

Gesture Analysis: Similarity Between Elicited and Enacted Gestures

Alejandro Andrade

10/6/2017

Summary

This brief paper summarizes my analysis of hand data extracted from the ESPD. The ESPD is an instructional system that cues the movement of the hands to create embodied representations of nonlinear dynamics of a predator-prey ecosystem (see Figure 1).

The goal of the analysis is to determine how closely the student is able to enact the elicited movement. In other words, the question is: Can we use some quantitative features of the student hand movements, as collected in the ESPD log data, to measure how close are these movements to the computer elicited movement?

Data are from two sources, the automated/elicited movement of predator and prey populations, and the student/enacted movement of both hands. The data is measured in pixels, and smaller values represent the hand being higher, and, conversely, larger values represent the hand being lower.

The analysis consists of four phases: (1) preprocessing, (2) state sequence modeling, (3) posterior estimation of state sequences, and (4) sequence similarity calculation. First, the hand position values are normalized and velocity values are computed. Velocity values are the difference between hand position at time t and $t - 1$. Second, a training phase takes place in which a Hidden Markov Model (HMM) is fit to the elicited data using the velocity values. Third, a prediction phase takes place where the HMM is used to estimate the sequence of states from the enacted movement. Finally, an Optimal Matching algorithm is used to compute the distance between the elicited and enacted sequences.

1. Preprocessing

The ESPD log captures the position of the hands and the markers (the automated horizontal bars used in cuing student movement of the hands). Table 1 shows the first 6 observations of the ESPD raw data.

1.1. Normalization

The raw data in the form of position values are first normalized dividing each value over the maximum position value and subtracting from 1 (to make large values up and small values down). Figure 2 plots the

Table 1: ESPD raw data

left_mark	right_mark	left_hand	right_hand
240	420	369	500
238	420	368	500
236	419	366	500
235	419	365	364
233	419	363	363
231	419	361	361

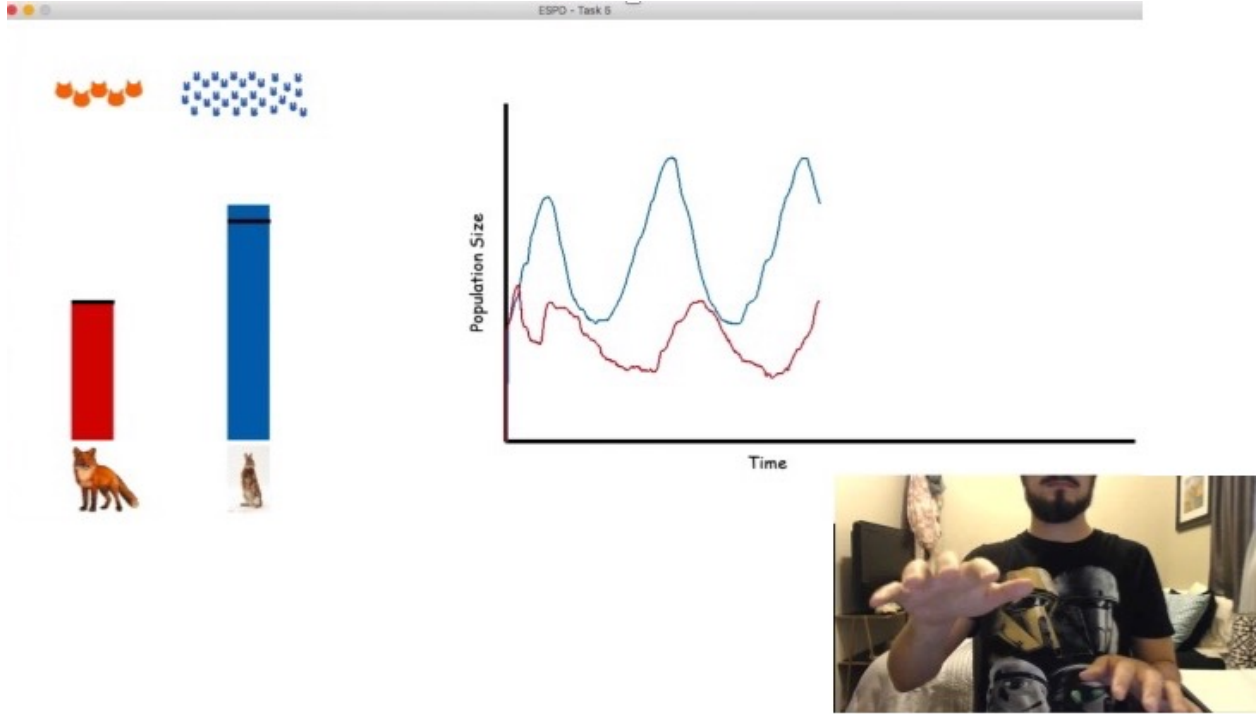


Figure 1: Snapshot of the Embodied Simulation of Population Dynamics.

movement of the automated markers and the movement of the hands using the normalized values for two students. It can be seen Student's 2 enacted movement was not as good as that of Student 1.

```
# normalize data
hand = dat %>% transmute(norm_left_hand = 1 - (left_hand / max(left_hand)),
                        norm_right_hand = 1 - (right_hand / max(right_hand)))
```

1.2. Velocity

The velocity of the movement is calculated by taking the difference between consecutive data points. Notice this operation results in real values (positive and negative), and thus the sign indicates the direction of the movement. Table 2 shows the results of this calculation for Student 1.

```
# compute velocity
hand_vel = with(hand, {
  data.frame(vel_left_hand = sapply(2:length(norm_left_hand),
    function(i) norm_left_hand[i] - norm_left_hand[i - 1]),
    vel_right_hand = sapply(2:length(norm_right_hand),
    function(i)
      norm_right_hand[i] - norm_right_hand[i - 1]))
})
```

2. State Sequence Modeling

To describe the characteristic pattern of the movement of the two hands, ten HMM models are fitted to the elicited data. The depmixS4 package is loaded and a loop is ran to iterate the fitting of the model over an increasing number of latent states: $k = 5, \dots, 15$. To select the best model, the Bayesian Information

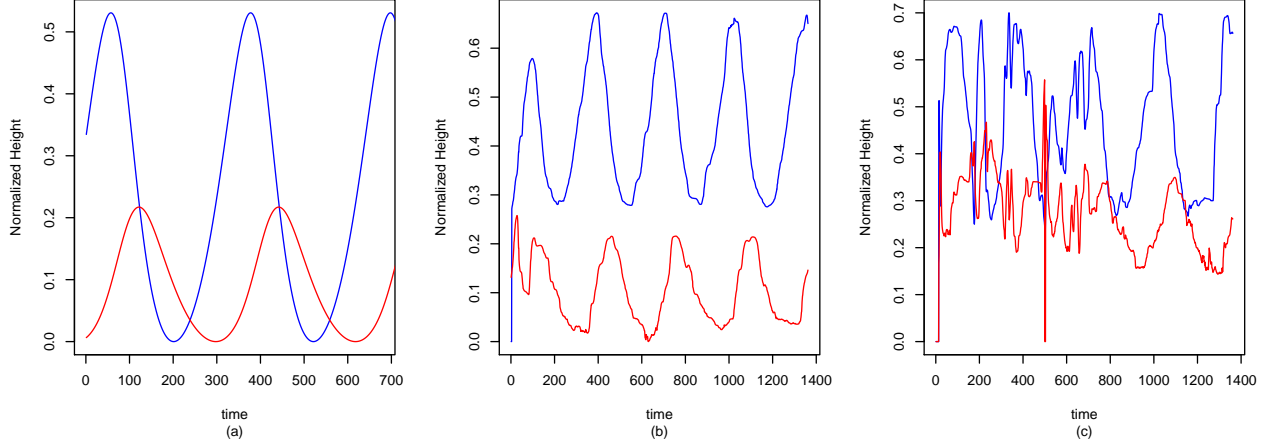


Figure 2: (a) Elicited movement, (b) enacted movement Student 1, and (c) enacted movement Student 2.

Table 2: Normalization and Velocity Calculation

left_hand	right_hand	norm_left_hand	norm_right_hand	vel_left_hand	vel_right_hand
368	500	0.136	0.000	0.004	0.000
366	500	0.139	0.000	0.004	0.000
365	364	0.143	0.272	0.003	0.272
363	363	0.147	0.275	0.004	0.003
361	361	0.151	0.278	0.004	0.003
361	360	0.152	0.279	0.001	0.001

Criterion (BIC) is used. The lower the BIC, the better the model fits to the data. Table 3 shows the model selection results.

According to Table 4, the best models is a 11-state model because it has the lowest BIC = -14884. Details of this model's mixture parameters are shown in Table 5. An interpretation of the latent states is given by the sign and magnitude of the mean velocity of the hands. If the left hand moves close to -0.002 speed and the right hand -0.003, the more likely it would be the gesture is of a State 1 type.

3. Sequence Similarity Calculation

The next step is to use the model parameters to predict the latent states. Using the hand velocity and the HMM model parameters, the posterior probability is calculated and a sequence of states is inferred for the enacted movement. This calculation uses a function of my own development, called *predictStates*, which calculates the most likely hidden states given a gaussian parameterization with m states and 2 observed continuous variables. Table 5 shows the first six entries for Student 1. Figure 3 shows the elicited and two enacted sequences.

```
# get HMM model parameters
m = k.best
emissionParams = getpars(model)[(m^2 + m + 1):length(getpars(model))] %>%
  matrix(ncol = 4, byrow = T)
mu = emissionParams[, c(1:3)]
sigma = emissionParams[, c(2:4)]
gamma = getpars(model)[(m + 1):(m^2 + m)] %>%
  matrix(ncol = m, byrow = T)
```

Table 3: HMM Model Selection

k	AIC	BIC
5	-14121.56	-13928.09
6	-14442.96	-14183.54
7	-14586.77	-14252.60
8	-14855.14	-14437.43
9	-15245.45	-14735.41
10	-15224.78	-14613.61
11	-15605.33	-14884.23
12	-15501.75	-14661.94
13	-15825.02	-14857.69
14	-15707.02	-14603.39
15	-15596.61	-14347.88

Table 4: Model Parameters

	mean_left_vel	sd_left_vel	mean_right_vel	sd_right_vel
St1	-0.002	0.000	-0.003	0.000
St2	0.001	0.001	-0.006	0.000
St3	0.001	0.000	0.005	0.000
St4	-0.002	0.000	-0.001	0.001
St5	-0.001	0.000	-0.005	0.000
St6	0.000	0.000	0.003	0.001
St7	0.002	0.000	0.004	0.000
St8	-0.002	0.000	0.000	0.000
St9	0.002	0.000	0.002	0.001
St10	-0.001	0.000	0.002	0.000
St11	0.003	0.000	-0.003	0.002

Table 5: Predicted States

vel_left_hand	vel_right_hand	State
0.0044	0.0000	9
0.0037	0.0000	9
0.0034	0.2715	9
0.0045	0.0034	11
0.0035	0.0027	2
0.0012	0.0014	5

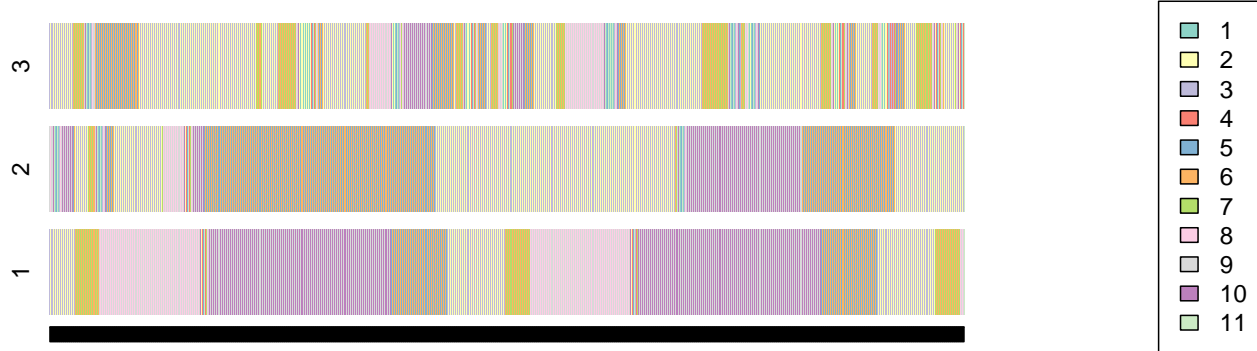


Figure 3: (1) Elicited, (2) Student 1, and (3) Student 2 enacted Sequences

```
# predict most likely sequence of states
elicited = predictStates(compu_vel, m, mu, sigma, gamma)
enactedS1 = predictStates(hand_vel, m, mu, sigma, gamma)
enactedS2 = predictStates(hand_vel2, m, mu, sigma, gamma)

data.frame(hand_vel[1:6, ],
           State = enactedS1[1:6]) %>%
  kable(caption = "Predicted States", digits = 4, format = "latex", align = "c")
```

4. Sequence Similarity Calculation

```
# compute indel and substitution costs from transition rates
costs = seqcost(seqdat[1, ], method = "TRATE")

## [>] creating substitution-cost matrix using transition rates ...
## [>] computing transition probabilities for states 1/2/3/4/5/6/7/8/9/10/11 ...

# compute similarity values
disVal = seqdist(seqdat, method = "OM", sm = costs$sm, indel = costs$indel, )

## [>] 3 sequences with 11 distinct states
## [>] checking 'sm' (one value for each state, triangle inequality)
## [>] 3 distinct sequences
## [>] min/max sequence length: 1361/1361
## [>] computing distances using the OM metric
```

Table 6: OM Distance Values

	Elicited	Student1	Student2
Elicited	0	1541	1782
Student1	1541	0	1264
Student2	1782	1264	0

```
##  [>] elapsed time: 0.618 secs
```

To compute the number of transformations required to convert the enacted sequence into the elicited sequence, an Optimal Matching algorithm is used. *Indel* costs are set to 1 and *Substitution* costs are set equal to the HMM transition rates. Larger values imply less similar sequences and, conversely, smaller values imply more similar sequences. The cost of transforming Student’s 1 enacted sequence into the elicited sequence is $d = 1541$ (see Table 6). The value of transforming Student’s 2 enacted sequence is $d = 1782$. It can be seen transforming this latter sequence is more costly than that of Student 1. This means the distance values provide information as to how well a student is able to coordinate their hands to enact the computer-elicited movement.

Conclusion

The above procedure can be regarded as a statistical measurement model of how well student enacted gestures are aligned with the computer elicited gestures. The information from this measurement model can help to make predictions about student learning gains. This predictive method can be implemented to support the personalization of the instructional system.