Introduction to n-grams and longest common subsequence in process data analysis

Qiwei He

Psychometric and Data Science Modeling Center

Educational Testing Service

July 19,2021

Overview

N-grams

- Item-based features extraction
- Disassemble long sequence into mini-sequences
- Applying term-weights
- Robust feature selection

Longest common subsequence (LCS) on process data

- Take sequence as a whole
- Calculate sequence distance
- Generate generable features across items

N-gram model on process data

(He & von Davier, 2015, 2016; Han, He, von Davier, 2019)

Objectives:

- To identify action patterns that are typically used by successful and unsuccessful groups.
- To identify differences in test-taking behaviors by countries.

Why n-grams?

- Disassemble long sequences into some pieces (easy compute).
- Extract information from observed response process.
- To identify action patterns that are typically used by successful and unsuccessful groups.
- To identify differences in test-taking behaviors by groups (countries).

N-grams in Language Model (LM)

- A language model is a probability distribution over entire sentences or texts
- N-gram is the minimum unit in LM (unigrams, bigrams, trigrams,…)
- In a simple *n-gram language model*, the probability of a word, conditioned on some number ***k*** of previous words.
- In other words, using the previous **n-1** words in a sequence we want to predict the next word.

N-grams in Language Model (LM)

Sue swallowed the large green ____.

- **A. Frog**
- **B. Mountain**
- **C. Car**
- **D. Pill**

Unigram: 1-1=0, use the word itself.

Bigram: 2-1=1, use one previous word "green" to predict the missing word.

Trigram: 3-1=2, use two previous words "large green" to predict the missing word.

Statistical view

- **Markov assumption:** The probably of the next word depends only on the previous *k* words (*k=n-1*). This gives a kth order Markov approximation:

$$P(w_n|w_1 \dots w_{n-1}) = P(w_n|w_{n-k}, w_{n-k+1} \dots w_{n-1})$$

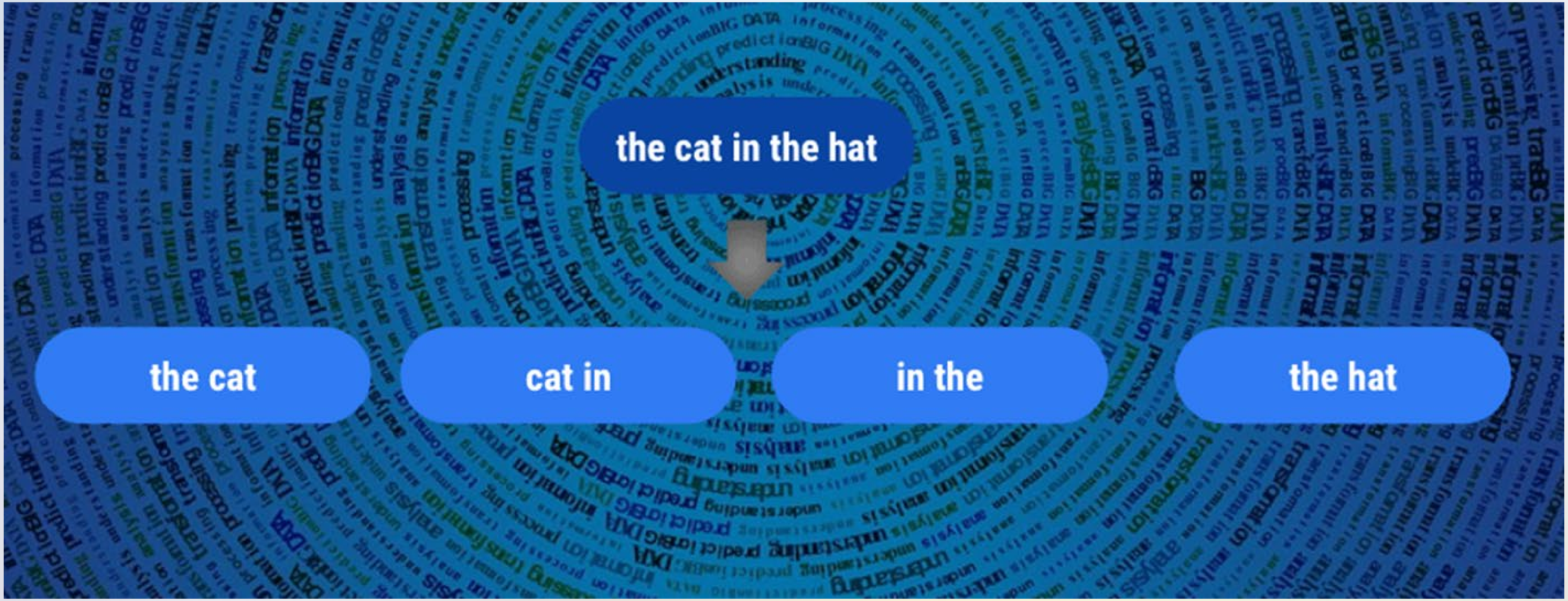
Common N-grams: $\mathbf{w} = (w_1, w_2, \dots, w_n)$

Unigram: $P(\mathbf{w}) = P(w_1)\, P(w_2) \dots P(w_n)$

Bigram: $P(\mathbf{w}) = P(w_1)\, P(w_2|w_1) \dots P(w_n|w_{n-1})$

Trigram: $P(\mathbf{w}) = P(w_1)\, P(w_2|w_1) P(w_3|w_2, w_1) \dots P(w_n|w_{n-2}, w_{n-1})$

A typical bigram representation



N-grams in action sequence

- N-gram methods decode a long sequence of actions into small pieces.
- Unigrams are defined as “bags of actions,” where each single action in a sequence collection represents a distinct feature.
- Bigrams, trigrams and higher-order grams are action sequences broken down into mini-sequences containing two and three or even higher number of ordered adjacent actions.

N-grams in action sequence

I am happy to give a talk today.

unigrams | bigrams | trigrams

Action sequence: STRT, SS, SS_Type_FN, E, E_S, Next, Next_OK, END

Unigrams (8)	"START", "SS", "SS_Type_FN", "E", "E_S", "Next", "Next_OK", "END"
Bigrams (7)	"START, SS", "SS, SS_Type_FN", "SS_Type_FN, E", "E, E_S", "E_S, Next", "Next, Next_OK", "Next_OK, END"
Trigram (6)	"START, SS, SS_Type_FN", "SS, SS_Type_FN, E", "SS_Type_FN, E, E_S", "E, E_S, Next", "E_S, Next, Next_OK", "Next, Next_OK, END"

Term weights

- In information retrieval, raw term frequency (from a corpus) usually suffers from a critical problem: All terms are considered equally important when assessing relevancy on a query. In fact, certain terms have little or no discriminating power in determining relevance.

  - Grams that every sentence has (e.g., stop words)
  - Grams occur multiple times in one sentence should be the same weight as the grams occur single time in one sentence but by multiple people?

Term Weights (tf.isf)

- An **inverse sequence frequency** was applied for attenuating the effect of actions that occurred too often in the collection to be meaningful.
- A **dampened term frequency** was also used to adjust the importance of an action with multiple occurrences in a single sequence.

Dampened term frequency

Inverse sequence frequency

$$\text{weight}(i, j) = \begin{cases} \underbrace{\boxed{[1 + \log(\text{tf}_{i,j})]}}_{\text{Dampened term frequency}} \underbrace{\boxed{\log(N / \text{sf}_i)}}_{\text{Inverse sequence frequency}} & \text{if } \text{tf}_{ij} \geq 1 \\ 0 & \text{if } \text{tf}_{ij} = 0 \end{cases}$$

i, j action i in sequence j

$\text{tf}_{i,j}$ frequency of action i in sequence j

sf_i frequency of sequence that contains action i

N number of sequences (test takers)

An example of term weights

$$a_1 = \{Start, ViewF, ViewM, Mdrag, Mdrop, Moved, ViewF, ViewM, Next, NextC, ViewF, ViewM, Mdrop, Moved, Next, NextOK\}$$

$$b_1 = \{ViewF, ViewM\}$$

$$tf_{1,1} = 3$$

$$Assume\ N_s = 500\ of\ at\ least\ length\ 2,$$

$$sf_1 = 300\ contain\ b_1\ at\ least\ once$$

$$weight\ (1,1) = [1 + \log(3)] \log\left(\frac{500}{300}\right) = 1.07$$

$$sf_1 = 100\ contain\ b_1\ at\ least\ once$$

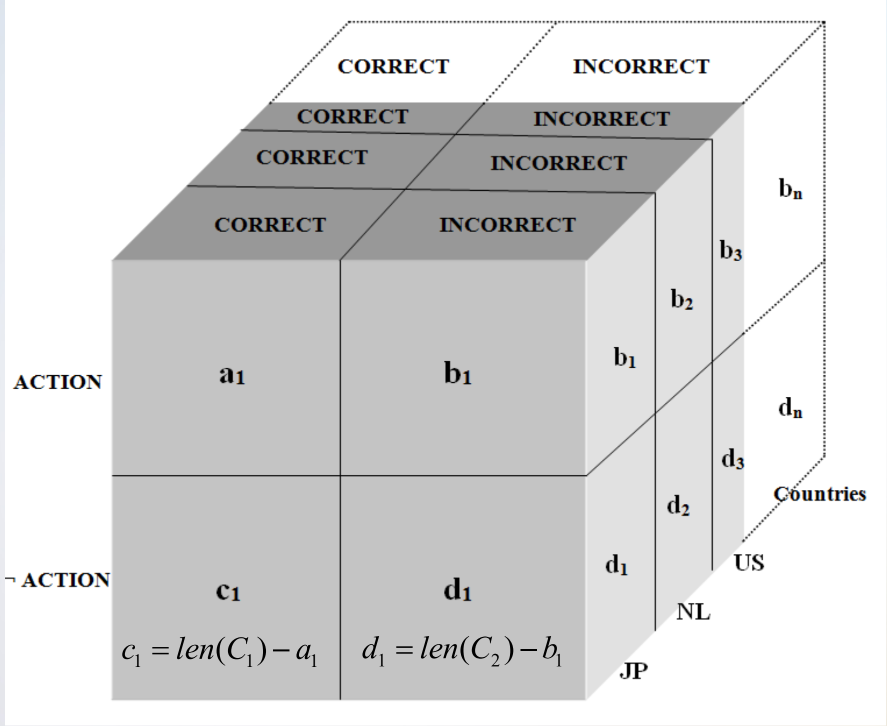
$$weight\ (1,1) = [1 + \log(3)] \log\left(\frac{500}{100}\right) = 3.38$$

Some general rules

- Actions that occur fewer than five times that occur in the whole collection **(ActFreq<5)** are usually removed from further analysis because of consideration on reliability.

- **Actions that are used by all the test takers** are usually deducted from the further analysis because of little information for prediction or differentiation by subgroups.

Chi-square Feature Selection Model



$$\chi^2 = \frac{M(ad - bc)^2}{(a+b)(a+c)(b+d)(c+d)}$$

$$c = len(C_1) - a$$

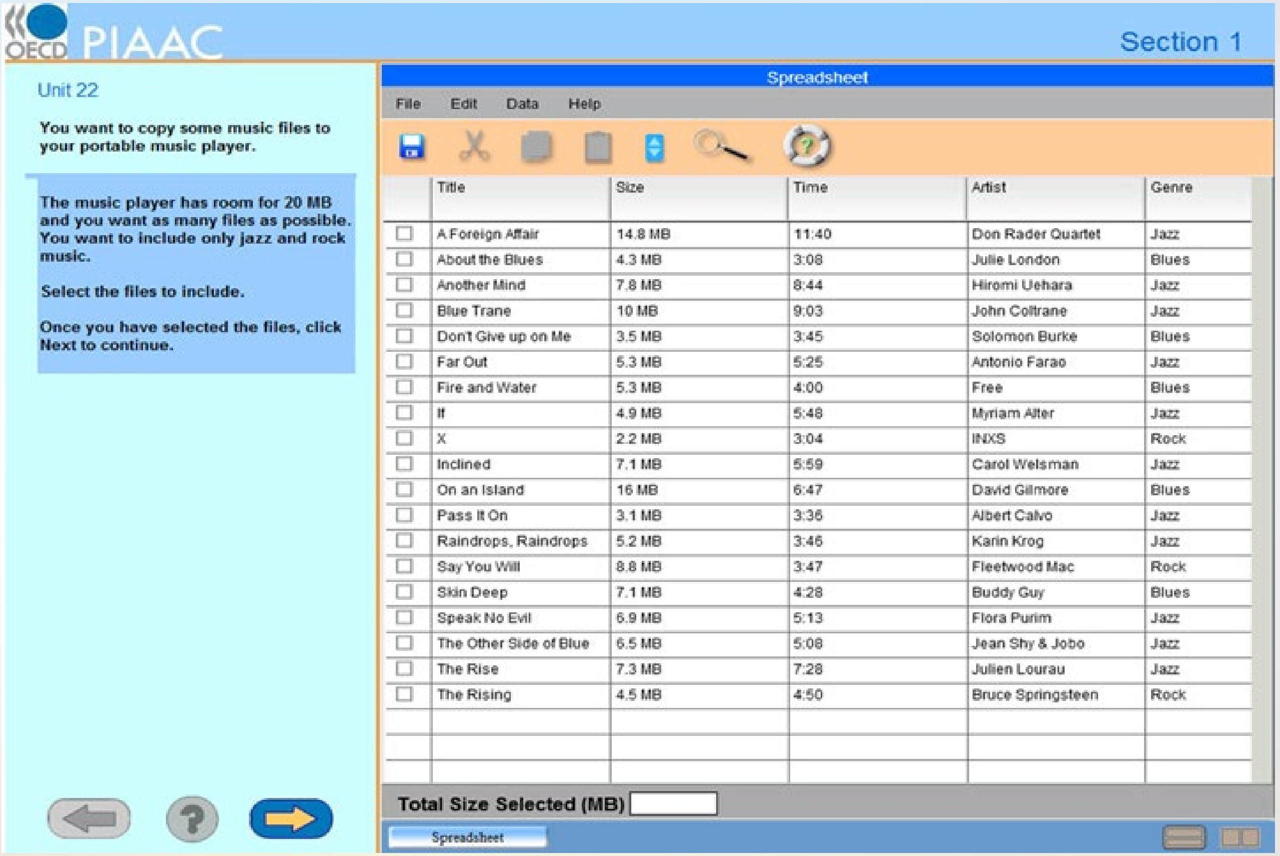
$$d = len(C_2) - b$$

$$M = a + b + c + d$$

The actions with **higher chi-square scores** are **more discriminative** in classification. Therefore, we ranked the chi-square score of each action in a **descending order**. The actions ranked to the top were defined as the robust classifiers.

An Example PSTRE Item

- The task is to identify the ID number of a specified person and send this number to a correspondent by email.
- Two environments are involved:
  - A spreadsheet environment that contains a database as the stimulus material that displays the information required to solve task.
  - An email environment to provide the response.
- The interim score is evaluated based only on the email responses.



He, Q., & von Davier, M. (2016). Analyzing Process Data from Problem-Solving Items with N-Grams: Insights from a Computer-Based Large-Scale Assessment. In Y. Rosen, S. Ferrara, & M. Mosharraf (Eds.) *Handbook of Research on Technology Tools for Real-World Skill Development* (pp. 749-776). Hershey, PA: Information Science Reference.

Sample description

Characteristics	Total	US	NL	JP
N	**3926**	1340	1508	1078
Correct (%)	**2754 (70.1)**	882(65.8)	1104 (73.2)	768 (71.2)
Incorrect (%)	**1172 (29.9)**	458 (34.2)	404 (26.8)	310 (28.8)
Gender				
Female	2025	629	711	526
Male	1901	711	629	552
Age (years)				
Mean (S.D.)	39.60 (14.01)	39.21 (14.00)	40.84 (14.29)	38.35 (13.49)
Educational level				
Less than high school	615	124	401	90
High school	1493	534	590	369
Above high school	1812	680	513	619
Missing	6	2	4	0

Note. US, NL and JP represent the sample from the United States, the Netherlands and Japan.

Results: Features of Actions by Performance Groups

Robust Features of Actions and Action Sequences Distinguishing Correct and

	Unigrams		Bigrams			
	Actions	χ^2	Actions	χ^2	[illegible]	[illegible]
Correct	SS	70.72	E, SS	229.99	E, S[illegible]	272.49
	SS_Type_SN	68.04	SS, E	191.18	[illegible]ART, E, SS	226.42
	SS_So_OK	64.58	SS_So_OK, E	153.90	SS, E, E_S	211.37
	SS_So_1B	59.66	SS_So_1B, SS_So_OK	122.49	SS_So_OK, E, SS	150.25
	[illegible]	[illegible]	[illegible]Type_SN, E	120.56	SS_So_1B, SS_So_OK, E	137.53
	[illegible]	[illegible]	[illegible]Se, SS_Type_SN	98.21	SS, E, SS	133.85
	[illegible]	[illegible]	[illegible]So, SS_So_1B	84.43	SS_Se, SS_Type_SN, E	108.55
	SS_So_2A	[illegible]	START, SS_Se	70.03	SS_Type_SN, E, SS	108.20
Incorrect	Next_C	892.8[illegible]	[illegible]ART, Next	2416.20	START, Next, FINALENDING	2420.26
	SS_Save	98.90	Next, Next_C	521.74	Next, Next_[illegible]	478.16
	SS_Type_PGN	33.19	Next_C, Next	504.22	START, E, Ne[illegible]	399.02
	SS_H	15.75	E_S, E_S	492.26	Next_[illegible]	[illegible]
	SS_So_3D	14.56	E_S, E	364.66	E_S, [illegible]	[illegible]
	SS_So_C	[illegible]	[illegible]S, SS	299.74	E, E_[illegible]	[illegible]
	E_S	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	SS_Type_P[illegible]	[illegible]	[illegible]	[illegible]	[illegible]S, E	338.26

Correct group: using tools such as searching engine and sorting with a clear sub-goal

Incorrect group: hesitative behaviors using "cancel" a lot

Nonresponse pattern: START, Next, FINALENDING (NONRESPONSE)

Incorrect group: using "Help" function a lot and aimless save the results in the server

Results: Country Level vs. Aggregate Level

Consistency Rate of Extracted Classifiers by Performance Groups Compared Between Country Level and Aggregate Level

	US	Netherlands	Japan
Correct			
Unigrams	0.88	0.88	0.63
Bigrams	0.75	0.88	0.75
Trigrams	0.75	0.88	0.75
Incorrect			
Unigrams	0.63	0.63	0.63
Bigrams	0.63	0.88	0.88
Trigrams	0.75	0.63	0.75

Mean=0.79

Mean=0.71

Results: Features of Actions by Countries

Robust Features of Actions and Action Sequences Across Countries

	Unigrams		Bigrams		[illegible]	
	Actions	χ^2	Actions	χ^2	[illegible]	[illegible]
US	Next_C	20.40	E, E	261.08	E, E, E	309.01
	SS_Type_FN	15.64	START, Next	39.82	E, E, Next	278.87
	E	13.25	Next, E	39.28	SS, E, E	132.21
	SS_Type_PGN	10.14	START, E	38.97	START, E, E	85.14
	SS_Save	6.22	SS_So_C, SS_Type_FN	37.63	SS_Type_FN, E, E	54.23
NL	SS_Type_FN	315.30	SS_Se, SS_Type_FN	252.93	START, SS_Se, SS_Type_GN	226.67
	SS_Type_GN	232.93	SS_Type_FN, SS_Type_FN	249.97	[illegible]	[illegible]
	SS_Se	60.88	SS_Type_FN, E	203.30	[illegible]	[illegible]
	SS_So_3B	31.59	SS_Se, SS_Type_GN	202.10	[illegible]	[illegible]
	SS_So_2A	16.15	START, SS_Se	117.42	SS_Se, SS_Type_FN, SS_Type_FN	[illegible]
JP	SS_Type_SM	383.58	SS_Type_SM, SS_Type_SM	308.58	SS_Type_SM, SS_Type_SM, SS_Type_SM	248.84
	SS_Type_null	123.49	[illegible], SS_So	166.12	E_S, Next, Next_C	149.25
	SS_Type_UM	70.75	[illegible]	137.22	SS_Type_SM, SS_So, SS_So_1B	149.21
	[illegible]	[illegible]	[illegible]	116.73	SS_Type_SM, SS_Type_SM, SS_So	140.96
	[illegible]	[illegible]	[illegible]	115.33	SS_Type_SM, SS_Type_SM, E	116.15

US: Double clicks on E-mail page

NL: More likely use full name and given names when doing searching

JP: Spelling mistakes (optimal space between first name and last name)

JP: strategy changed

Demo example of n-grams in Program R

```
library (ngram)
```

```
##read in  demo_data
DATA <- read.csv(file="data_demo.csv",header = TRUE,sep=",")
```

```
##====================================================
##n-grams model for one sequence example
##====================================================
num <- 1 ##set a number between 1-600
obs <- as.character(DATA$Coded_Action_Sequences[num])
obs <- gsub(',','',obs)  ##remove the comma between each action

##ngrams
ng1<-ngram(obs,n=1) ##unigram
ng1

ng2<-ngram(obs,n=2) ##bigram
ng2

ng3<-ngram(obs,n=3) ##trigram
ng3

##summarize n-grams
get.phrasetable(ng1)
get.phrasetable(ng2)
get.phrasetable(ng3)
```

Demo example of n-grams in Program R

Analysis on unigrams and bigrams in the dataset

```
##uniqrams analysis
ng1<-lapply(ActionSequences[which(l2 == TRUE)], function(x) ngram(x,n=1)) # extract all unigrams (from non-skipping respondents)
ngs1<-unlist(lapply(ng1, function(x) get.ngrams(x))) ##all unigram token (with repetition)
uni_type <- unique(ngs1) ## all unigram type (without repetition)
length_uni_type <-length(unique(ngs1)) ## how many unique uniqrams used by the sample
tail(sort(table(ngs1)))
Nng1<-length(ng1) # number of effective sequences (exclude skipping) equals to number of respondents who should be taken into account


##bigrams analysis
ng2<-lapply(ActionSequences[which(l2 == TRUE)], function(x) ngram(x,n=2)) # extract all bigrams (from non-skipping respondents)
ngs2<-unlist(lapply(ng2, function(x) get.ngrams(x))) ##all bigrams token (with repetition)
bi_type <- unique (ngs2) ## all bigram type (without repetion)
length_bi_type <- length(unique(ngs2)) ## how many unique bigrams used by the sample
tail(sort(table(ngs2)))
Nng2<-length(ng2) # number of effective sequences (exclude skipping)
```

Demo example of n-grams in Program R

Robust n-grams selection with Chi-square score

```
# Results table: frequencies and weighted frequencies
wgtng2<-setNames(data.frame(matrix(ncol =4, nrow =Nng2*length(unique(ngs2)))), c("ngram","freq","weightfreq", "score"))
wgtng2[,1]<-rep(unique(ngs2),Nng2)
for(i in 1:length(unique(ngs2))){
  ISF<-log(Nng2/sum(unlist(lapply(ng2, function(x) unique(ngs2)[i]%in%get.ngrams(x))))) # inverse sequence frequency of bigram
  print(ISF)
  for(n in 1:Nng2){
    nG<-get.ngrams(ng2[[n]]) # bigrams for person
    if(unique(ngs2)[i] %in% nG){
      freq<-sum(unique(ngs2)[i]==nG) # frequnecy in person-level sequence
      weightfreq<-freq*(1+log(freq))*ISF # weighted frequency person-level seqeence
      wgtng2[(n-1)*length(unique(ngs2))+i,2:4]<-c(freq,weightfreq,score[n]) # fill table
    }else{
      wgtng2[(n-1)*length(unique(ngs2))+i,2:4]<-c(0,0,score[n]) # set frequencies to zero if bigram did not occur in person-specific sequence
    }
  }
}
```

Demo example of n-grams in Program R

Apply term weights tf.isf

```
# Results table: frequencies and weighted frequencies
wgtng2<-setNames(data.frame(matrix(ncol =4, nrow =Nng2*length(unique(ngs2)))), c("ngram","freq","weightfreq", "score"))
wgtng2[,1]<-rep(unique(ngs2),Nng2)
for(i in 1:length(unique(ngs2))){
  ISF<-log(Nng2/sum(unlist(lapply(ng2, function(x) unique(ngs2)[i]%in%get.ngrams(x))))) # inverse sequence frequency of bigram
  print(ISF)
  for(n in 1:Nng2){
    nG<-get.ngrams(ng2[[n]]) # bigrams for person
    if(unique(ngs2)[i] %in% nG){
      freq<-sum(unique(ngs2)[i]==nG) # frequncy in person-level sequence
      weightfreq<-freq*(1+log(freq))*ISF # weighted frequency person-level seqeence
      wgtng2[(n-1)*length(unique(ngs2))+i,2:4]<-c(freq,weightfreq,score[n]) # fill table
    }else{
      wgtng2[(n-1)*length(unique(ngs2))+i,2:4]<-c(0,0,score[n]) # set frequencies to zero if bigram did not occur in person-specific sequence
    }
  }
}
```

Summary

- N-grams function well at item-level to provide fine-grained action analysis.
- N-grams approach could quickly provide an initial result on the most informative actions (mini-sequences) by each group, thus could apply in item quality checking, especially after getting process data for field trial. It can help quickly spot the potential issues in the item design.
- It is recommended to use n<=3 in process data analysis. With the n goes higher, the frequency of each gram may drop down. The low frequency may also not be reliable in the analysis.

- Although many similarities are shared between sequential data structure of language and action sequences. There are still many differences.
    - In language model, the bag-of-words (unigrams) are usually found the most informative in prediction. While in process data, mini-sequences (esp. bigrams and trigrams) are often recommended to take more concerns on dependence of actions that are in high possibility of joint occurrence.
    - Timing information could be additional source to strengthen the function of n-gams features in discriminating groups (e.g., time interval between actions, which has similarity with the speech recognition but not necessarily used in the text mining on words.

Sequence similarity and efficiency with longest common subsequence

(He, Borgonovi, & Paccagnella, 2019, 2021)

Objectives:

- To compute the sequence distance between individual observed sequence with predefined reference sequence.
- To generalize process data variables across interactive problem-solving items.

Why Longest Common Subsequence?

- The Longest Common Subsequence (LCS) method (Maier, 1978; Hirschberg,1975; Chvatal & Sankoff, 1975), a sequence-mining technique used in natural language processing and biostatistics to grasp test-takers' strategy when solving digital tasks.

- The longest common subsequence was first introduced into educational assessment by Sukkarieh, Yamamoto, & von Daiver (2012) as a tool for automated scoring in multiple linguistic environment.

- The unique application of LCS in process data is to identify the action sequences that are most similar to the predefined, "optimal" sequences for each item. That is, we calculate the distance between each individual against the predefined optimal sequence(s).

- Measurement indicators are developed in order to analyze behaviors across items and subgroups of respondents.

- This approach extends the research capacity from understanding individuals' problem-solving behaviors in a single item to a general perspective across multiple items that form an assessment.

- This approach could also be applied well to check the item design, i.e., whether test-takers' problem solving strategy match with item developers' expectation.

He, Q., Borgonovi, F., Paccagnella, M. (2021). Leveraging process data to assess adults' problem-solving skills: Identifying generalized behavioral patterns with sequence mining. *Computers and Education, 166*, 104170. https://doi.org/10.1016/j.compedu.2021.104170

Compute LCS

Let $X = (x_1, x_2, \ldots, x_i)$ and $Y = (y_1, y_2, \ldots, y_j)$ be two sequences. x_i and y_j are actions within the sequence X and Y, respectively. Assume Y is the predefined sequence. The prefixes of X and Y are $X_{1,} X_{2,}, \ldots, X_i$ and $Y_{1,} Y_{2,}, \ldots, Y_j$, respectively. Let $LCS\left(X_i, Y_j\right)$ represent the set of longest common subsequence of prefixes X_i and Y_j. The set of sequences is given as:

$$LCS\left(X_i, Y_j\right) = \begin{cases} \emptyset & if\ i = 0\ or\ j = 0 \\ LCS\left(X_{i-1}, Y_{j-1}\right),\ x_i & if\, x_i = y_i \\ \text{longest}\ \left(LCS\left(X_i, Y_{j-1}\right), LCS\left(X_{i-1}, Y_j\right)\right) & if\ x_i \neq y_i \end{cases}$$

$$length(LCS\left(X_i, Y_j\right)) = \begin{cases} 0 & if\ i = 0\ or\ j = 0 \\ \text{length}(i-1, j-1) + 1 & if\, x_i = y_i \\ \max\left(length(i, j-1), length(i-1, j)\right) & if\ x_i \neq y_i \end{cases}$$

$$LCS(X, \mathbf{Y}) = \text{longest}\ \left(LCS\left(X_i, Y_{kj}\right)\right)$$

For multiple predefined optimal sequences

Longest Common Subsequences (LCS)

		0	1	2	3	4	5	6	7
		Ø	M	Z	J	A	W	X	U
0	Ø	0	0	0	0	0	0	0	0
1	X	0	0	0	0	0	0	1	1
2	M	0	1	1	1	1	1	1	1
3	J	0	1	1	2	2	2	2	2
4	Y	0	1	1	2	2	2	2	2
5	A	0	1	1	2	3	3	3	3
6	U	0	1	1	2	3	3	3	4
7	Z	0	1	2	2	3	3	3	4

Obs: M Z J A W X U
Ref: X M J Y A U Z
LCS: M J A U

LCS Computation Example

RS_1: searching from toolbar (length=11)
Start, Toolbar_SS_Find, On_SearchBox, Off_SearchBox, Search_OK, SS_SEARCH, Email, On_Email_Message, Off_Email_Message, Next, Next_OK

RS_2: searching from menu item (length=11)
Start, Menuitem_Find, On_SearchBox, Off_SearchBox, Search_OK, SS_SEARCH, Email, On_Email_Message, Off_Email_Message, Next, Next_OK

RS_3: sorting from toolbar (length=9)
Start, Toolbar_SS_Sort, Sort_1_B, Sort_OK, Email, On_Email_Message, Off_Email_Message, Next, Next_OK

RS_4: sorting from menu item (length=9)
Start, Menuitem_Sort, Sort_1_B, Sort_OK, Email, On_Email_Message, Off_Email_Message, Next, Next_OK

OBSERVATION (length=25)
Start,Toolbar_SS_Help,Menu_SS_Edit,Menu_SS_Data,Menuitem_Sort,Sort_1_B,Sort_1A,Sort_OK,SS_Sort_1Ba,Email,On_Email_Message,Off_Email_Message,SS,On_Email_Message,Off_Email_Message,Email,On_Email_Message,,,,,,,Off_Email_Message,Toolbar_E_Send,On_Email_Message,Off_Email_Message,Next,On_Email_Message,Off_Email_Message,Next_OK

LCS1 (length=6): Start, Email, On_Email_Message, Off_Email_Message, Next, Next_OK
LCS2 (length=6): Start, Email, On_Email_Message, Off_Email_Message, Next, Next_OK
LCS3 (length=8): Start, Sort_1_B, Sort_OK, Email, On_Email_Message, Off_Email_Message, Next, Next_OK
LCS4 (length=9): Start, Menuitem_Sort, Sort_1_B, Sort_OK, Email, On_Email_Message, Off_Email_Message, Next, Next_OK

LCS indicators

- ***Similarity = Length(LCS)/length (ref_seq)*** range=[0,1]
  - 1 is the highest similarity, completely match with the predefined action sequence;
  - 0 is the lowest similarity, nothing overlaps with the predefined action sequence.

- ***Efficiency = Length(LCS)/length (obs_seq)*** range=[0,1]
  - 1 is the highest efficiency, all actions are related actions (no redundant actions)
  - 0 is the lowest efficiency, all actions are unrelated actions

LCS Indicators

- Similarity
  - $Similarity = \text{len}(LCS)/\text{len}(RS)$
  - $SM = \text{Mean}(Sim_1, Sim_2, \ldots, Sim_n)$
  - $SSD = \text{SD}(Sim_1, Sim_2, \ldots, Sim_n)$
- Efficiency
  - $Efficiency = \text{len}(LCS)/\text{len}(OS)$
  - $EM = Mean(Eff_1, Eff_2, \ldots, Eff_n)$
  - $ESD = \text{SD}(Eff_1, Eff_2, \ldots, Eff_n)$

		Average Similarity (MEAN)		
		M1 (Low Similarity)	M2 (Moderate Similarity)	M3 (High Similarity)
Consistency (SD)	SD1 (High Consistency)	S11 High Consistency Low Similarity	S12 High Consistency Moderate Similarity	S13 High Consistency High Similarity
	SD2 (Moderate Consistency)	S21 Moderate Consistency Low Similarity	S22 Moderate Consistency Moderate Similarity	S23 Moderate Consistency High Similarity
	SD3 (Low Consistency)	S31 Low Consistency Low Similarity	S32 Low Consistency Moderate Similarity	S33 Low Consistency High Similarity

Example Data and Instrument

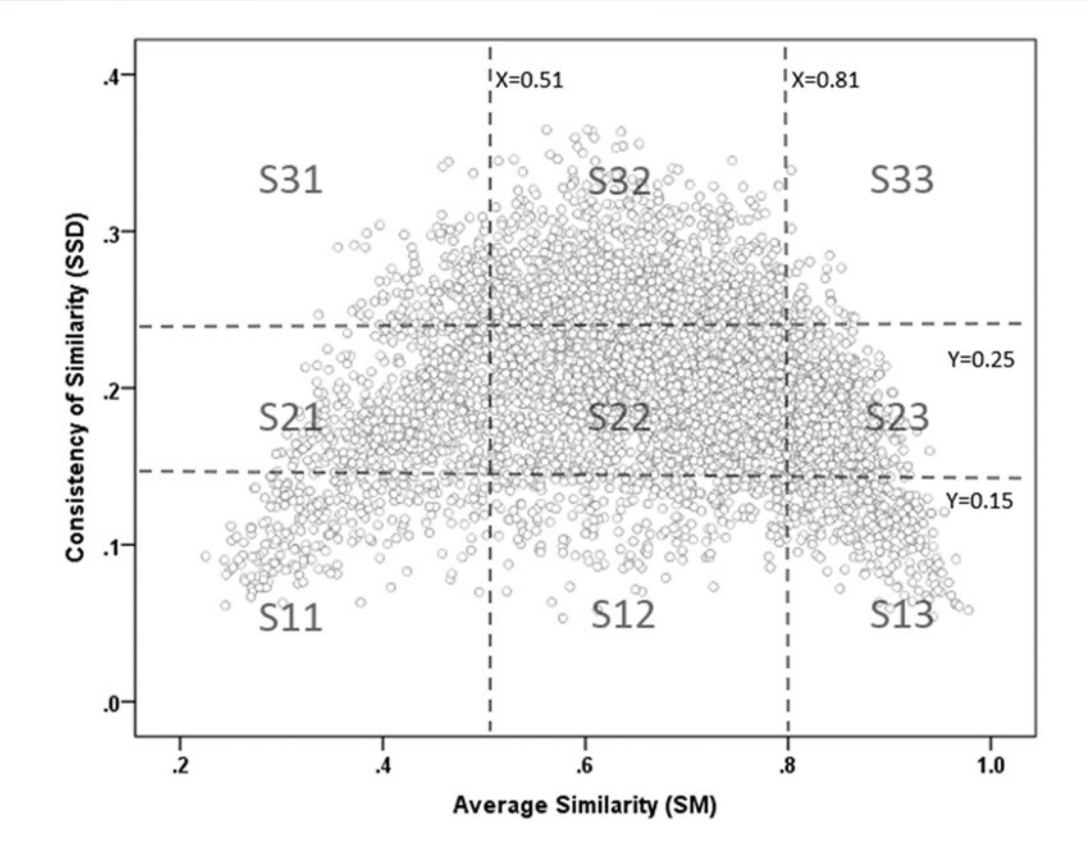
- PIAAC PSTRE PS2 module with fixed 7-item booklet. Each respondent has 7 PSTRE items in a row.
- 5 countries: GBR, IRL, JPN, NLD, USA
- 7,462 respondents ("Start, Next, Next_OK" patterns removed, resulted in 5,302 respondents in LCS)

Item Concepts and Sequence Characteristics of the Seven PSTRE Items in PIAAC PS2

	Environment				Difficulty Level	Average Sequence Length	Number of Reference Sequences	Minimal Number of Actions
	Email	Web	Word Processor	Spreadsheet				
U19a	X			X	1	19.63	4	9
U19b			X	X	2	21.18	4	12
U07		X			2	18.08	2	18
U02	X	X	X		3	53.02	5	25
U16	X				1	97.71	16	8
U11b	X				3	29.61	18	10
U23	X	X			2	28.51	1	17

Note: Reference sequences indicate the expert-predefined action sequences for each item. The minimal number of actions indicates the least number of actions to correctly solve the item. The items are ordered according to the order or appearance in the assessment.

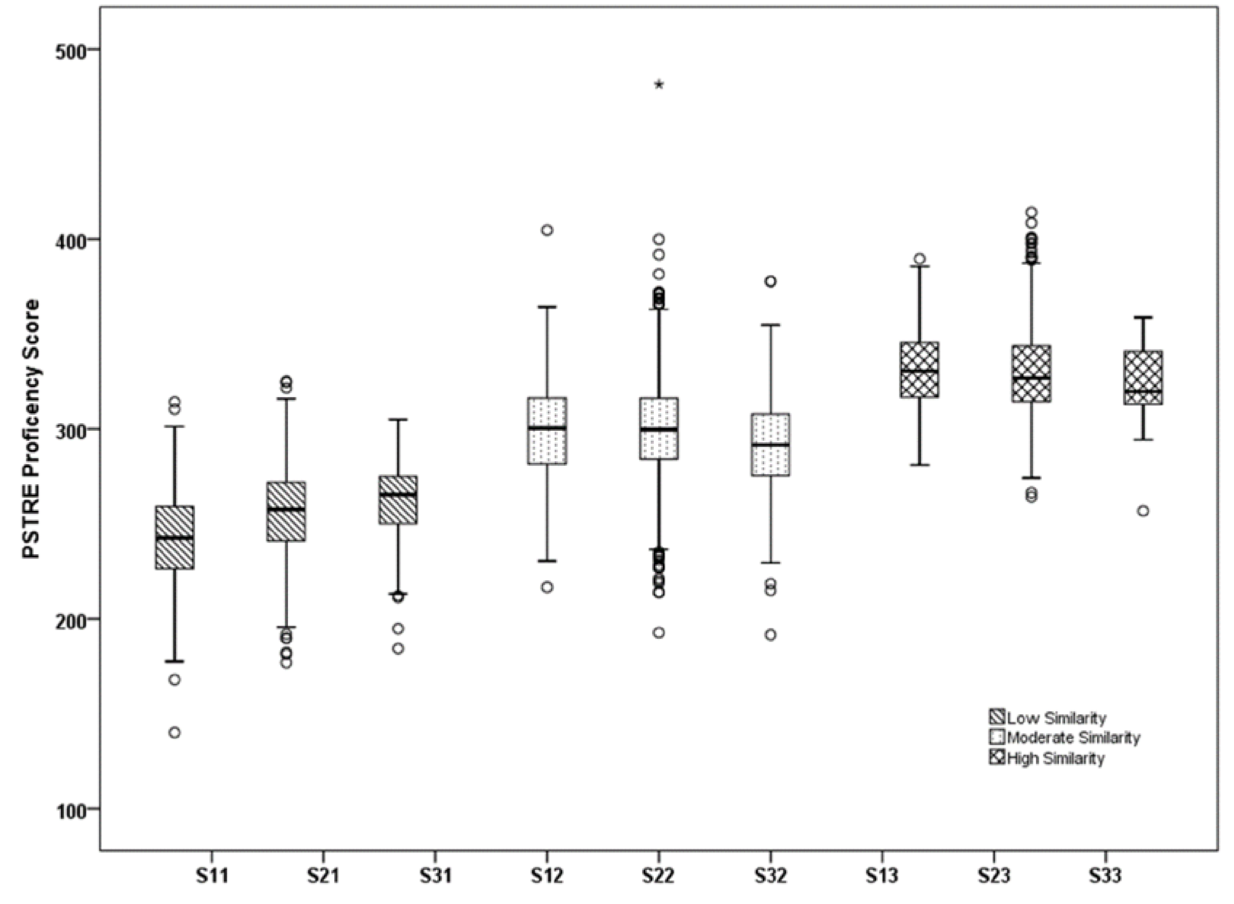
Results: Do people consistently follow pre-defined strategies in solving different tasks?



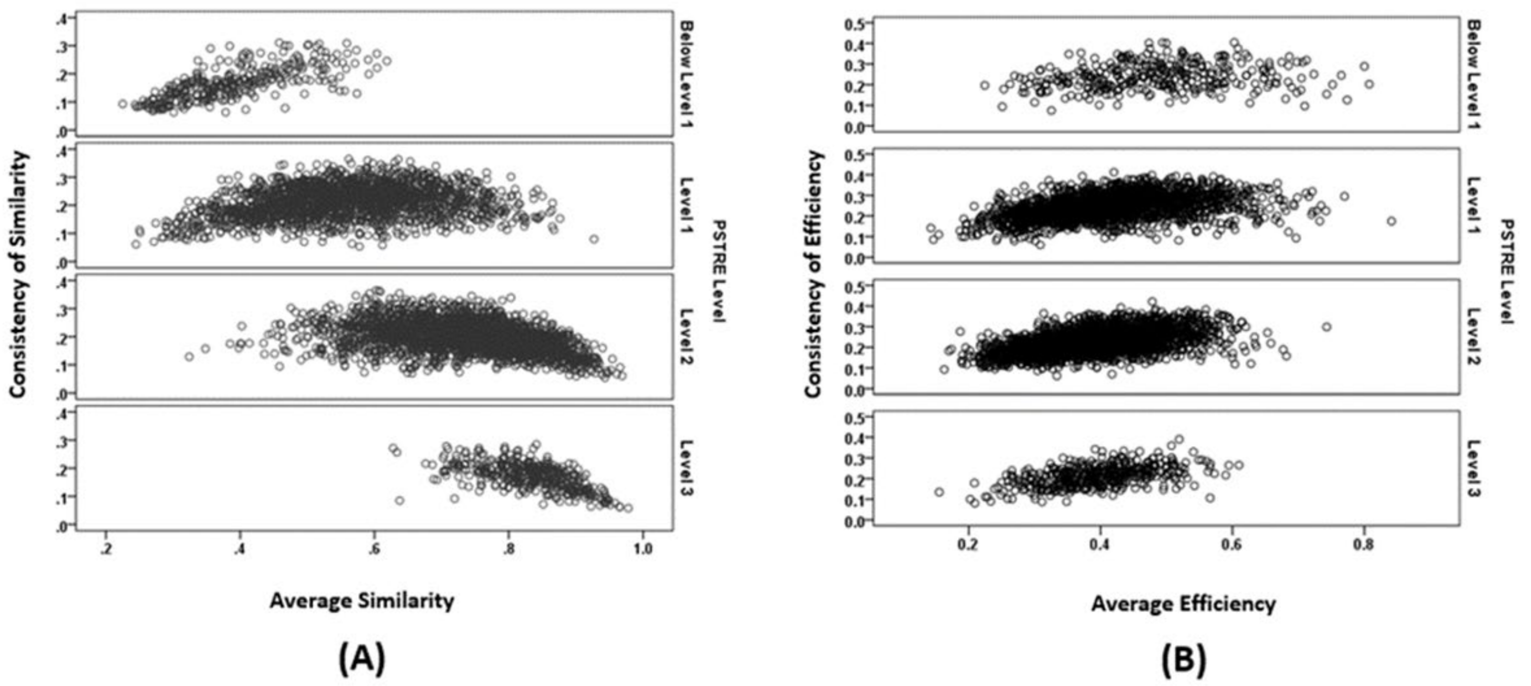
- Most respondents adopted strategies similar to the predefined ones. Small proportion of respondents in the low-similarity cells S11, S21 and S31.
- Respondents with average levels of similarity tend to display average levels of consistency (cell S22), meaning that for these respondents the distance between the observed and the reference sequences does not vary much across items.
- Respondents at the extreme of the similarity distribution, i.e. whose sequences were on average very close or very far from the reference sequence (e.g., S11 and S13), tended to do so in a very consistent way across items.

Results: Problem-solving strategies are associated with PSTRE proficiency

		Average Similarity (MEAN)		
		M1 (Low Similarity)	M2 (Moderate Similarity)	M3 (High Similarity)
Consistency (SD)	SD1 (High Consistency)	S11 High Consistency Low Similarity	S12 High Consistency Moderate Similarity	S13 High Consistency High Similarity
	SD2 (Moderate Consistency)	S21 Moderate Consistency Low Similarity	S22 Moderate Consistency Moderate Similarity	S23 Moderate Consistency High Similarity
	SD3 (Low Consistency)	S31 Low Consistency Low Similarity	S32 Low Consistency Moderate Similarity	S33 Low Consistency High Similarity

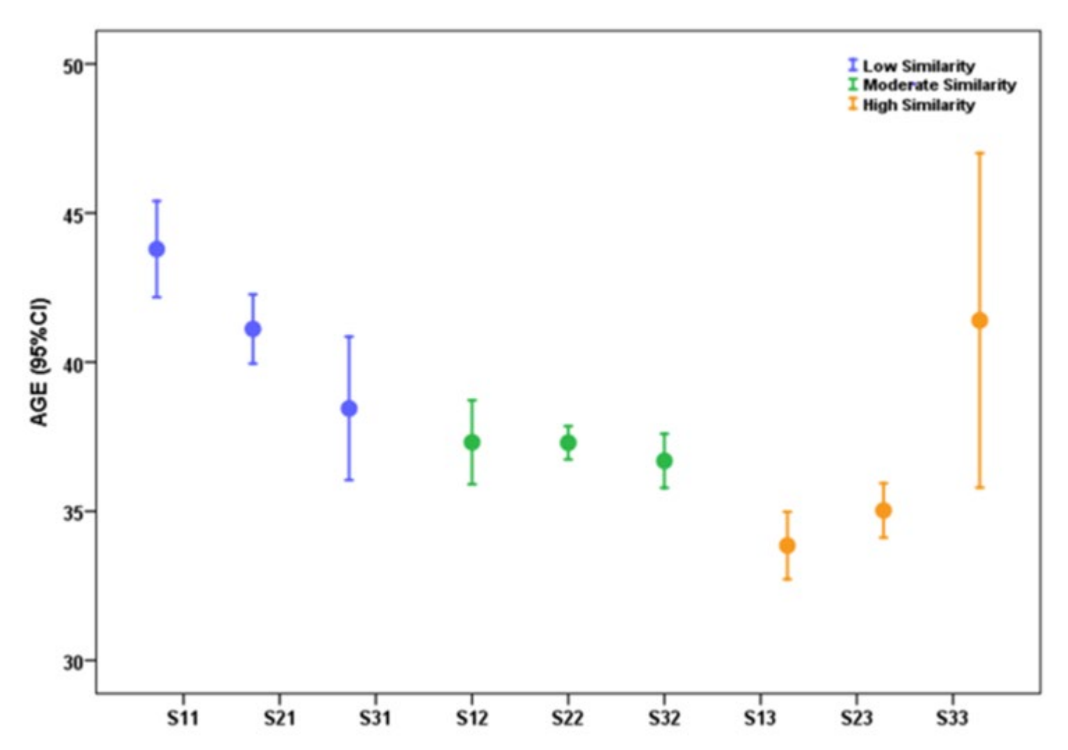


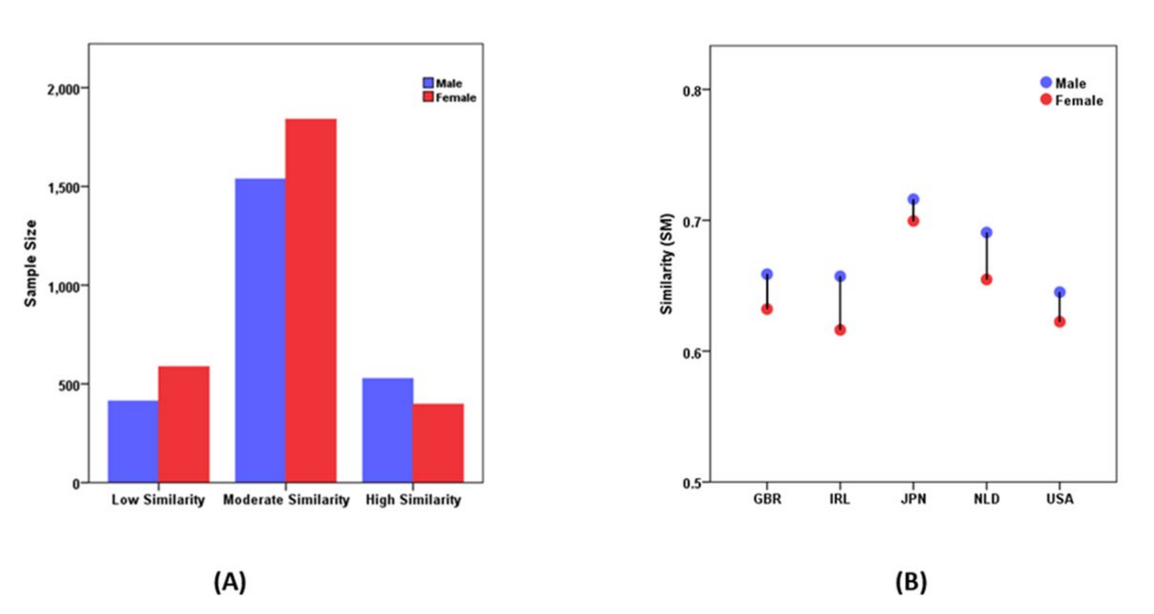
Results: Problem-solving strategies are associated with PSTRE proficiency



	N	SM	SSD	EM	ESD
Below Level 1	326	0.39 (0.08)	0.16 (0.06)	0.48 (0.11)	0.24 (0.06)
Level 1	1935	0.56 (0.11)	0.21 (0.06)	0.41 (0.10)	0.24 (0.06)
Level 2	2620	0.73 (0.10)	0.20 (0.05)	0.39 (0.08)	0.23 (0.05)
Level 3	421	0.83 (0.06)	0.17 (0.05)	0.39 (0.08)	0.21 (0.05)

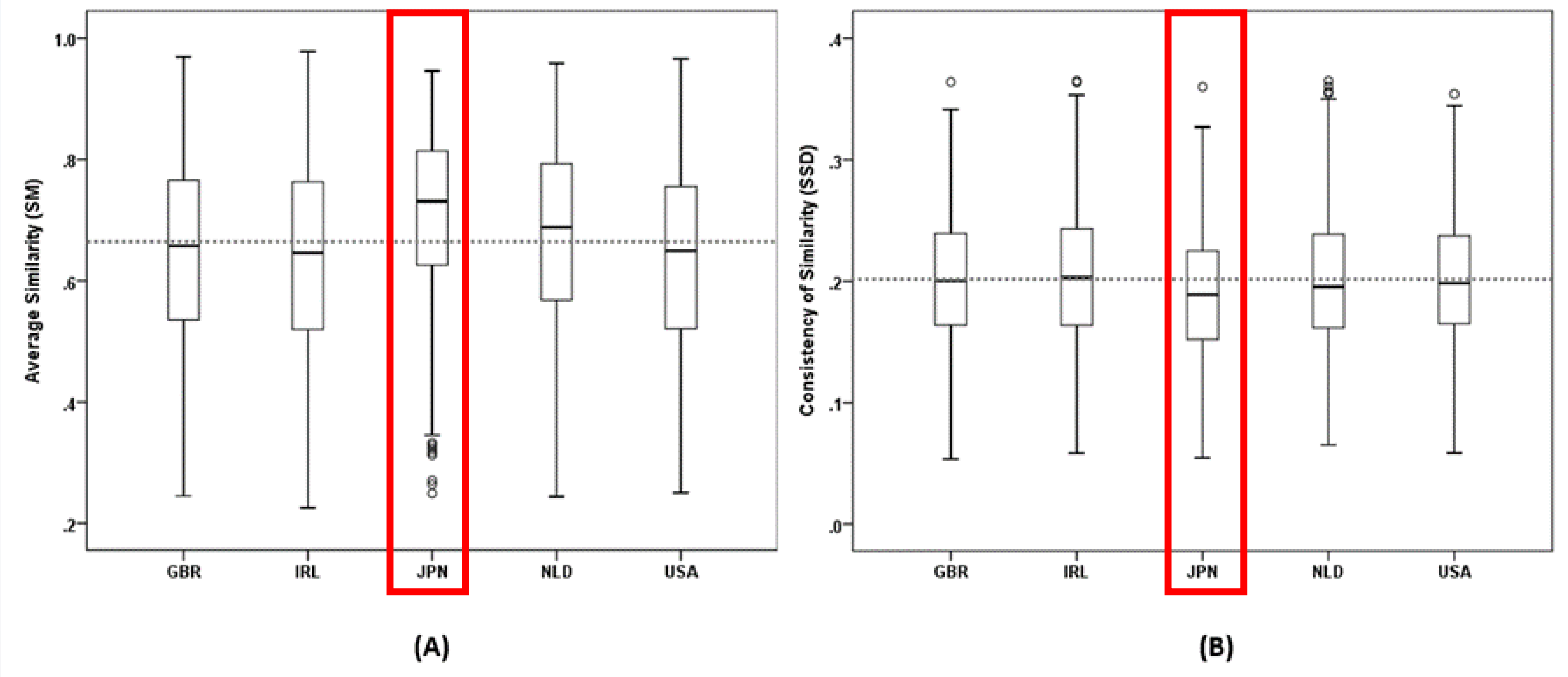
Results: Problem-solving strategies are associated with background variables



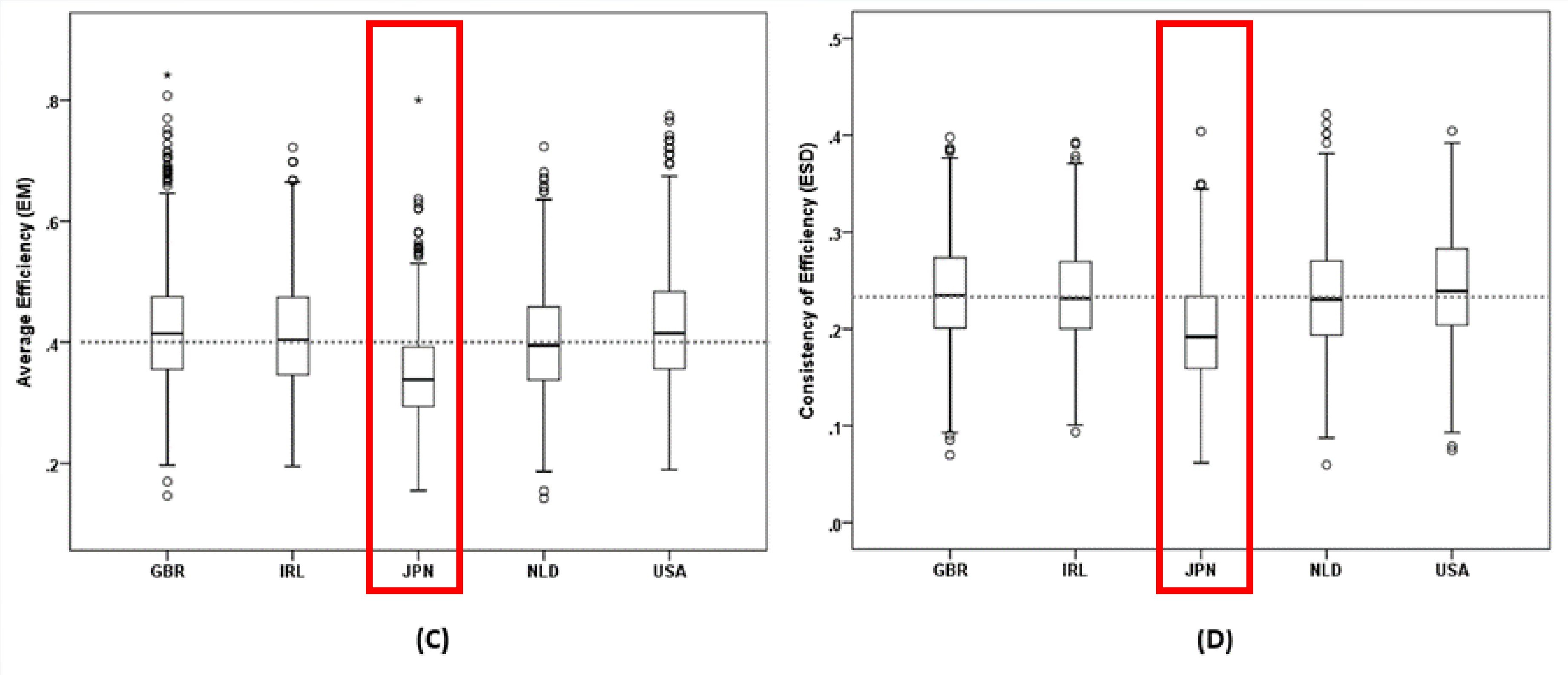


He, Q., Borgonovi, F. & Paccagnella, M. (2019). Using process data to understand adults' problem-solving behaviour in the Programme for the International Assessment of Adult Competencies (PIAAC): Identifying generalised patterns across multiple tasks with sequence mining. *OECD Education Working Papers, No. 205*, OECD Publishing, Paris.

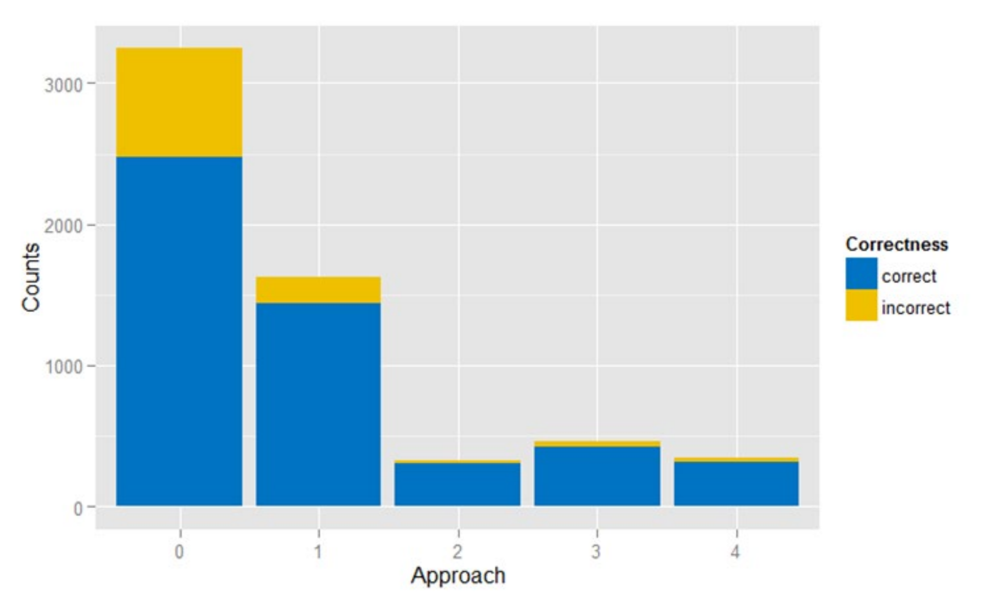
General patterns in similarity across countries



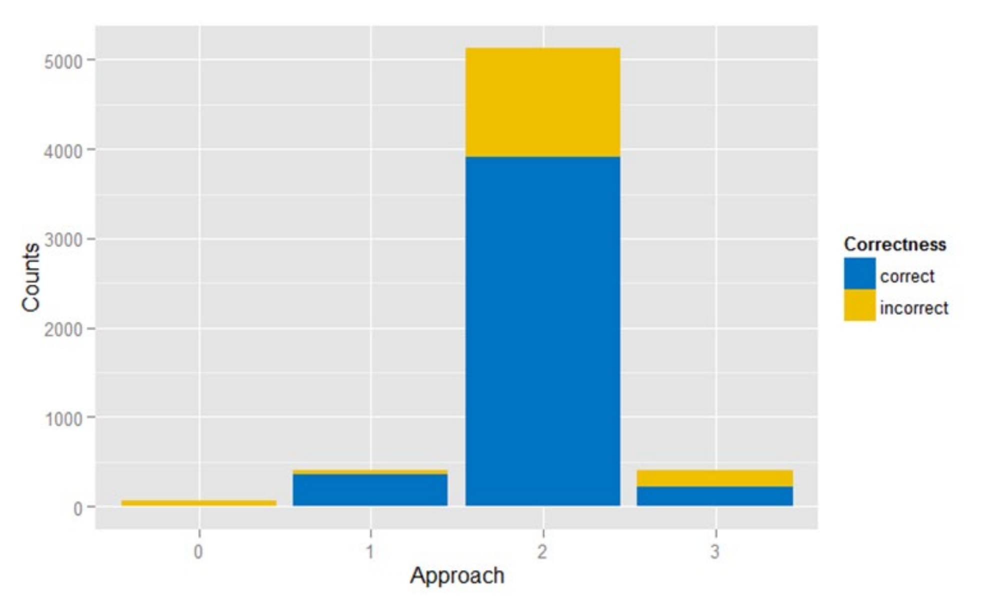
General patterns in efficiency across countries



Results: Item quality check

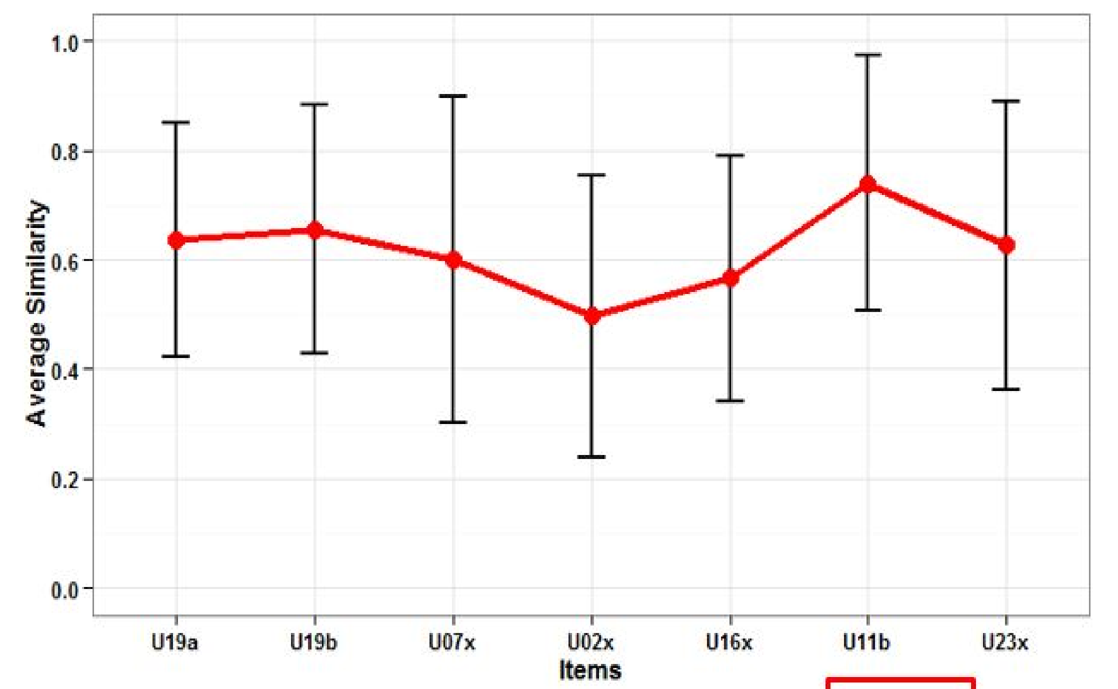


U19a



U16x

Joint modeling with response data and process data

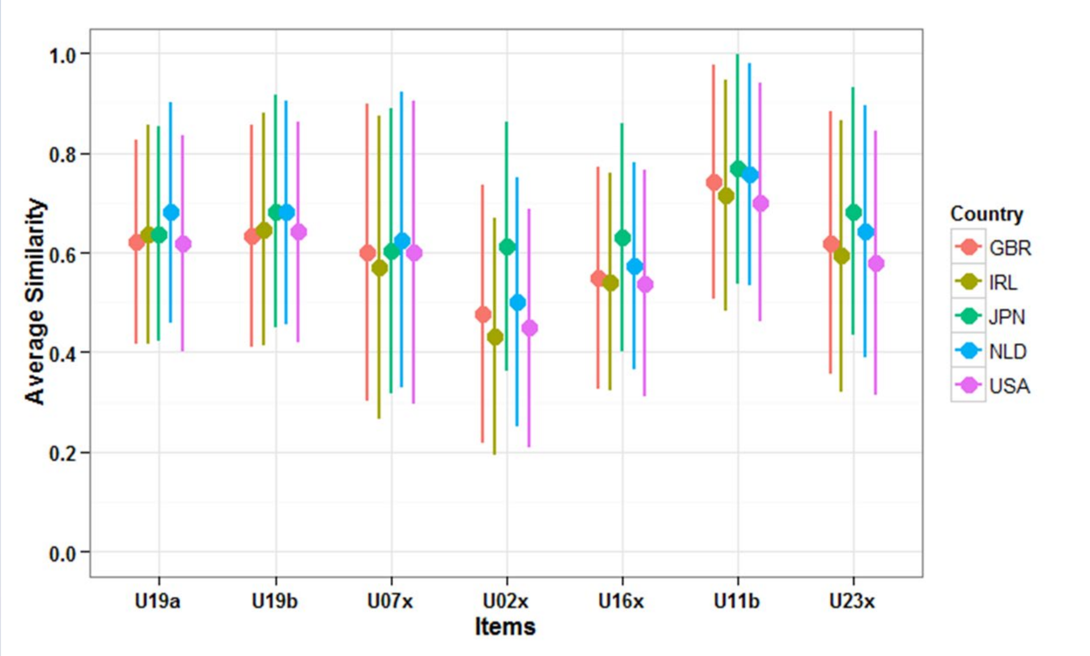


	U19a	U19b	U07x	U02x	U16x	U11b	U23x
a-slope	1.414	1.072	1.104	1.184	1.377	0.471	0.533
b-difficulty	-1.367	-0.677	-0.237	0.784	-0.773	0.774	-0.052
RP67 level	1	2	2	3	1	3	2

- Item difficulty is not always consistent with the action similarity with the predefined sequences.

- Easy to make correct actions, but hard to get final correct responses

Leveraging process data in validity issues and measurement invariance



- Measurement invariance from both response (DIF) and problem-solving process
- Equity and fairness issue in testing
- Group differences

Demo example of LCS in Program R

```
library(qualv)
library(ggplot2)
```

```
##read in  demo_data and example predefined reference sequence
DATA <- read.csv(file="data_demo.csv",header = TRUE,sep=",")
RS <- read.csv(file="RefSeq.csv",header = TRUE,sep=",")
```

```
##=====================================================
##longest common subsequence example by one person
##=====================================================

num <-1 ##pick a number 1-600

obs <- as.character(DATA$Coded_Action_Sequences[num])
obs_seq <- unlist(strsplit(obs, ", "))


result<-{}
for (j in 1:nrow(RS)){

  ref <- as.character(RS$RefSeq[j])  ##reference:predefined action sequence
  ref_seq <- unlist(strsplit(ref,", "))
  A <- LCS(obs_seq,ref_seq)
  obs <- paste(A$a,collapse=",")     ##observation action sequence
  Lobs <- as.numeric(length(A$a))    ##length of observation sequence
  ref <- paste(A$b,collapse=",")     ##reference action sequence (predefined)
  Lref <- as.numeric(length(A$b))    ##length of reference sequence
  LCS <- paste(A$LCS,collapse=",")   ##longest common subsequence
  LLCS <- as.numeric(as.character(A$LLCS))      ##length of longest common subsequence between observation and reference
  eff <- round(LLCS/Lobs,3)                     ##efficiency. length of LCS/length of obersvation
  sim <- round (LLCS/Lref,3)                    ##similarity. length of LCS/length of references
  path=j  ##path number

  LCSresult<- cbind(obs, Lobs, ref, Lref, LCS, LLCS, eff, sim, path)
  result <- as.data.frame(rbind(result,LCSresult))

}
```

Summary

- LCS could take the sequence as a whole set to calculate the sequence distance, not need to disassemble into mini-sequences.

- The length of each pair of sequences could be different, which is a big advantage in process data analysis when individual sequence is flexible to be short or long.

- LCS could be used in distance calculation for any pair of sequences, not necessary to be only between observed and predefined ones. A pairwise LCS distance could be a matrix of individual's sequence against each peer. The distance matrix could be further used for prediction and clustering.

References

- Han, Z., He, Q., & von Davier, M. (2019). Predictive Feature Generation and Selection Using Process Data from PISA Interactive Problem-Solving Items: An Application of Random Forests. *Frontiers in Psychology, 10*: 1421.
- He, Q., Borgonovi, F., Paccagnella, M. (2021). Leveraging process data to assess adults' problem-solving skills: Identifying generalized behavioral patterns with sequence mining. *Computers and Education*, *166*, 104170. https://doi.org/10.1016/j.compedu.2021.104170
- He, Q., & von Davier, M. (2016). Analyzing Process Data from Problem-Solving Items with N-Grams: Insights from a Computer-Based Large-Scale Assessment. In Y. Rosen, S. Ferrara, & M. Mosharraf (Eds.) *Handbook of Research on Technology Tools for Real-World Skill Development* (pp. 749-776). Hershey, PA: Information Science Reference.
- He, Q., & von Davier, M. (2015). Identifying Feature Sequences from Process Data in Problem-Solving Items with N-grams. In A. van der Ark, D. Bolt, S. Chow, J. Douglas & W. Wang (Eds.), *Quantitative Psychology Research: Proceedings of the 79th Annual Meeting of the Psychometric Society* (pp.173-190). New York: Springer.
- Ulitzsch, E., He, Q., Pohl, S. (2021). Using sequence mining techniques for understanding incorrect behavioral patterns on interactive tasks. *Journal of Educational and Behavioral Statistics*. https://doi.org/10.3102/10769986211010467
- Ulitzsch, E., He, Q., Ulitzsch, V., Nichterlein, A., Molter, H., Niedermeier, R., Pohl, S. (2021). Combining clickstream analyses and graph-modeled data clustering for identifying common response process using time-stamped action sequence. *Psychometrika*. https://doi.org/10.1007/s11336-020-09743-0

Thank you very much!

Welcome and appreciate any question and suggestions!

Qiwei-Britt He

Psychometric and Data Science Modeling

Educational Testing Service

qhe@ets.org