation. A total of 200 sequential process traces on this subtask, with eye fixations and mouse clicks listed in chronological sequence, were created by these participants. Collecting, extracting, verifying, and labeling the data took, conservatively, 80 hours.

4 ACT-R/E

We chose the cognitive architecture ACT-R/E to model human performance on the RESCHU task because we needed a cognitive architecture that captures human behavior at a fine-grained level of analysis (goals, eye fixations, physical actions, etc.), and because it has a long history of providing detailed process descriptions of how people perform tasks [2]. Additionally, it can easily model different strategies and can generate large amounts of data with human-like variability, an important component for training predictive models [15]. Note, though, that our methodology should work with any cognitive architecture, depending on the task and the level of granularity desired.

ACT-R/E (Adaptive Character of Thought- Rational/Embodied) is a hybrid symbolic/sub-symbolic production-based system based on ACT-R [31]. For the purposes of this report there are no critical differences between ACT-R and ACT-R/E. An ACT-R/E model is, essentially, a set of if-then rules that make requests and access information in the model's working memory (which is designed to reflect people's working memory). Depending on the current contents of working memory, different if-then rules can fire, producing behavior and potentially changing the contents of working memory to encourage other if-then rules to fire. These sequences of firing if-then rules capture the process of human cognition and, via the contents of working memory, are influenced by what the model sees, knows and does (just like people are).

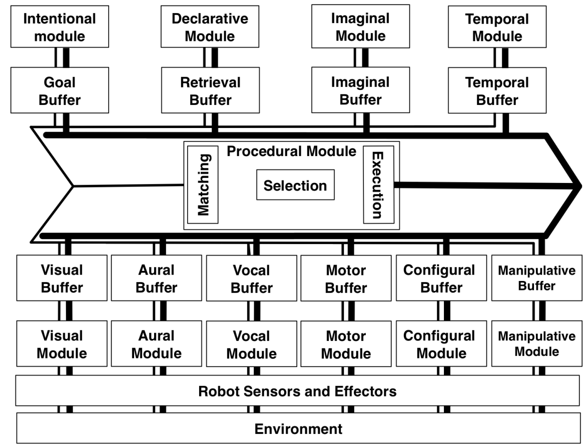

Figure 2: An architecture diagram of ACT-R/E.

More technically, ACT-R/E consists of a number of modules, buffers (that collectively represent working memory), and a central pattern matcher. Modules contain a relatively specific cognitive faculty associated with a specific region of the brain. For each module, there are one or more buffers that communicate directly with that module as an interface to the rest of ACT- R/E. At any point in time, there may be at most one item in any individual buffer; thus, the module's job is to decide what and when to put an object into a buffer. The pattern matcher uses the contents of the buffer to match specific productions (if-then rules). ACT-R/E interfaces with the outside world through the visual module, the aural module, the motor module, and the vocal module. Other current modules include the intentional, imaginal, temporal and declarative modules. ACT-R/E perceives the physical world by either robotic or virtual sensors [31]. ACT-R/E's goals are to maintain cognitive plausibility as much as possible while providing a functional architecture to create models of human-level intelligence.

We discuss the modules that are especially relevant to this project below. Figure 2 shows a schematic of ACT-R/E, which is discussed more fully in [31].

4.1 The Intentional Module

The intentional module is used to set, change, track, and remove goals. Like people, ACT-R/E does not enforce a strong goal order or goal-stack [1] and enables both top-down and bottom-up goal execution.

4.2 Declarative Module

The declarative module provides a method to encode, store, and retrieve items from long-term memory into working memory. This helps support, for example, step (4) above of visually acquiring an object, below. Because people don't always retrieve the correct memory, it is critical in allowing cognitive models to capture how long a specific memory takes to remember and the likelihood that a specific memory is correctly retrieved. This provides another important source of cognitively-plausible variability.

4.3 The Visual Module

The visual module is used to provide a model with information about what can be seen in the current environment. This module is what finds objects in the world that, for example, are needed to perform an action. Critically, numerous empirical studies have shown that there is a specific set of perceptual and declarative steps that need to occur in order for a person to recognize, understand, and identify an object in the world in order to take action on it [7]:

(1) People set a goal to search for specific features (e.g., color or shape);
(2) People move their attention to objects that have those features;
(3) People make a saccade to that object and identify it visually (e.g., provide an internal label);
(4) People make memory retrievals in order to interpret what that object is in the current context

Additionally, because perceptual attention is rarely on the same thing for an extended period of time, there is a great deal of variability where people "look around" in environments, especially dynamic ones. ACT-R/E is also able to capture this variability in a cognitively plausible way.

4.4 Motor Module

ACT-R/E's motion is controlled by the motor module. In this work, motor is used to control physical actions (e.g., mouse movement and clicks). It can be used to provide cognitively-plausible mouse movements including timing and variability.

4.5 Model Trace Logging

Given an ACT-R/E model of a task, we can easily and readily execute that model to generate synthetic observational data labeled with observations, actions, and the goals behind those actions. Recall that when a model is executed in the surrounding architecture, behavior is produced by firing if-then rules matching against the fluctuating contents of working memory. At each step of the way, the architecture can log what is occurring. It can log, for example, what the model saw as it executed (such as by looking at the visual module's fixations), and what it did (such as by recording actions taken by the motor module). Each time it logs these events, it can also record the goal driving the events by inspecting the current state of the intentional module. As we show below, this execution and logging can happen very rapidly, allowing us to quickly generate multiple logged sequential process traces of simulated human performance a model's task.

5 MODEL DESCRIPTION

In this paper, we focus on an important component of the RESCHU task: changing a UAV's target. As discussed earlier, this action occurs for a variety of reasons, including avoiding a hazard area, or changing to a more efficient target. There were several different interface methods for changing a UAV's target, as described above. Predicting what action a person is about to take (i.e., what target a user will send a UAV to next) could allow the system to facilitate the interaction by pre-selecting the object, or by suggesting a better one.

People also use different cognitive strategies to change a UAV's target. The two strategies that we model here are a **planning** strategy and an **opportunistic** strategy. Most people begin with a planning strategy and then switch (sometimes frequently) to an opportunistic strategy, especially in dynamic tasks [14, 29].

We next describe these two strategies, as well as our cognitive models of each of them. We describe them here generally; for those interested, we can share the model code upon request. Additionally, note that these models were not complicated or difficult to build: they relied on the architecture and common known programming idioms for cognitive architectures.

5.1 Planning Strategy

The planning strategy captures that people sometimes plan a few actions ahead, or search for the best action to do, before performing the steps of action. For the goal of changing a UAV's target, the ACT-R/E model of this strategy uses the intentional module to organize its actions and drive its behavior. The model first searches for a UAV whose target needs to be changed (because it is on a collision course with a threat or needs to move to a closer target, etc.), primarily by leveraging the visual module to see and the declarative module to interpret what it sees. It then has to hold this target in its working memory while continuing on to search for an appropriate new target (e.g., one that does not not intersect a hazard or that is closer). The model then makes the appropriate interface actions (e.g., clicking on the UAV, selecting a change-target from a drop-down menu, and then clicking on the desired target) using the goal and motor modules. The model additionally searches for other threats and checks other UAVs throughout the execution of the action.

5.2 Opportunistic Strategy

The opportunistic strategy recognizes that people sometimes do not plan ahead, and just take the best or convenient action in that moment. The ACT-R/E model of this strategy still has the goal, via the intentional module, to re-route the UAV; however, it sequences its actions differently. It first searches for a UAV whose target needed to be changed, and then clicks to select the UAV without a specific target yet in mind, using the visual, declarative and motor modules. Next, the model searches for a target where the UAV could be sent using the visual and declarative modules. After the appropriate target is found, it clicks on it to change the UAV to go to that target using the motor module. The model additionally searches for other threats and checked other UAVs throughout.

The differences between the two models are subtle, but they reflect different cognitive strategies of accomplishing the human's goals. These differences arise from human's working memory constraints: the planning model requires searching for a target while remembering which UAV needs to be selected, while the opportunistic models requires less memory but is less effective. These subtle differences in strategy allow the models to collectively capture the variability of human behavior while executing the task. A critical aspect of both models is that there is enough simulated time to execute both strategies within the constraints of the task.

5.3 Synthetic Data Generation

With the two strategy models in hand, we are able to use them to generate variable synthetic data of humans performing the task. Critically, these models can generate traces of observational data and actions that were identical in form to the traces that were generated from the human participants, including eye fixations and mouse clicks listed in chronological order. The models can additionally attribute eye fixations and mouse clicks to the eventual action that was driving that behavior (e.g., its eye gaze was fixating on a specific target because it was considering sending a UAV to it). Further, because of the way that the models accommodate human variability, they can generate synthetic data reflecting that variability, such as with noisy eye fixations, reaction times, etc. All together, both the planning model and the opportunistic model were each run 20,000 times to generate 20,000 individual, distinct traces of synthetic human performance for each strategy. These traces were stored in computer files alongside the saved traces from the empirical data until they were preprocessed for the deep neural network, described next.

6 DEEP NEURAL NETWORK

To predict a human's next action, we turn to deep neural networks as a common and effective way of performing statistical prediction.

Our goal was to take sequential process traces from either the ACT-R/E models or empirical data and output a prediction of the user's upcoming action (here, again, which target that the user is going to send a selected UAV to).

6.1 Model Inputs and Outputs

To begin, we converted each trace into a padded 200 length vector. The vector consisted of each individual eye fixation and mouse click in the trace. For example, "fixation-uav4", "fixation-tar2", "fixation-carrier", "action-select-uav1" (representing looking at UAV4, then target 2, then the carrier, and then clicking on UAV1) could be part of a vector that was fed into the network. The entire vector was the input into the network. The predicted action of interest was not, of course, part of the input vector.

The model then provided an output prediction of which target was about to be selected.

6.2 Model Architecture

The overall architecture of the network is shown in Figure 3. The first layer of the network, an embedding layer, generates a fixed-length numerical representation of the input vector. For each value of the vector, the layer outputs a matrix of floating points where each original item in the vector is represented by a 5-dimensional

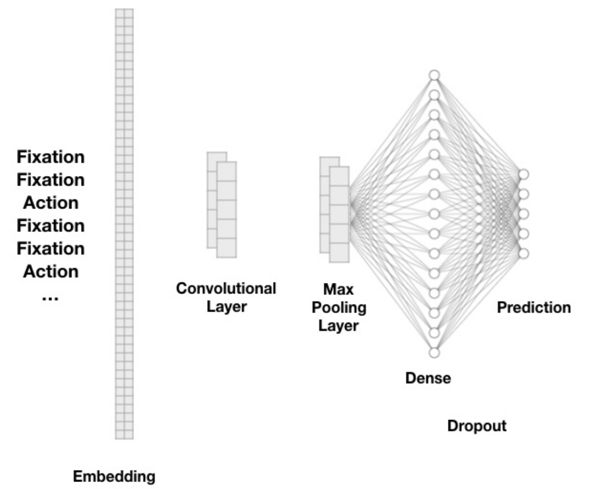

Figure 3: The predictive deep network architecture. It takes, as input, vectors of consecutive eye fixations and mouse clicks (actions); it outputs predictions of which target the user is about to click on in order to send the UAV to it.

embedding vector. This simplifies the computations and representations needed, and allows for relationships across items in the input vector.

The embedding layer is followed by a 1 dimensional convolutional layer, a max pooling layer, and a fully-connected 30-unit layer with a 20% dropout rate. The final layer was a fully-connected output layer with a softmax activation to optimize the action with the highest predicted value.

Note that a recurrent neural network was considered as an alternative to the current architecture; because the structure of the problem was not a next-item prediction, but rather an accumulation of evidence prediction, the model above was a more appropriate choice than a recurrent architecture.

7 EXPERIMENTS AND RESULTS

We developed our experiments to answer the following series of questions. First, can a cognitive process model be used to facilitate generating predictive models of human behavior? Second, how effective were the different strategies in capturing the variability of human behavior, both separately and together? Finally, does combining the model and empirical data increase the overall predictive power of the model versus training with one or the other alone?

To this end, we train deep networks under several different conditions:

- **Baseline:** The limited empirical data available was used to train the model.
- **Planning:** All synthetic data generated by the planning strategy model was used to train the model.
- **Opportunistic:** All synthetic data generated by the opportunistic strategy model was used to train the model.
- **Combined:** Synthetic data from the planning strategy and opportunistic strategy models were combined and used to

train the model. In order to keep the amount of training data constant across model-training scenarios, half of the planning data and half of the opportunistic data was used in all combined conditions.

Planning+Empirical: All synthetic data generated by the planning strategy model, as well as the the empirical data, were used to train the model.

Opportunistic+Empirical: All synthetic data generated by the opportunistic strategy model, as well as the empirical data, were used to train the model.

Combined+Empirical: Synthetic data from the planning strategy and opportunistic strategy models were combined and, along with the empirical data, were used to train the model.

For all models, 10-fold cross validation was used to divide the empirical data into training and testing data. All conditions used the same folds for training and testing, and all models were evaluated on the empirical data.

7.1 Results

Recall that correct performance is when the model chooses the same target chosen by the user. We consider four different baselines for comparison.

The first two baselines we considered are simple heuristics for predicting a person's next action that did not require explicit models. One possible naive heuristic that could predict what target was acted upon could be "the last target looked at will be the one acted upon." However, this does not seem to be the case: only 30% of the time was the last target actually selected as the UAV's next goal. Another possible heuristic could be "the target looked at the most will be acted upon." This also does not seem to be the case: only 23% of the time was the most frequent target selected as the UAV's next goal. Unsurprisingly, even on a moderately complex dynamic task, simple heuristics can not predict final actions.

The third baseline we considered was that of the baseline deep network trained with only empirical data. Its performance was not particularly strong; it selected the correct target for the UAV 40% of the time. While this was better than the simple heuristics above, it is still not adequate. The reason this baseline deep network did not perform well was almost assuredly because of the small amount of available empirical data (as suggested above and discussed more below).

The final baseline we considered was a naive Bayes model, which can perform well on more limited data. The accuracy of the naive Bayes model was, in fact, than the other baselines at 55% accuracy. The naive Bayes model is not, however, as good as the combined models described below.

The overall results are shown in Figure 4, including the baseline deep network as the left-most bar and the better simple heuristic shown as a horizontal line. This shows the percentage of test examples where the different deep networks correctly predicted the target that the user would next assign to the UAV. We discuss the results with respect to the questions above.

The most basic and important question is whether our overall methodology – using a cognitive process model to generate synthetic data that is then used to train a deep network – can be used

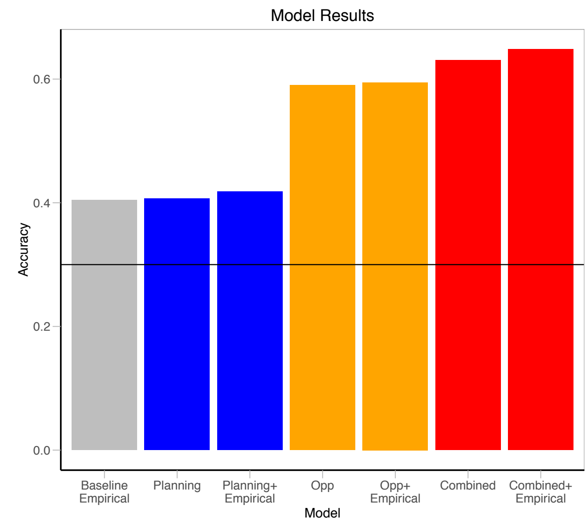

Figure 4: Predictive model results. The X-axis indicates DL models trained with different types of empirical and synthetic data. The Y-axis is the trained model's accuracy on the test data; the horizontal line is performance from a naive approach. Details of each model condition, and the baseline, are in the text.

to predict human behavior. As Figure 4 suggests, all synthetically-trained models are superior to the empirically-trained model. Because predicting human behavior remains extremely challenging for AI and cognitive science, this first result is encouraging.

We next consider the success of the cognitive model in capturing people's different strategies for the task. Both the opportunistic-trained model and the planning-trained model were better than the pure empirical-trained model, though the opportunistic model perform much better than the planning model. This indicates that we have successfully identified and modeled strategies that people use on this task.

To see whether combining strategies increases the power of the model, we next consider results where data from both strategies is used to train the deep network (using the same amount of total data). The results show that the combined model that used both planning and opportunistic synthetic data to train the network performed better than either strategy alone. This indicates that, for complicated decision-making tasks, capturing different strategies people use is a key facet of giving the statistical models enough variability to capture user behavior and predict future actions.

The last critical question concerns combining some of the empirical data with the synthetic data. This is of particular interest because it is possible that the ACT-R models only approximate the true distribution of what people actually do during this task; additional strategies may have been missed; the modeling framework may be lacking something, etc. One way to deal with this possibility is to integrate a portion of the empirical data with the synthetic data. As Figure 4 shows, integrating empirical data and synthetic data did slightly increase prediction accuracy for all three strategy conditions, indicating that including a little empirical data with the synthetic data can further boost performance. Statistically, the Combined+Empirical model performed better than all other models ($p < 0.05$) except the Combined model, where it performed marginally better ($p < 0.10$).

8 DISCUSSION

We have introduced a novel system architecture that combines cognitive and statistical AI models to predict human behavior from limited amounts of data. The strength of our approach relies on using a model that captures internal processes (cognitive models) to generate data for a model that can predict behavior using outward observations (deep networks). We used a computational cognitive architecture (ACT-R/E) to model the internal states and different strategies that people used in a dynamic supervisory control task. The cognitive model was able to generate a large amount of diverse data (traces of its behavior on the task) that could then be used to train a deep network. The deep network was then able to predict actions that a human will soon take as they perform their task. We found that deep networks trained on synthetic data representing only one strategy for completing the task was able to outperform deep networks trained only on the limited empirical data; networks trained on multiple strategies, however, far outperformed both of those conditions.

Earlier we discussed several challenges that arise when predicting human actions with relatively little empirical data. We revisit these challenges below and describe how our work resolves each of them.

Lack of data challenge: In this task, as in many other tasks of human behavior, we had very little human data. We were able to augment the data we had by building high fidelity computational cognitive models to generate large amounts of synthetic process traces. These process traces were identical in form to the empirical data, which allowed us to integrate them together. Note that there are other methods of data augmentation to address this problem, but most of them are not applicable to human behavioral tasks. For example, rotating and cropping images to increase the training set does not work on individual sequential process traces. Nor does performing large-scale resampling and combinatorials to increase the data. These approaches struggle with sequential process traces we consider here because there is a strong structure to the individual components in the sequence. The structure is generated naturally by a cognitive model, but more arbitrary methods of combining or mixing them do not seem to be successful.

Labeling challenge: We used a mix of automatic and manual methods to label and validate the empirical data (eye fixations, actions, etc.). The model data, however, did not have to be manually labeled at all, because the labels are generated automatically as part of the process traces of the cognitive model. This automated labelling could also potentially help to determine the impact of different strategies on human performance, or to perform more detailed explanations of human behavior.

Strategy challenge: Limited amounts of empirical data typically do have enough coverage for the diverse strategies that people can bring to bear on sequential action tasks. Cognitive models, however, can model those strategies and generate arbitrary amounts of data for each. We found that using data from different strategies allowed a better predictive model to be built: not only were different strategies present in human behavior, but the model was able to appropriately use the features and representations it learned from being trained on both strategies.

Balanced data challenge: With relatively little empirical data that has high variance, it can be difficult to balance training data so that the model learns close to the true distribution. Generating arbitrary amounts of synthetic data, that is variable in the way that people are variable solves this problem. This is because the data is generated based on the true distribution of human behavior, while also providing enough data to be ensure that the predictive model learns the distribution.

8.1 Future work

There are several components of this methodology that could be improved or explored further. For example, in terms of overall performance, we may be able to improve prediction by adding additional strategies. We could also explore why data generated from the planning strategy was relatively unsuccessful compared to data from the opportunistic strategy. Both of these actions would help us better understand how strategies, in general, cover the span of human behavior.

The current cognitive models capture a relatively small (though critical) set of the actions people take when completing supervisory control tasks. In future work, we plan to expand and generalize this work to include more portions of the task tree. One way to build more of the task tree would be to build cognitive models for additional actions and then build a single deep network with an integrated task model. Alternatively, explicitly switching between different networks for task components, or giving them a hierarchical structure, may be necessary depending on the similarity or complexity of the model.

Finally, we plan to use the results of the deep network to facilitate human actions. For example, in the supervisory control task, if the network predicts that the person is searching for a UAV, highlighting all the UAVs could be very useful for overall performance. Similarly, the cognitive model's goals could be used to increase explainability or transparency by showing the human what goal it believes they are working on.

9 ACKNOWLEDGMENTS

This work was supported in part by ONR to GT and LH. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the US Navy.

REFERENCES

[1] E. M. Altmann and J. G. Trafton. 2002. An Activation-Based Model of Memory for Goals. *Cognitive Science* (2002), 39–83.
[2] John R Anderson. 2009. *How can the human mind occur in the physical universe?* Vol. 3. Oxford University Press.
[3] John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. 2004. An integrated theory of the mind. *Psychological review* 111, 4 (2004), 1036.

[4] David D Bourgin, Joshua C Peterson, Daniel Reichman, Thomas L Griffiths, and Stuart J Russell. 2019. Cognitive model priors for predicting human decisions. *arXiv preprint arXiv:1905.09397* (2019).

[5] Yves Boussemart and ML Cummings. 2008. Behavioral recognition and prediction of an operator supervising multiple heterogeneous unmanned vehicles. *Humans operating unmanned systems* (2008).

[6] Leonard A Breslow, Daniel Gartenberg, J Malcolm McCurry, and J Gregory Trafton. 2014. Dynamic operator overload: A model for predicting workload during supervisory control. *IEEE Transactions on Human-Machine Systems* 44, 1 (2014), 30–40.

[7] M. D. Byrne and J. R. Anderson. 1998. Perception and action. In *Atomic Components of Thought*, J. R. Anderson and C. Lebiere (Eds.). Lawrence Erlbaum, Mahwah, NJ, 167–200.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[9] Hilda M Fehd and Adriane E Seiffert. 2008. Eye movements during multiple object tracking: Where do participants look? *Cognition* 108, 1 (2008), 201–209.

[10] Mary L Gick. 1986. Problem-solving strategies. *Educational psychologist* 21, 1-2 (1986), 99–120.

[11] Alec Go, Richa Bhayani, and Lei Huang. 2016. Sentiment140. *Site Functionality, 2013c. URL http://help. sentiment140. com/site-functionality. Abruf am* 20 (2016).

[12] Patrick J Grother. 1995. NIST special database 19. *Handprinted forms and characters database, National Institute of Standards and Technology* (1995).

[13] Glenn Gunzelmann, Joshua B Gross, Kevin A Gluck, and David F Dinges. 2009. Sleep deprivation and sustained attention performance: Integrating mathematical and cognitive modeling. *Cognitive science* 33, 5 (2009), 880–910.

[14] Barbara Hayes-Roth and Frederick Hayes-Roth. 1979. A cognitive model of planning. *Cognitive science* 3, 4 (1979), 275–310.

[15] Laura M. Hiatt, Anthony M. Harrison, and J. Gregory Trafton. 2011. Accommodating Human Variability in Human-Robot Teams through Theory of Mind. In *Proceedings of (IJCAI)*.

[16] Laura M. Hiatt, Cody Narber, Esube Bekele, Sangeet S. Khemlani, and J. Gregory Trafton. 2017. Human modeling for human-robot collaboration. *International Journal of Robotics Research* 36, 5-7 (2017), 580–596.

[17] Bonnie E John and David E Kieras. 1996. The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction (TOCHI)* 3, 4 (1996), 320–351.

[18] Sangeet Khemlani and Philip N Johnson-Laird. 2012. Theories of the syllogism: A meta-analysis. *Psychological bulletin* 138, 3 (2012), 427.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[20] John E Laird. 2012. *The Soar cognitive architecture*.

[21] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science* 350, 6266 (2015), 1332–1338.

[22] Jill Larkin, John McDermott, Dorothea P Simon, and Herbert A Simon. 1980. Expert and novice performance in solving physics problems. *Science* 208, 4450 (1980), 1335–1342.

[23] Yann LeCun, Corinna Cortes, and CJ Burges. 2010. MNIST handwritten digit database. *AT&T Labs* 2 (2010), 18.

[24] Takaki Makino and Johane Takeuchi. 2012. Apprenticeship Learning for Model Parameters of Partially Observable Environments. In *Proceedings of the 29th International Conference on Machine Learning*. icml.cc / Omnipress, 117.

[25] Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis. 2010. Running experiments on amazon mechanical turk. *Judgment and Decision making* 5, 5 (2010), 411–419.

[26] Roger Ratcliff, Philip L Smith, Scott D Brown, and Gail McKoon. 2016. Diffusion decision model: Current issues and history. *Trends in cognitive sciences* 20, 4 (2016), 260–281.

[27] Paul S Rosenbloom. 2013. The Sigma cognitive architecture and system. *AISB Quarterly* 136 (2013), 4–13.

[28] Jule Schatz, S. J. Jones, and John E. Laird. 2018. An Architecture Approach to Modeling the Remote Associates Test. In *Proceedings of the 16th International Conference on Cognitive Modelling (ICCM)*.

[29] Colleen M Seifert, David E Meyer, Natalie Davidson, Andrea L Patalano, and Ilan Yaniv. 1994. Demystification of cognitive insight: Opportunistic assimilation and the prepared-mind hypothesis. (1994).

[30] Hidetaka Taniguchi, Hiroshi Sato, and Tomohiro Shirakawa. 2018. A machine learning model with human cognitive biases capable of learning from small and biased datasets. *Scientific reports* 8, 1 (2018), 1–13.

[31] J. Gregory Trafton, Laura M. Hiatt, Anthony M. Harrison, Franklin P. Tamborello, II, Sangeet S. Khemlani, and Alan C. Schultz. 2013. ACT-R/E: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction* 2, 1 (2013), 30–55.

[32] Susan Bell Trickett, J Gregory Trafton, Lelyn Saner, and Christian D Schunn. 2007. "I Don't Know What's Going On There": The Use of Spatial Transformations to Deal With and Resolve Uncertainty in Complex Visualizations. In *Thinking with data*. Psychology Press, 81–101.

[33] Amin Ullah, Jamil Ahmad, Khan Muhammad, Muhammad Sajjad, and Sung Wook Baik. 2017. Action recognition in video sequences using deep bi-directional LSTM with CNN features. *IEEE Access* 6 (2017), 1155–1166.

[34] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine* 13, 3 (2018), 55–75.