A Computational Model of Event Segmentation and Learning

Hamit Başgöl (hamit.basgol@boun.edu.tr)
Cognitive Science, Boğaziçi University

İnci Ayhan (inci.ayhan@boun.edu.tr)
Department of Psychology, Boğaziçi University

Emre Uğur (emre.ugur@boun.edu.tr)
Department of Computer Engineering, Boğaziçi University

Keywords: event segmentation; event learning; computational model; point-light displays.

Introduction

We are subject to a continuous flow of information and we have to utilize it to show robust, adaptive, and intelligent behavior. For the utilization of continuous information, we discretize it into meaningful units. In the spatial realm, we segment scenes into objects; in the temporal realm, we segment a sequence of scenes into meaningful events (Zacks,2020).

Newtson (1973) observed that there is a similarity between where people segment ongoing activity into events. He asked participants to segment a movie by pressing a button to detect event boundaries in a procedure called unitization and realized that event boundaries detected by participants had a substantial agreement. Moreover, Speer, Swallow, and Zacks (2003) demonstrated that the decisions of participants did not change as time passed and showed stability. Also, they found that participants were capable of segmenting an activity into small (fine-grained) or large (coarse-grained) events. Event Segmentation Theory (EST) tries to explain the mechanism behind event segmentation (Zacks, Speer, Swallow, Braver, & Reynolds,2007). According to EST, event segmentation is a side effect of ongoing perceptual predictions, which are made by event models. The theory asserts that transiently increasing prediction error corresponds to event boundaries and triggers the use of another event model for the prediction of the next sensory input.

There are several computational models of event segmentation developed in cognitive science (Reynolds, Zacks, & Braver,2007;Zacks et al.,2007;Gumbsch, Kneissler, & Butz,2016;Gumbsch, Otte, & Butz,2017;Franklin, Norman, Ranganath, Zacks, & Gershman,2019), artificial intelligence and robotics (Nery & Ventura,2011;Butz, Bilkey, Humaidan, Knott, & Otte,2019). Based on the assumptions of the EST, we propose a computational model of event segmentation and learning. Our proposed model was inspired from the models developed by Gumbsch et al. (2016) and Gumbsch et al. (2017). The main incapability of these models is that they cannot learn and segment non-linear relationships in a sequence of information (Gumbsch et al.,2016,2017). Additionally, these models cannot learn and segment an event by passive observation because they assume the role of sensory-motor interaction for learning and segmenting events.

We introduced a model that can segment ongoing activity into meaningful parts and learn them through passive observation. We expect that the model can reliably segment human behaviors that are depicted as point-light displays (PLDs) and learn these behavior units.

Architecture

Before explaining the details of the model, we will explain the model we are inspired. Gumbsch et al. (2016,2017) developed a computational model of event segmentation. The computational model assumes that event segmentation is grounded on sensory-motor experiences and proposes a process that can autonomously divide a sequence of information into more than one event models that correspond to event segments. Event models involve a set of active linear forward models, each of which is responsible for continuously predicting the next sensory input by taking the current sensory input and the generated motor command. This complies with the assumptions of the predictive coding (Gumbsch et al.,2016,2017).

In the training phase, each forward model generates prediction, calculates prediction error and memorizes the moving average of prediction error and its variance. If the prediction error of a forward model is higher than an adaptive value (so-called surprise threshold), the system enters into a search period. In the search period, forward models coupled with a new forward model developed for a sensory dimension are trained within a number of timesteps and a new model is generated if all models are ineffective for predicting the next sensory input (Gumbsch et al.,2016,2017). Gumbsch et al. (2017) introduced a new parameter named confidence threshold to generate events with different granularities (fine- or coarse-grained). When the confidence threshold is high, the system starts determining coarser events because of the reduced frequency of entering the search period. Forward model transitions and the corresponding sensory inputs are associated with each other by multivariate Gaussian distribution (Gumbsch et al.,2016,2017).

Proposed model

Normally, combination of forward models, each of which is responsible from one type of sensory observation, constitute an event model. Gumbsch et al. (2016,2017), sensory modalities are handled independent of each other. In other words,

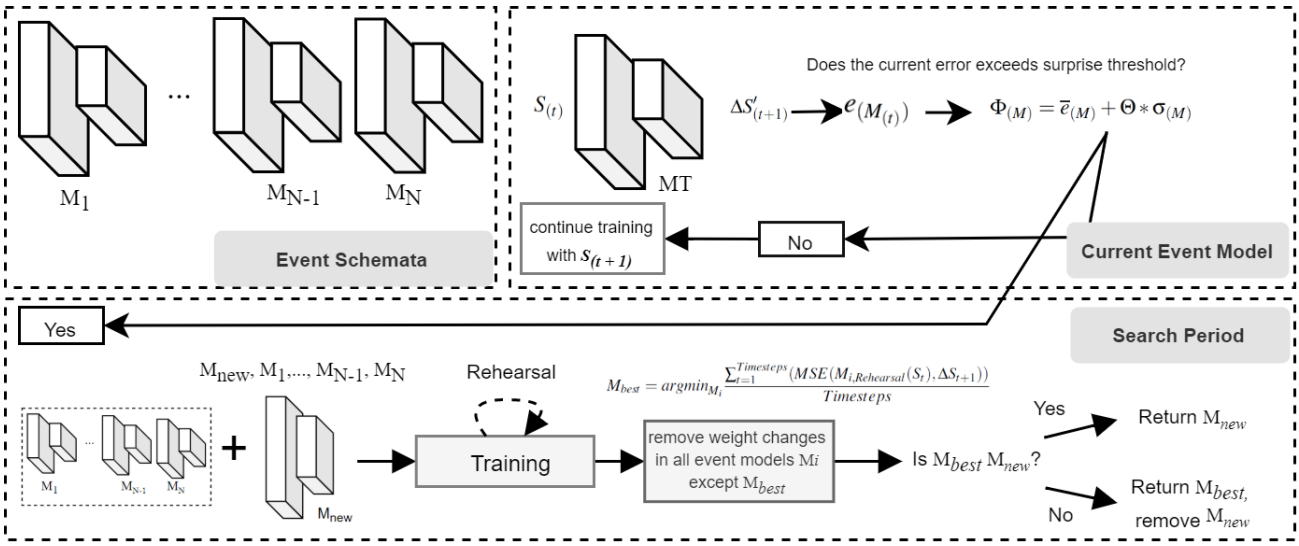

Figure 1: The algorithm used for training is given. A current event model makes prediction, if the error exceeds surprise threshold, the system goes to search period in which all models are trained for given timesteps and rehearsal rate. With the help of errors, the best event model is selected to be the current event model. If the new event model is the best event model, it becomes the current event model.

each forward model is responsible from a single sensor and its prediction. However, segmenting complex actions such as human motions require processing of combination of sensor modalities. Considering the complexity of event segmentation, we believe that linear models might not be able to make successful predictions. For this reason, we changed the type of an event model to be a multilayer perceptron that is able to approximate to nonlinear functions.

In detail, our proposed model has one active model in a time t, which will be mentioned as event model. While predicting a sequence of behavior in the environment, M_t which is responsible from predicting the change of every points observed. M_t receives current sensory observation vector S_t and tries to predict sensory change vector ΔS_{t+1}, which further gives the predicted sensory observation by

$$S'_{t+1} = S_t + \Delta S'_{t+1}$$

For each prediction, M_t learns and updates its weights. Then, M_t stores its prediction error.

Search period, event model training and switching

Each event model M_t stores its prediction error. From that error, a threshold rate, namely surprise threshold Φ, is calculated. $\Phi_{(M)}$ is calculated by the rolling mean of the prediction error $\overline{e}_{(M)}$ and of the variance $\sigma_{(M)}$. If the prediction error of an event model is greater than $\Phi_{(M)}$, the algorithm enters the search period. That is, $\Phi_{(M)}$ decides whether the current observation S_t is surprising. The event threshold Θ regulates the coarseness of the event to be segmented and therefore learned by an event model.

$$\Phi_{(M)} = \overline{e}_{(M)} + \Theta * \sigma_{(M)}$$

If the current error $e_{(M_t)}$ exceeds the surprise threshold $\Phi_{(M)}$, the model enters the search period, which shows the possibility that the current event model is not suitable for predicting the next sensory observation. In the search period, a new event model is generated. Then, the new event model and all event models that are in the event schemata are trained for a given timestep and rehearsal rate. Rehearsal rate corresponds to the number of iterations that event models trained. Errors generated by all models are accumulated during training. The model that has the least mean squared error in the last rehearsal is used for prediction. If the best model does not correspond to the new model generated at the beginning of the search period, the new model is removed. Weights of all models except the weights of the best model is restored. With the help of the search period, new event models are generated and all event models are trained if they are eligible to make prediction. The overview of the algorithm is given in Figure 1.

Method

Dataset

To assess the capability of the algorithm in segmenting and learning events, we prepared a training set involving human actions depicted by PLDs. Behaviors in the training set are taken from the KIT Motion-Language Dataset (Plappert, Mandery, & Asfour,2016). The training set involves a type of behavior defined as “a person picks up and object and place it on a surface in front of him” in the dataset. X and Y dimensions of 14 markers were used for depicting human movements as PLDs. Each behavior taken from the dataset was normalized. Normally, in the dataset, each behavior has 120 frame-per-second, which is reduced into 12 frame-per-

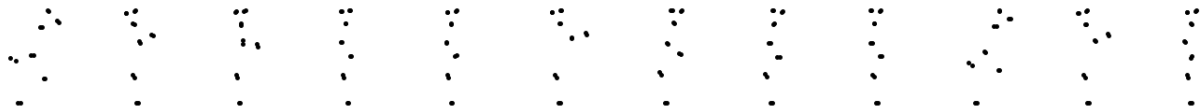

Figure 2: The behavior sequence used for training.

second, considering the computational efficiency. Trajectories are formed by attaching the trajectories of the same behavior three times. For the second attachment of the behavior, its trajectories are flipped (behavior + flipped version of the behavior + behavior).The behavior sequence prepared for training is shown in Figure 2.

Event models and hyperparameters

For depicting an event, we use PLDs, which are the way of non-pictorial depictions of human movements. PLDs developed by Johansson (1973), who invented the famous point-light walker, which is a depiction of human walking represented by several points. With the help of relative movement of points, people can perceive various kind of human movements, emotions, actions (Alaerts, Nackaerts, Meyns, Swinnen, & Wenderoth,2011), gender (Troje,2008). We believe that using the marker positions displayed by PLDs provide sufficient information for the underlying behavior and provides efficiency compared to the full-body RGB image.

Each event model has two hidden layers involving 64 and 32 neurons with RELU activation functions, respectively. Since each timestep is represented by 28 values that are X and Y positions of points and the input window is selected to be 5, the input size of an event model is determined to be 140. The output size of an event model is 28 and the activation function of the output layer is linear. The other hyperparameters regulating the behavior of the algorithm is given in Table 1.

Results

We evaluated the capability of our method with the same algorithm for the same sequence of information. The only difference in testing is that weights of event models are frozen and they are not trained. That is, no weight changes occur in the search period in the testing phase.

To check whether errors of event models in predicting the change between the current and the next sensory observation reduces, we calculated the cumulative error for an epoch by the mean of mean-squared error of all event models. Figure 3 shows the cumulative error the algorithm produces. It can be seen that errors of events models reduce in average, although they are not optimized cumulatively. Figure 4 shows target and predicted change in X and Y locations of one point representing the behavior sequence used in the training. It can be seen that event models can approximate to trajectories they are responsible from.

In addition to the learning events, our model can generate finer segments for a given behavior. The segmentation results of our method is given in Figure 5. It can be seen from the figure that the algorithm can detect patterns in a behavior. To explain better, we can symbolize the data such that the behavior taken from the dataset is coded as b1. The flipped version of b1 is coded as 1b. Then, we can represent the whole trajectory used in training as b1 + 1b + b1. It can be seen from the figure that b1 is composed of event models colored by green (0), cream (1), red (2) and blue (6) colors. 1b (the flipped version of b1) is composed of event models colored by purple (3), green (0), and cream (1) colors. That is, for segmenting a sequence of behavior, the same event model is used if needed and this occurs in a reliable manner. Colors, labels and possible definitions of the fine-grained events segmented by our method are given in Table 2.

Table 1: Hyperparameters of the computational model

Parameters	Values
Event threshold	2.0
Error window	10
Number of timesteps to predict (Timesteps)	10
Rehearsal	50
Input window	5
Number of pochs	50

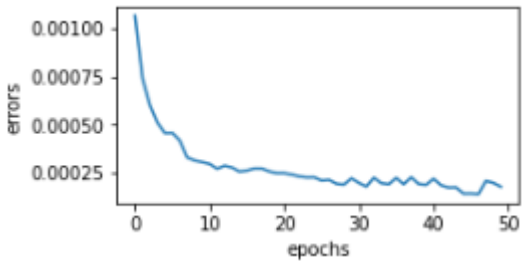

Figure 3: The figure shows that cumulative error decreases.

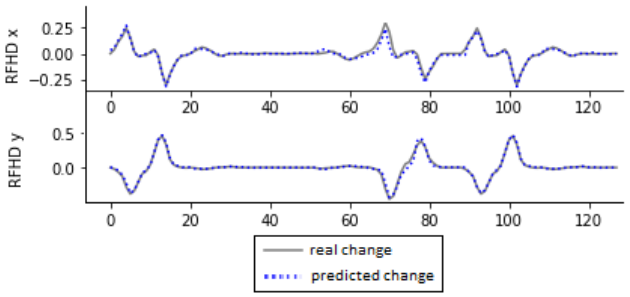

Figure 4: The prediction of event models for one point.

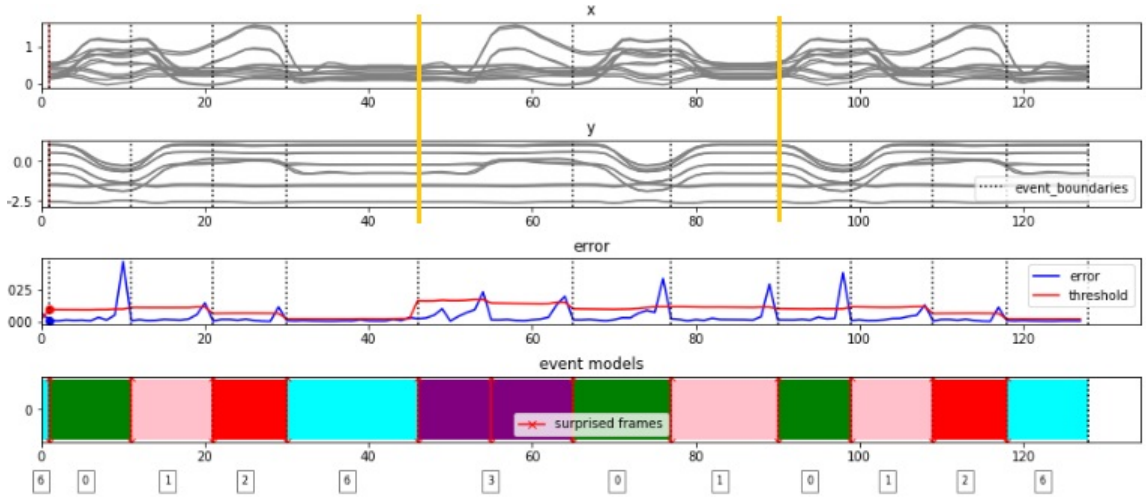

Figure 5: The segmentation results produced by our method. The yellow lines in the first and second figures show the merging points of the behavior. The color change in the fourth figure shows the point at which an event transition occurs. These points correspond to points at which prediction error shows a transient increase.

Table 2: Events segmented by the computational model

Event Labels	Colors	Definitions
0	Green	bending over to get something
1	Cream	straightening up with something
2	Red	putting something on the shelf
3	Purple	taking something from the table
6	Blue	waiting

Discussion and Conclusion

In this study, we introduced a computational model that is able to segment a sequence of human behavior into verbally definable and meaningful parts and generate events models that learn them. In this application, the length of considered event segments is short. In future, we aim to determine and learn longer event segments with the same method.

References

Alaerts, K., Nackaerts, E., Meyns, P., Swinnen, S. P., & Wenderoth, N. (2011). Action and emotion recognition from point light displays: an investigation of gender differences. *PloS one*, *6*(6), e20989.

Butz, M. V., Bilkey, D., Humaidan, D., Knott, A., & Otte, S. (2019). Learning, planning, and control in a monolithic neural event inference architecture. *Neural Networks*, *117*, 135–144.

Franklin, N., Norman, K. A., Ranganath, C., Zacks, J. M., & Gershman, S. J. (2019). Structured event memory: a neuro-symbolic model of event cognition. *BioRxiv*, 541607.

Gumbsch, C., Kneissler, J., & Butz, M. V. (2016). Learning behavior-grounded event segmentations. In *Cogsci.*

Gumbsch, C., Otte, S., & Butz, M. V. (2017). A computational model for the dynamical learning of event taxonomies. In *Cogsci.*

Johansson, G. (1973). Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, *14*(2), 201–211.

Nery, B., & Ventura, R. (2011). A dynamical systems approach to online event segmentation in cognitive robotics. *Paladyn, Journal of Behavioral Robotics*, *2*(1), 18–24.

Newtson, D. (1973). Attribution and the unit of perception of ongoing behavior. *Journal of Personality and Social Psychology*, *28*(1), 28.

Plappert, M., Mandery, C., & Asfour, T. (2016, dec). The KIT motion-language dataset. *Big Data*, *4*(4), 236–252.

Reynolds, J. R., Zacks, J. M., & Braver, T. S. (2007). A computational model of event segmentation from perceptual prediction. *Cognitive science*, *31*(4), 613–643.

Speer, N. K., Swallow, K. M., & Zacks, J. M. (2003). Activation of human motion processing areas during event perception. *Cognitive, Affective, & Behavioral Neuroscience*, *3*(4), 335–345.

Troje, N. F. (2008). Biological motion perception. *The senses: A comprehensive reference*, *2*, 231–238.

Zacks, J. M. (2020). Event perception and memory. *Annual Review of Psychology*, *71*, 165–191.

Zacks, J. M., Speer, N. K., Swallow, K. M., Braver, T. S., & Reynolds, J. R. (2007). Event perception: a mind-brain perspective. *Psychological bulletin*, *133*(2), 273.