A Cognitive Model of Feature Selection and Categorization for Autonomous Systems

Michael Martin[1], Christian Lebiere[1], MaryAnne Fields[2] and Craig Lennon[2]

[1]Carnegie Mellon University, Pittsburgh, PA, USA
mkm@andrew.cmu.edu,cl@cmu.edu
[2]Army Research Laboratory, Aberdeen, MD, USA
mary.a.fields22.civ@mail.mil,craig.t.lennon.civ@mail.mil

Abstract

Abstract. We describe a computational cognitive model intended to be a generalizable classifier that can provide context-based feedback to semantic perception in robotic applications. Many classifiers (including cognitive models of categorization) perform well at the task of associating features with objects. Underlying their performance is an effective selection of the features used during classification. This Feature Selection (FS) process is usually performed outside the boundaries of the model that learns and performs the classification task, often by a human expert. In contrast, the cognitive model we describe simultaneously learns which features to use, as it learns the associations between features and classes. This integration of FS and class learning in one model makes it complementary to other Machine-Learning (ML) techniques that automate the FS process (e.g., deep learning methods). But their integration in a cognitive architecture also provides a means for creating a dynamic context that includes disparate sources of information (e.g., environmental observations, task knowledge, commands from humans); this richer context, in turn, provides a means for making semantic perception goal-directed. We demonstrate automated FS, integrated with an Instance-Based Learning (IBL) approach to classification, in an ACT-R model of categorization by labeling facial expressions of emotion (e.g., happy, sad) from a set of relevant, irrelevant, distinct, and overlapping facial action features.

Keywords: autonomous systems, feature selection, ACT-R, cognitive architectures, machine-learning, classification, categorization, instance-based learning

1 Introduction

Robots tend to process information with perceptual systems that feed forward to cognitive systems that do something intelligent (e.g., planning, reasoning). Thus, cognitive systems tend to provide little to no feedback to autonomous perception. We assume that autonomous perception can be improved by establishing a feedback loop between perceptual and cognitive systems. The feedback loop would make autonomous perception more of an interactive process than it is today by providing a means for exploiting cognitive context (e.g., goals) to augment perceptual processes dominated by stimulus-driven information.

Semantic perception in autonomous systems is a complex process that generally involves parsing sensor data into features, grouping features into likely objects, and then assigning labels to objects. Semantic perception thus anchors recognized objects in the environment to labels that refer to those objects. The labels are placed in semantic maps for data interchange, and then passed to cognitive systems that can operate on the symbols.

More often than not, semantic perception refers to a computer vision system with independently trained object detectors. These object detectors tend to operate on local spatial features in the images they process. This means anchoring, the process of grounding semantic labels in sensor data, is frequently based on a local, stimulus-driven context that varies because of noisy sensor data, occlusion, lighting conditions and so on. To reduce classification error, some detectors employ additional context obtainable from global image features such as intensity, predominant color, etc. Others incorporate expectations about objects based on their location within a scene.[1]

Including more context in autonomous perceptual systems is one approach to improving anchoring. Another approach is to get additional context from cognitive systems and iteratively integrate it with stimuli. This is the approach we have been exploring. The basic idea is to create a general model of perception in a cognitive architecture (i.e., a computational instantiation of a unified theory of cognition) that can encode features or labels from object detectors; combine that environmental context with task-relevant information; and then provide the resulting cognitive context (e.g., expectations, points of ambiguity, points of interest, etc.) in a feedback channel that can be used by perceptual systems to augment accuracy (and efficiency).

Our efforts thus far are focused on the feedforward signal from semantic perception to cognition. As a first step, Fields et al. [1] addressed the issue of whether a cognitive architecture (ACT-R) can encode the kinds of information provided by semantic perception. In addition to object labels, semantic maps may include

---

[1] In computer vision there are both local features such as edges, Scale-Invariant Feature Transform (SIFT), or Speeded Up Robust Features (SURF); and global features such as parts, Histogram of Oriented Gradients (HOG), textures or contours. We refer to all of these as local features since they focus on the object and not the environment.

feature information and continuous variables (e.g., confidence, intensity). Fields et al. [1] demonstrated that Instance-Based Learning (IBL) in ACT-R performs similarly to a k-Nearest Neighbors classifier, while ACT-R's partial matching mechanism supports classification using continuous features.

Fields et al. [1] also demonstrated that adding goal-directed meaning to scene recognition processes improves accuracy. When a robot's goal is to classify public spaces (e.g., conference rooms vs dining rooms), performance improves by using features representing the general notion of social immediacy rather than counts of common objects (i.e., chairs and tables) found in the rooms. Thus the goal-directed selection of features improved recognition performance. The categorization model described below (CAT) performs goal-directed FS as our next step toward a mechanism that supports context-sensitive semantic perception.

2 Approach

CAT is an ACT-R classifier that simultaneously learns: (1) associations between features and classes, and (2) which features to encode for which classes. In ML terms, it combines an IBL paradigm for the classification problem with Reinforcement Learning (RL) for the FS problem. IBL allows CAT to use memory of past examples of a class to generalize to novel members of the class (for a discussion of IBL theory in ACT-R, see Gonzalez et al. [2]). RL allows CAT to select features as a function of their effectiveness. The domain-independence of RL and IBL means CAT may generalize across classification problems; allowing interactions with perceptual classifiers regardless of the classes or features involved. Indeed, the approach generalizes to any decision process that includes decision instances and feedback.

2.1 A Cognitive Theory of Categorization

In terms of cognitive theory, CAT maps mechanisms of categorization phenomena in humans to mechanisms in ACT-R described as underlying cognitive phenomena in general (e.g., [3]). Anderson and Betz [3] mapped the Exemplar-Based Random Walk (EBRW) model of categorization [4] onto ACT-R mechanisms. Here, we describe CAT as a comparable mapping of the EBRW Response Time (EBRW-RT) model of categorization [5], even though we initially developed CAT based on the computational mechanisms of the ACT-R framework. EBRW-RT extends EBRW from a model focused only on class decision processes to one that includes feature-encoding processes.

EBRW assumes features are encoded in a single step because it is concerned with classification tasks that involve integral features, which are encoded simultaneously. EBRW-RT extended EBRW to classification tasks that involve separable features, which are encoded sequentially. Thus, while EBRW and EBRW-RT are similar information accumulation models, EBRW-RT associates the random walk process of EBRW with iterative feature-encoding processes. Accordingly, features are encoded one at a time until a category decision is made. The decision about category membership depends on similarity between an evolving stimulus representation and instances in memory.

2.2 ACT-R Implementation

Cognitive architectures provide a software implementation of cognitive theory. As software, ACT-R can be viewed as a set of asynchronous modules (e.g., memory, perceptual, motor) that store and process information, plus a production system that stores or retrieves information from buffers associated with each module. Creating a cognitive model of a task involves writing a set of productions that performs the steps required by the task. Productions encode condition-action rules that perform actions on the buffers anytime they match conditions on the contents of the buffers. Each buffer contains at most one chunk of information, where a chunk is a data structure with a set of slot-value pairs.

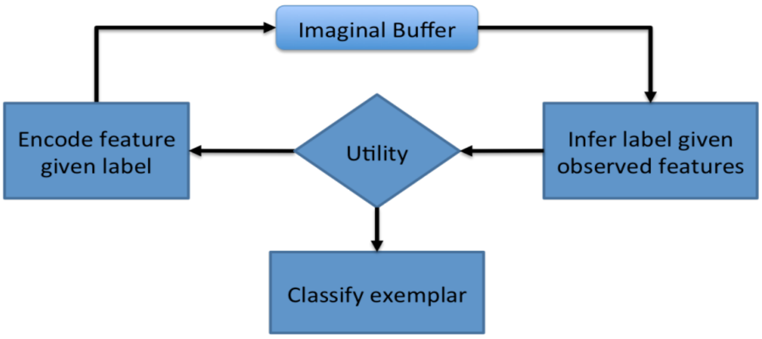

Fig. 1. The general structure of the categorization model

Figure 1 depicts the general structure of CAT. Three types of productions drive the classification process: feature encoding, class inference, and decision. Feature-encoding productions take a single feature (e.g., from a semantic map) and place it in the imaginal buffer used to construct the object representation. Configuration chunks in the imaginal buffer represent the associations among features and classes, having a slot for each feature encoded and the class label. Due to the serial firing of feature-encoding productions, stimulus information accumulates in the imaginal buffer.

Class-inference productions use the context provided by observed features in the imaginal buffer to retrieve class labels that have been associated with those features in the past. That is, observed features serve as a cue for finding a configuration chunk in declarative memory (i.e., the declarative module) that approximately matches current observations but also has a class label. Retrieving such a chunk amounts to an inference about what the class label should be (i.e., a class hypothesis), given the features observed thus far in the classification process.

The firing of a class-inference production initiates an exemplar retrieval process. Activation (see Equation 1), a numerical quantity associated with each chunk in ACT-R Declarative Memory (DM), drives the retrieval process. Activation spreads from features in the imaginal buffer to configuration chunks in DM that share one or more of the observed features; chunks are activated by the degree to which they share features, plus a random noise component, ε. The configuration chunk with highest activation is retrieved.

$$\mathrm{A_i} = \sum_{\mathrm{k=1}}^{\mathrm{K}} W_k S_{ki} + \varepsilon \tag{1}$$

A decision production terminates the cycle of feature-encoding and class-inference productions firing over time and assigns a class label to the stimulus. CAT's process of choosing between competing productions involves a numeric quantity associated with each ACT-R production called utility. Utility reflects a production's usefulness for achieving a goal - stimulus classification in this case. In cases where multiple productions' conditions match the context provided by buffers, the production with the highest utility will fire. Therefore, conflict resolution is based on a production's utility value, including a random noise component (see Equation 2), which promotes response variability and exploration of decision strategies. In CAT, the decision production competes with feature-encoding productions. Thus, the information accumulation process terminates when the utility of a decision production exceeds the utility of all feature-encoding productions.

$$Pr_i(i) = e^{U_i/\sqrt{2s}} / \sum_j e^{U_j/\sqrt{2s}} \tag{2}$$

Production utilities can be learned from experience by reinforcing productions that lead to successful task performance. RL involves the distribution of rewards in the production system. The reward received by a production is scaled based on the time between the distribution of a reward and when a production fired. Rewards are temporally discounted so that productions that fired immediately before reward distribution receive more than productions that fired in the more distant past. Temporal discounting is necessary because rewards are distributed to all productions that have fired since the last time rewards were distributed. The reward is then used to gradually adjust utility until it matches the average reward received (see Equation 3).

$$U_i(n) = U_i(n-1) + \alpha[R_i(n) - U_i(n-1)] \tag{3}$$

where $U_i(n)$ is the utility of production *i* after its *n*th firing and $R_i(n)$ is the reward received by production *i* after its *n*th firing.

Implementing the classification process described above in an IBL process supports the bootstrapping of class knowledge from experience, and provides a hook for controlling the distribution of rewards. That is, IBL involves feedback about model performance, which can be used to distribute rewards too. We thus added three feedback productions to CAT: one distributes rewards for a correct classification; one distributes rewards for an incorrect classification and corrects the class slot in the imaginal buffer; and one harvests (stores) the configuration chunk in the imaginal buffer as an instance in DM.

Feedback productions fire after a class decision production fires, and create an instance in DM that associates observed features and (corrected) class labels. Simultaneously, the utility of a subset of feature-encoding and decision productions is adjusted. As the utilities of these competing productions evolve, the order and number of features encoded will vary.

CAT, however, will have no effective way to learn a classification task in which some features are important signs of one class but not so important for others because, as described above, productions encode features without regard to the current class hypothesis. We avoided this problem by including the class hypothesis as an additional constraint in the condition side of feature-encoding productions. This constrains the distribution of rewards in a more task-relevant manner by drawing a distinction between encoding a particular feature in the context of one class hypothesis and encoding that same feature in the context of another class hypothesis. Thus expectations, in the form of class hypotheses, change the way CAT encodes the world and subsequently parses it into categories.

3 Testing and Revision

We tested CAT using the publicly available Cohn-Kanade facial expression dataset [6]. The dataset associates emotion labels for expressions from 123 participants with facial Action Units (AUs). We used 327 exemplars of 7 elicited expressions (see Figure 2). Their associated AUs are based on Ekman's Facial Action Coding System [7, 8], which separates the observation of facial actions from inferences about emotional state.

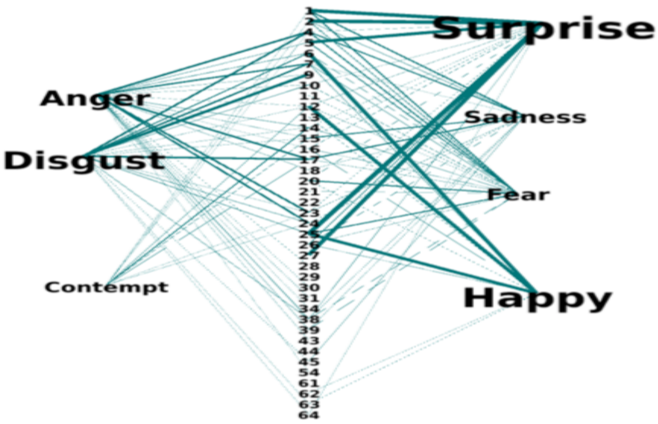

Fig. 2. The relationship between features and expressions

We removed 25 AUs that were constant across the exemplars we used, leaving a total of 39 AUs. Figure 2 shows the association of the remaining AUs (as numbers in the center) with emotion classes (as words on the sides). Font size of the words codes class frequency, whereas the weight of the line connecting emotion to AU codes co-occurrence frequency. Notice that the base rates of the classes differ; features differ among classes; the number of relevant features varies between classes; and classes share features.

3.1 Testing Procedure

Revisions to CAT were based on a simple testing procedure in which the 327 exemplars were randomized, and then presented to the model one by one. For each exemplar, CAT makes a classification decision and receives feedback about the exemplar's true class. This procedure was repeated for a total of 10,000 trials to examine learning over an extended period. Unless otherwise noted, global ACT-R parameters were left at their default settings.

3.2 Results and Incremental Refinement

The learning curves in Figure 3 depict classification accuracy for the Canonical CAT model described above, plus four, cumulative revisions. The points plotted for each line represent the proportion of correct responses in a sliding window of 1000 trials. The proportions are right-aligned so that the mean of the first 1000 trials is plotted with an x-coordinate of 1000, the mean of the second window as 1001, and so on. The low amplitude oscillations from trial to trial reflect noise in the retrieval of chunks from DM used for classification and the firing of relevant productions in procedural memory used for feature encoding.

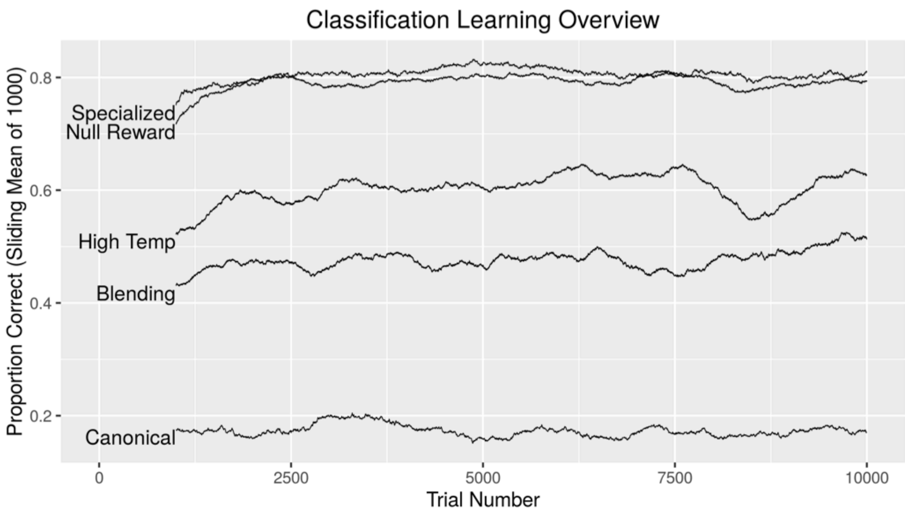

Fig. 3. A comparison of learning results for all models.

The Canonical CAT model reflects the basic IBL and RL processes in ACT-R (see Figure 3). As can be seen, the Canonical model exhibits near chance-level classification performance and no long-term learning trend.

The Blending model uses the ACT-R blending buffer rather than its retrieval buffer to access DM. All other aspects of the Canonical model remained the same. The blending module provides a mechanism for constructing chunks from relevant past experience, in contrast to the retrieval module, which returns the most relevant chunk. It was developed for tasks that require quantitative responses (e.g., magnitude estimation) to support a form of interpolation between slot values found in relevant chunks in DM [9]. Thus the slot-values V in the constructed chunk are synthesized from slot-values V_i in corresponding slots across multiple DM chunks in some situations by minimizing their dissimilarity, weighted by the retrieval probability P_i of each chunk (see Blending Equation 4). That retrieval probability is similar to the utility selection (2), with activation playing the role of utility, scaled by a temperature parameter.

$$V = argmin \sum_i P_i \cdot Sim^2(V, V_i) \qquad (4)$$

In terms of categorization theory, the use of the blending mechanism changes CAT from being strictly exemplar-based to a model based on a mix of exemplars and prototypes. The Blending model exhibits a gradual learning trend and an overall classification accuracy near 50%. To increase the impact of observed features (context) on the class inferences made, we increased blending temperature in the High Temp model to broaden the range of generalization. This produced about a 10% gain in performance, along with improved overall performance of around 60% and a learning trend.

An examination of class confusions in the High Temp model indicated that overgeneralization might be hurting accuracy. To correct this, the Null Rewards model adds "guessing" productions to prevent rewards from being distributed in cases where the hypothesized class changes immediately before a class decision is made, a heuristic for uncertain classifications. The addition of guessing productions increases overall accuracy to about 80%.

The Specialized model replaces the decision production that terminates the encoding process with class-specific decision productions. This supports modeling the consequences of decisions, which may be important in some domains (e.g., identifying mines vs clutter as in [10]). Accordingly, the costs/benefits of different class decisions can be represented as rewards, which would lead to liberal/conservative class decisions. The Specialized model performs like the Null Rewards model, indicating that such a representation is viable.

Examining FS in the Specialized model, we see that the number of AUs encoded decreases with experience for almost all emotions (Figure 4). Thus CAT gradually gains efficiency while maintaining or improving classification accuracy. It is currently unclear why exemplars of surprise, the most prevalent class, are processed differently than the other classes. The heat maps in Figure 5 indicate some overlap in the features selected for each class by CAT (right panel) and those present in the Cohn-Kanade exemplars (left panel). Thus CAT appears to be reducing the dimensions of the classification problem (in most cases), while maintaining an overall accuracy near 80%.

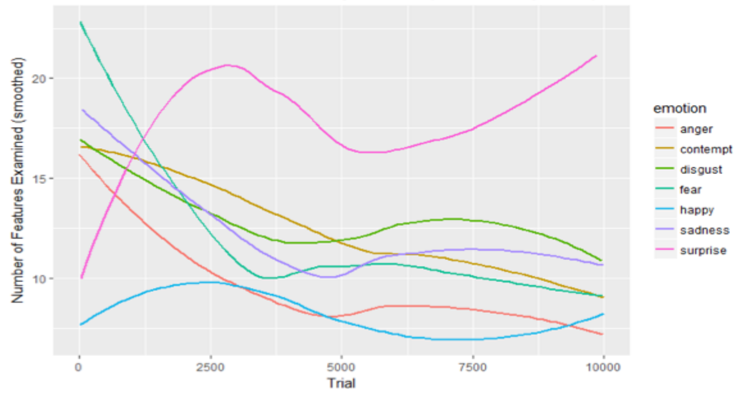

Fig. 4. Dimension reduction in the Specialized model

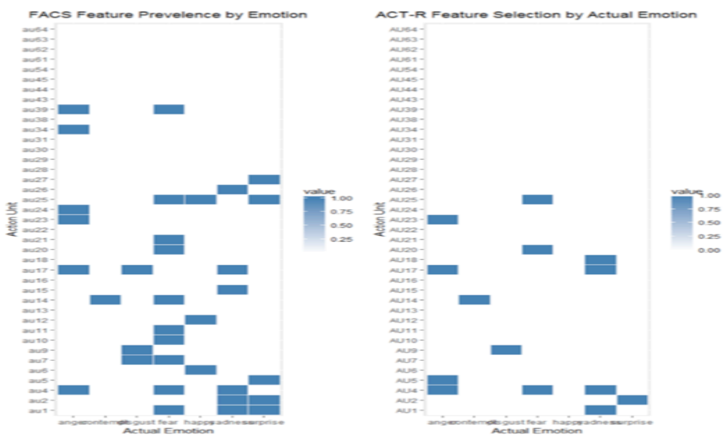

Fig. 5. Feature selection in the Specialized model

4 Conclusions and Future Directions

Cognitive robotics refers to the application of psychological frameworks, theories or models of information processing to autonomous systems. The basic idea behind cognitive robotics is that we might build better autonomous systems by using insights from the amassed body of knowledge about human cognition. In turn, we might learn more about human cognition by seeing how cognitive models fare in autonomous systems facing real-world constraints.

ACT-R models that "listen" to the architecture and avoid clever, unrealistic model engineering are more likely to generalize across domains. Our categorization model relies only on ACT-R's subsymbolic learning mechanisms and simple symbolic representations. We showed that the model learned to correctly classify new stimuli while learning to select task-relevant features. By using ACT-R's blending mechanism and adjusting the reward strategy, we improved overall classification accuracy from a baseline near 15% to around 80%.

In the future we plan to compare CAT to more traditional ML approaches including deep learning approaches that simultaneously learn feature selection and classification. We also plan to analyze the model's generality by applying it to datasets from different domains.

Finally, we want to establish a feedback loop between the perceptual and cognitive systems. So far in our work, information flows from the perceptual system to the cognitive system allowing us to learn to classify stimuli based on the features observed. Information, such as confidence measures, flowing from the cognitive system could support both active perception strategies that try to confirm the presence of important features and human robot interaction that elicit help from humans to establish and name new categories.

Acknowledgements. This research was supported by OSD ASD (R&E) and by the Army Research Laboratory's Robotics Collaborative Technology Alliance.

5 References

1. Fields, M., Lennon, C., Lebiere, C., & Martin, M. K. (2015, August). Recognizing Scenes by Simulating Implied Social Interaction Networks. In *International Conference on Intelligent Robotics and Applications* (pp. 360-371). Springer International Publishing.
2. Gonzalez, C., Lerch, F. J., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science* 27(4): 591-635.
3. Anderson, J. R., & Betz, J. (2001). A hybrid model of categorization. *Psychonomic Bulletin & Review*, *8*(4), 629-647.
4. Nosofsky, R. M., & Palmeri, T. J. (1997). An exemplar-based random walk model of speeded classification. *Psychological review*, *104*(2), 266.
5. Lamberts, K. (2000). Information-accumulation theory of speeded categorization. *Psychological review*, *107*(2), 227.
6. Kanade, T., Cohn, J. F., & Tian, Y. (2000). Comprehensive database for facial expression analysis. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on* (pp. 46-53). IEEE.
7. Ekman, P. (1999). Facial expressions. *Handbook of cognition and emotion*, *16*, 301-320.
8. Cohn, J. F., Ambadar, Z., & Ekman, P. (2007). Observer-based measurement of facial expression with the Facial Action Coding System. *The handbook of emotion elicitation and assessment*, 203-221.
9. Lebiere, C. (1999). The dynamics of cognitive arithmetic. *Kognitionswissenschaft* [Journal of the German Cognitive Science Society] Special issue on cognitive modelling and cognitive architectures, D. Wallach & H. A. Simon (eds.)., 8 (1), 5-19.
10. Lebiere, C. & Staszewski, J. (2010). Expert Decision Making in Landmine Detection. In *Proceedings of Human Factors and Ergonomics Society Conference. San Francisco, CA.*