Latent Feature Extraction for Process Data via Multidimensional Scaling

Xueying Tang, Zhi Wang, Qiwei He, Jingchen Liu & Zhiliang Ying

ONLINE FIRST

Psychometrika

ISSN 0033-3123

Psychometrika
DOI 10.1007/s11336-020-09708-3

Springer

PSYCHOMETRIKA
https://doi.org/10.1007/s11336-020-09708-3

LATENT FEATURE EXTRACTION FOR PROCESS DATA VIA MULTIDIMENSIONAL SCALING

XUEYING TANG
UNIVERSITY OF ARIZONA

ZHI WANG
COLUMBIA UNIVERSITY

QIWEI HE
EDUCATIONAL TESTING SERVICE

JINGCHEN LIU AND ZHILIANG YING
COLUMBIA UNIVERSITY

Computer-based interactive items have become prevalent in recent educational assessments. In such items, detailed human–computer interactive process, known as response process, is recorded in a log file. The recorded response processes provide great opportunities to understand individuals' problem solving processes. However, difficulties exist in analyzing these data as they are high-dimensional sequences in a nonstandard format. This paper aims at extracting useful information from response processes. In particular, we consider an exploratory analysis that extracts latent variables from process data through a multidimensional scaling framework. A dissimilarity measure is described to quantify the discrepancy between two response processes. The proposed method is applied to both simulated data and real process data from 14 PSTRE items in PIAAC 2012. A prediction procedure is used to examine the information contained in the extracted latent variables. We find that the extracted latent variables preserve a substantial amount of information in the process and have reasonable interpretability. We also empirically prove that process data contains more information than classic binary item responses in terms of out-of-sample prediction of many variables.

Key words: response process, log file analysis, PIAAC, multidimensional scaling.

1. Introduction

Computer-based problem-solving items have become prevalent in large-scale assessments. These items are developed to measure skills related to problem solving in work and personal life. Thanks to the human–computer interface, it is possible to record the entire problem-solving process, as is the case of scientific inquiry items in the Programme for International Student Assessment (PISA) and Problem Solving in Technology-Rich Environments (PSTRE) items in the Programme for the International Assessment of Adult Competencies (PIAAC). The responses

Liu and Ying's research is supported by National Science Foundation Grants SES-1826540 and IIS-1633360. He's research is supported by National Science Foundation Grant IIS-1633353. The authors would like to thank Educational Testing Service for providing the data, and Hok Kan Ling for cleaning it.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s11336-020-09708-3) contains supplementary material, which is available to authorized users.

Correspondence should be made to Jingchen Liu, Department of Statistics, Columbia University,1255 Amsterdam Ave, New York, NY10027, USA. Email: jcliu@stat.columbia.edu

Published online: 22 June 2020

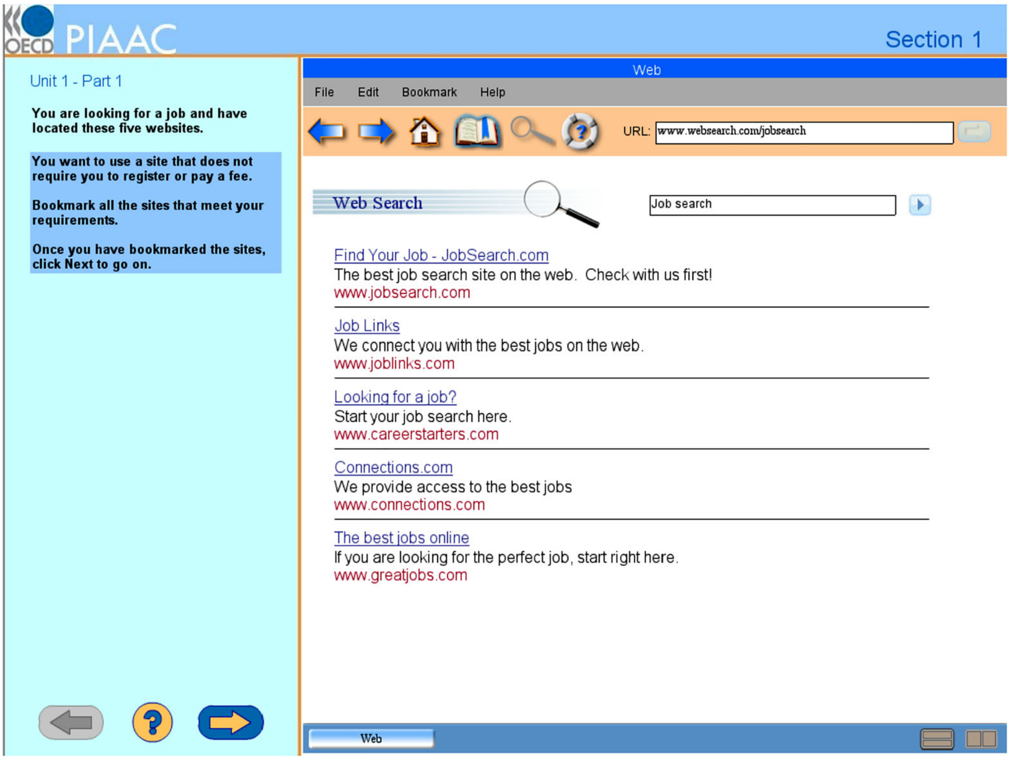

FIGURE 1.
Main page of the sample item.

of such items are complex and are often in the form of a process. More precisely, the record of each item response contains a sequence of ordered and time-stamped actions.

An example of a PIAAC PSTRE item is shown in Figures 1, 2 and 3. Figure 1 displays the main page of the item. The left panel of the main page provides item instructions. In this item, respondents are asked to identify websites that do not require registration or a fee from those listed in the web browser in the right panel. Respondents can visit a website by clicking its link. Figures 2 and 3 show the web pages of the first and the second links, respectively. Further information of the second website can be found by clicking on the "Learn More" button shown in Figure 3. If a website is considered useful, it can be bookmarked by either using the menu item "Bookmark" or clicking the bookmark icon in the tool bar. Suppose that a respondent completes the task through the following steps: click on the first link, read the first website, go back to the main page, click on the second link, and bookmark the second website by clicking the bookmark icon. All these actions are recorded in the log file in order. The sequence "Start, Click_W1, Back, Click_W2, Toolbar_Bookmark, Next" constitutes a response process.

In this paper, we present a generic method to extract useful information regarding participants from their response processes. Latent variable or latent class models have been used in the literature to summarize item responses. Existing models and methods such as item response theory models (Lord, 1980) and cognitive diagnosis models (Rupp, Templin, & Henson, 2010) are not directly applicable to response processes. The analysis of process data is difficult for several reasons. First, response processes are in a nonstandard format. A response process is a sequence of actions and each action is a categorical variable. In addition, process length varies across individuals. Because of the nonstandard format, classic models do not apply to process data. Second, computer-based

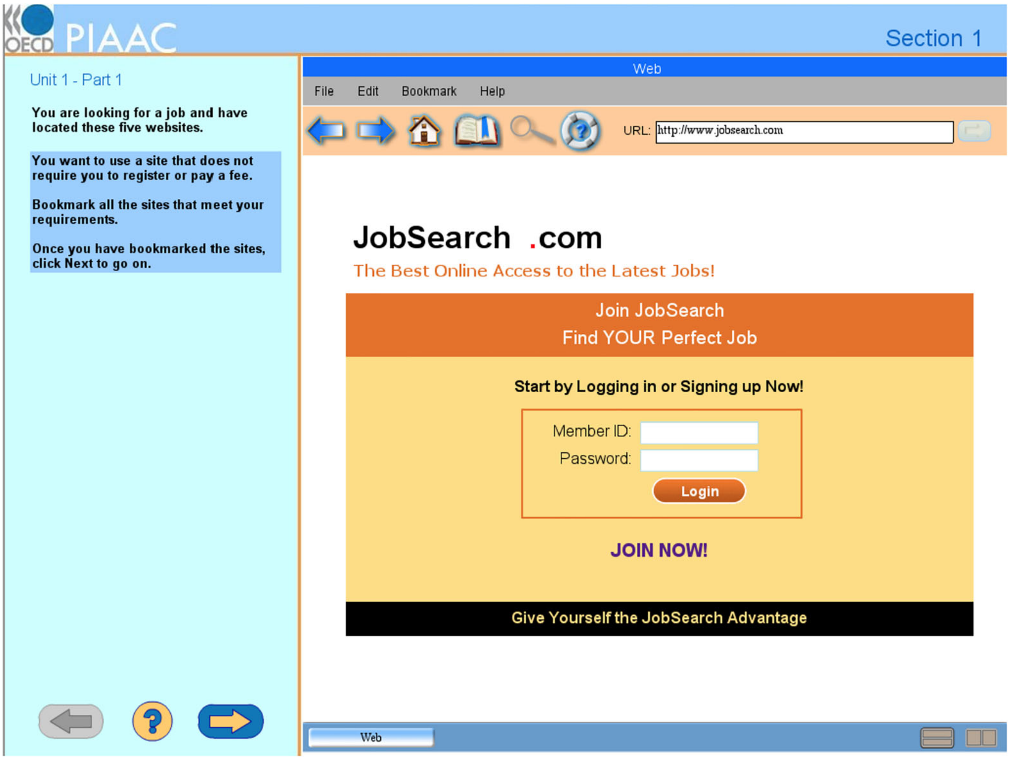

FIGURE 2.
Website in the first link in Figure 1.

assessments and their log files cover a large variety of items. Many human–computer interfaces generate log files that records detailed information of response processes. This makes confirmatory analysis practically infeasible due to the large amount and variety of items. It is too expensive to perform confirmatory analysis for each potential human–computer interface and then verify it empirically. Furthermore, the cognitive process of human–computer interaction is not thoroughly understood, which adds to the difficulty of confirmatory analysis. Lastly, response processes are often very noisy. For instance, the lagged correlations of action occurrences are often very close to zero, that is, response processes behave like white noise from an autoregressive process viewpoint.

Assessment of data beyond traditional responses has been studied previously. It has been shown that item response time can reveal test-taker response behaviors that are helpful for test design (Qian, Staniewska, Reckase, & Woo, 2016; van der Linden, 2008). Models have been proposed to perform cognitive assessments using both traditional responses and response time (Klein Entink, Fox, & van der Linden, 2009, Wang, Zhang, Douglas, & Culpepper, 2018, Zhan, Jiao, & Liao, 2018). The study of process data is at a more preliminary stage. Most works such as Greiff, Niepel, Scherer, and Martin (2016) and Kroehne and Goldhammer (2018) first summarized process data into several variables and then investigated their relationship with other variables of interest using standard statistical methods. The design of these summary variables is usually item specific and thus hard to generalize. He and von Davier (2015, 2016) explored the association between action sequence patterns and traditional responses using n-grams. Although the procedure of extracting n-gram features is generic, nontrivial preprocessing steps are needed to enhance the interpretability of the selected n-grams.

The objective of the present analysis is to perform exploratory analysis on process data. In particular, we propose a generic method to extract features (latent variables) from response

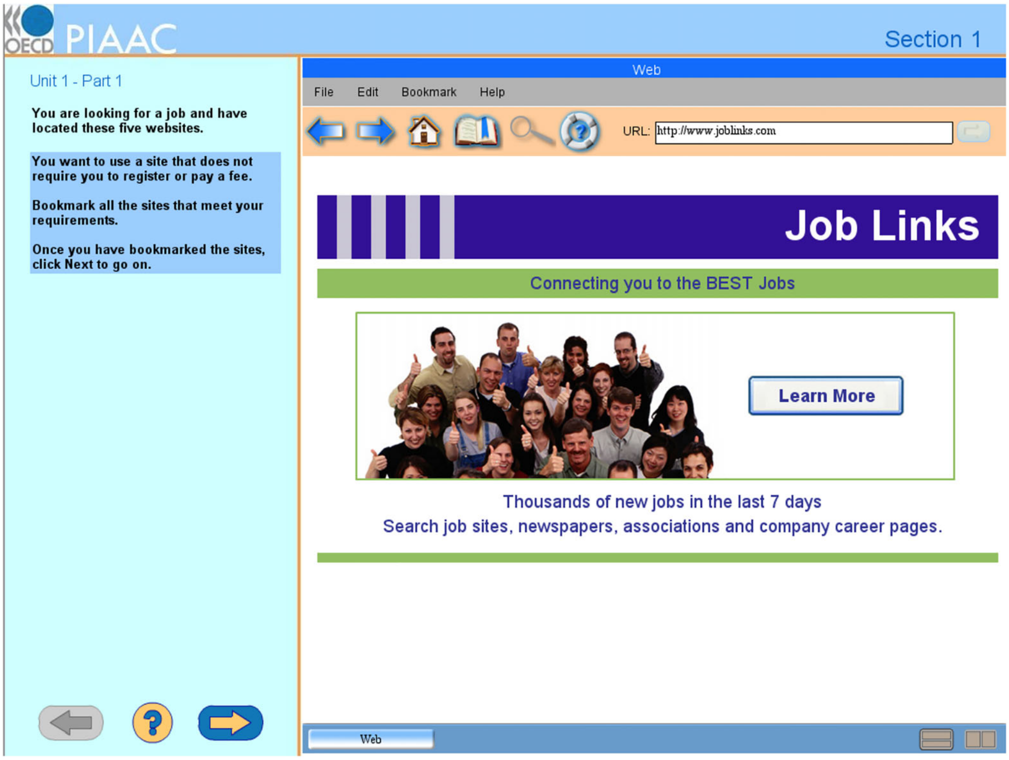

FIGURE 3.
Website in the second link in Figure 1.

processes. The proposed method does not rely on prior knowledge of the items or the response processes and is applicable essentially to all process responses. We apply it to all 14 PIAAC PSTRE items that cover a range of human–computer interfaces.

The basic technique of our proposed method is multidimensional scaling (Borg & Groenen, 2005). It constructs features based on the relative differences among individuals. Though numerous variants of multidimensional scaling (MDS) exist, their common goal is to locate objects in a vector space according to their pairwise dissimilarities in such a way that similar objects are close together, while less similar objects are far apart. MDS has been used for data visualization and dimension reduction in cognitive diagnosis, test analysis, and many other areas of psychometrics (Karni & Levin, 1972; Meyer & Reynolds, 2018; Shoben, 1983; Skager, Schultz, & Klein, 1966; Subkoviak, 1975). In the context of process data analysis, if the differences between two processes can be properly summarized by a dissimilarity measure, then the coordinates obtained from MDS can be treated as features storing information of the original processes. With a proper rotation, each feature describes the variation of certain ability or behavior pattern among the group of respondents.

We use a prediction procedure to demonstrate that response processes contain more information than traditional item responses. We denote the features extracted from response processes by θ. For each response process, there is a binary response, denoted by r, indicating whether the respondent has successfully accomplished the task. To compare the information contained in θ and r, we adopt a third variable, denoted by y (such as numeracy score, literacy score, etc.), and inspect the prediction of y based on r and that based on θ. In the empirical analysis of PSTRE in PIAAC, we find that the prediction based on θ outperforms that based on r for a wide range of y variables including assessment scores, basic demographic variables, and some background variables.

The rest of this paper is organized as follows. In Section 2, we introduce a dissimilarity measure for action sequences and describe the proposed feature extraction procedure. A simulation study is presented in Section 3 to demonstrate the procedure and how the latent structure of action sequences is reflected in extracted features. In Section 4, we show through a case study of PIAAC PSTRE item response processes that features extracted from process data contain much richer information than binary responses. Section 5 contains some concluding remarks.

2. Feature Extraction via Multidimensional Scaling

Consider a problem-solving item in which a student takes a number of actions to complete a task. We use $\mathcal{A} = \{a_1, \ldots, a_N\}$ to denote the set of possible actions of this item where N is the number of distinct actions. A response process is a sequence of actions $\boldsymbol{s} = (s_1, \ldots, s_L)$ where each s_i is an action in $\mathcal{A}$ and L is the process length, i.e., the number of actions taken in the response process. An action in $\mathcal{A}$ may appear multiple times or never appear in $\boldsymbol{s}$. We observed the response processes of n students and use subscript to index different observations: $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$. The process length also varies among individuals; we use L_i to denote the length of $\boldsymbol{s}_i$. The heterogeneous length of response processes for the same item is one of the technical difficulties in process data analysis. In what follows, we describe a procedure that transforms the response processes with heterogeneous length to homogeneous-dimension latent vectors that may be used for standard analysis.

The core of the procedure is MDS, which has been widely used as a data visualization and dimension reduction tool in many fields including psychometrics (Takane, 2006). The goal of MDS is to locate objects in a vector space according to their pairwise dissimilarities in such a way that similar objects are close together, while dissimilar objects are far apart. We begin the discussion with a description of a dissimilarity measure between discrete action sequences. This measure is key to the subsequent application of multidimensional scaling, and it summarizes the variation among response processes. An appropriate dissimilarity measure should accommodate three characteristics of response processes. First, process data are a collection of discrete processes on which arithmetic calculation cannot be performed. Second, processes from different respondents are of very different lengths. Third, the order of actions matters. Although the order of actions may not affect the final outcome of the task, it can reflect respondents' problem solving strategies and other useful information.

Based on these considerations, we adopt the following dissimilarity measure that was first proposed in Gómez-Alonso and Valls (2008). Let $\boldsymbol{s}_i = (s_{i1}, \ldots, s_{iL_i})$ and $\boldsymbol{s}_j = (s_{j1}, \ldots, s_{jL_j})$ be two action sequences. Define the dissimilarity between $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ as

$$d(\boldsymbol{s}_i, \boldsymbol{s}_j) = \frac{f(\boldsymbol{s}_i, \boldsymbol{s}_j) + g(\boldsymbol{s}_i, \boldsymbol{s}_j)}{L_i + L_j}, \tag{1}$$

where $f(\boldsymbol{s}_i, \boldsymbol{s}_j)$ quantifies the dissimilarity among the actions that appear in both $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ and $g(\boldsymbol{s}_i, \boldsymbol{s}_j)$ is the count of actions appearing in only one of $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$.

We now provide the precise definition of f and g. For an action $a \in \mathcal{A}$, let $\boldsymbol{s}^a$ be a sequence consisting of chronologically ordered positions of a in sequence $\boldsymbol{s}$. The length of $\boldsymbol{s}^a$, L^a, is the number of times that a appears in $\boldsymbol{s}$. We use $\boldsymbol{s}^a(k)$ to denote the kth element of $\boldsymbol{s}^a$, namely the position of the kth appearance of a in $\boldsymbol{s}$. For two sequences $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, let C_{ij} denote the set of actions that appear in both $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ and U_{ij} denote the set of actions that appear in $\boldsymbol{s}_i$ but not in $\boldsymbol{s}_j$. Then, $f(\boldsymbol{s}_i, \boldsymbol{s}_j)$ and $g(\boldsymbol{s}_i, \boldsymbol{s}_j)$ are defined as

$$f(\boldsymbol{s}_i, \boldsymbol{s}_j) = \frac{\sum_{a \in C_{ij}} \sum_{k=1}^{K_{ij}^a} |\boldsymbol{s}_i^a(k) - \boldsymbol{s}_j^a(k)|}{\max\{L_i, L_j\}}, \tag{2}$$

and

$$g(\boldsymbol{s}_i, \boldsymbol{s}_j) = \sum_{a \in U_{ij}} L_i^a + \sum_{a \in U_{ji}} L_j^a, \tag{3}$$

where $K_{ij}^a = \min(L_i^a, L_j^a)$.

We use a simple example to demonstrate how the dissimilarity is calculated. Consider a set of four possible actions $\mathcal{A} = \{X, Y, Z, W\}$ and two sequences, $\boldsymbol{s}_1 = (X, Y, X, Y, Z)$ and $\boldsymbol{s}_2 = (W, X, Y, W)$. Since X and Y appear in both sequences, $C_{12} = \{X, Y\}$. Action X appears in $\boldsymbol{s}_1$ at positions 1 and 3 and appears in $\boldsymbol{s}_2$ in position 2, so $\boldsymbol{s}_1^X = (1, 3)$ and $\boldsymbol{s}_2^X = (2)$. The difference between $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ in the appearance of X is $|1 - 2| = 1$. Similarly, we can find $\boldsymbol{s}_1^Y = (2, 4)$, $\boldsymbol{s}_2^Y = (3)$ and the difference in the appearance of Y is $|2 - 3| = 1$. Therefore, $f(\boldsymbol{s}_1, \boldsymbol{s}_2) = (|1 - 2| + |2 - 3|)/5 = 0.4$. Since $U_{21} = \{W\}$ and $U_{12} = \{Z\}$ with W appearing twice in $\boldsymbol{s}_2$ and Z appearing once in $\boldsymbol{s}_1$, $g(\boldsymbol{s}_1, \boldsymbol{s}_2) = 2 + 1 = 3$. According to (1), $d(\boldsymbol{s}_1, \boldsymbol{s}_2) = (0.4 + 3)/9 = 0.38$.

The calculation of the dissimilarity described in (1) does not require inputs of informative behavior patterns or the meaning of each action. This is crucial for our automated feature extraction procedure at the exploratory stage of the analysis.

For action sequences $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$, let an $n \times n$ symmetric matrix $\boldsymbol{D} = (d_{ij})$ denote their dissimilarity matrix, where $d_{ij} = d(\boldsymbol{s}_i, \boldsymbol{s}_j)$ measures the dissimilarity between $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, $i, j = 1, \ldots, n$. Higher dissimilarities indicate larger differences and the dissimilarity between two identical objects is zero, namely $d_{ii} = 0$ for $i = 1, \ldots, n$. MDS maps each action sequence to a latent vector $\boldsymbol{x}$ in the K-dimensional Euclidean space $\mathbb{R}^K$ such that they govern the dissimilarities. Mathematically, applying MDS to objects with dissimilarity matrix $\boldsymbol{D}$ essentially minimizes

$$\sum_{i<j} \left(d_{ij} - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|\right)^2 \tag{4}$$

with respect to $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$, where $\boldsymbol{x}_i \in \mathbb{R}^K$ is the latent vector of $\boldsymbol{s}_i$ in $\mathbb{R}^K$ and $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T(\boldsymbol{x}_i - \boldsymbol{x}_j)}$. Many algorithms have been proposed to solve the optimization problem. For simplicity, we use stochastic gradient descent (Robbins & Monro, 1951) to minimize (4).

Combining the calculation of the dissimilarity matrix and MDS, we present the feature extraction procedure for process data.

Procedure 1. (Feature extraction for process data)

1. *Compute the dissimilarity matrix $\boldsymbol{D}$ of n action sequences $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n$ by calculating the pairwise dissimilarities d_{ij}, $1 \le i, j \le n$ according to* (1)–(3).
2. *Obtain K raw features $\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_K$ by minimizing* (4).
3. *Obtain K principal features $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K$ by performing principal component analysis* (*PCA*) *on the K raw features.*

Procedure 1 extracts features with homogeneous dimension from action sequences with heterogeneous length. These features have a standard form and, as we will show in the simulation and case study, contain compressed information of the original sequences. Therefore, they can be easily incorporated as a surrogate of the action sequences in well-developed statistical models such as (generalized) linear models to study how process data reflect respondents' latent traits and how it is related to other quantities of interest. We will demonstrate how these can be achieved in the next two sections.

Principal component analysis is performed in Step 3 of Procedure 1 mainly for seeking feature interpretations. As we will show in the case study, the first several principal features usually have clear interpretations, although the feature extraction procedure does not take into account the meaning of actions.

Procedure 1 requires the specification of K, the number of features to be extracted. If K is too small, there are not enough features to characterize the variation of action sequences, leading to substantial information loss in extracted features. On the other hand, if K is too large, some features can be redundant and can cause overfitting and instability in subsequent analyses. A suitable K can be chosen by m-fold cross-validation. We randomly split the $n(n-1)/2$ pairwise dissimilarities into m disjoint subsets. For each candidate value of K and each subset of dissimilarities, we perform MDS using the rest of dissimilarities and calculating the discrepancy between the estimated and true dissimilarities for the subset. The value of K that produces the smallest total discrepancy among m subsets is chosen as the number of features to be extracted. This cross-validation procedure is summarized in Procedure 2.

We conclude this section with a remark on the random split in Step 2 of Procedure 2. To guarantee that the validation loss $V(K)$ is computable, each $\Omega_{(-q)}$ must include at least one index pair involving i for $i = 1, \ldots, n$. A random split of Ω can violate this requirement but with only a slim chance if n is moderately large. For implementation, one can check the requirement after generating a random split. If it is violated, simply repeat the generation and checking steps until an appropriate split is obtained.

Procedure 2. (Choose K by cross-validation)

1. *Randomly split* $\Omega = \{(i, j) : i < j; i, j = 1, \ldots, n\}$ *into* m *disjoint subsets* $\Omega_1, \Omega_2, \ldots, \Omega_m$.
2. *For each candidate value of* K *and each* q *in* $\{1, 2, \ldots, m\}$, *obtain* $\boldsymbol{x}_i^{(K,q)}$, $i = 1, \ldots, n$, *by minimizing*

$$\sum_{(i,j)\in\Omega_{(-q)}} \left(d_{ij} - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|\right)^2$$

   with respect to $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, *where* $\Omega_{(-q)} = \Omega \backslash \Omega_q$.
3. *For each candidate value of* K, *calculate*

$$V(K) = \sum_{q=1}^{m} \sum_{(i,j)\in\Omega_q} \left(d_{ij} - \|\boldsymbol{x}_i^{(K,q)} - \boldsymbol{x}_j^{(K,q)}\|\right)^2.$$

4. *Choose* K *that produces the smallest* $V(K)$.

3. Simulations

In this section, we demonstrate the proposed feature extraction procedure on simulated data.

3.1. Data Generation

Twenty-six possible actions ($N = 26$) are considered in our simulations. Each possible action is denoted by an upper-case English letter, namely $\mathcal{A} = \{\text{A}, \text{B}, \ldots, \text{Z}\}$ with $a_1 = \text{A}$ and $a_N = \text{Z}$. We use A and Z to denote the start and the end of an item. As a result, each action sequence starts with A and ends with Z.

The action sequences used in this section are generated from a Markov model, which is characterized by a probability transition matrix $\mathbf{P} = (p_{ij})_{1\le i,j\le N}$, whose element in the ith row and jth column is the probability that the next action is a_j given the current action is a_i, i.e., $P(s_{t+1} = a_j \mid s_t = a_i) = p_{ij}$. Because of the special roles of A and Z, the first element in each row of $\mathbf{P}$ is zero and all the elements in the last row except for the last one are zeros. Therefore, the Markov model for generating action sequences is determined by the $(N-1)\times(N-1)$ submatrix in the upper right corner of $\mathbf{P}$. We call this submatrix the core matrix of $\mathbf{P}$ and denote it by $\tilde{\mathbf{P}}$. The probability transition matrices used in our simulation study are randomly generated. The way in which they are generated will be explained in detail in the experiment settings. Given a probability transition matrix $\mathbf{P}$, we generate an action sequence by starting from A and sampling the subsequent actions according to $\mathbf{P}$ until Z appears.

3.2. *Experiment Settings*

We consider two strategies for generating action sequences. With strategy I, a set of n action sequences are generated from the previous Markov model under two different transition matrices, $n/2$ sequences for each matrix. Action sequences generated from this strategy have a latent group structure. Sequences generated from the same transition matrix form a group and tend to be similar. The two probability transition matrices, $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$, are randomly generated. More specifically, $\mathbf{P}^{(g)}$ is generated by first constructing an $(N-1)\times(N-1)$ matrices $\mathbf{U}^{(g)} = (u_{ij}^{(g)})_{1\le i,j\le N-1}$ for $g = 1, 2$. The elements of $\mathbf{U}^{(g)}$ are generated independently from a uniform distribution on interval $[-10, 10]$. Then, $\tilde{\mathbf{P}}^{(g)} = (\tilde{p}_{ij}^{(g)})_{1\le i,j\le N-1}$, the core matrix of $\mathbf{P}^{(g)}$, is computed from $\mathbf{U}^{(g)}$ by

$$\tilde{p}_{ij}^{(g)} = \frac{\exp(u_{ij}^{(g)})}{\sum_{l=1}^{N-1}\exp(u_{il}^{(g)})}. \tag{5}$$

In strategy II, each of n action sequences is generated from a unique probability transition matrix. To construct these matrices, we first obtain a uniform matrix $\mathbf{U}$ as in strategy I. Then, we draw n independent samples, $\theta_0^{(1)}, \ldots, \theta_0^{(n)}$, from $N(0, 4)$ and compute the core matrix $\tilde{\mathbf{P}}^{(i)}$ for the ith sequence according to

$$\tilde{p}_{jk}^{(i)} = \frac{\exp(\theta_0^{(i)} u_{jk})}{\sum_{l=1}^{N-1}\exp(\theta_0^{(i)} u_{jl})}. \tag{6}$$

With this strategy, sequences with similar θ_0 resemble each other. In other words, θ_0 serves as a continuous latent variable determining the characteristics of the sequences.

We consider three choices of n, 200, 500, and 1000. For each strategy and each choice of n, we generate 100 sets of action sequences and extract features according to Procedure 1. The number of features to be extracted is chosen by fivefold cross-validation described in Procedure 2.

To show that extracted features retain the information in action sequences, we derive several variables from action sequences for each dataset and examine how well these derived variables can be predicted from the extracted features. Good prediction performances indicate that a significant amount of information in action sequences is preserved in extracted features. The derived variables are indicators describing whether an action or an action pair appears in a sequence. We say an action pair (a_i, a_j) appears in a sequence if both actions appear in the sequence and action a_j is immediately after action a_i. For example, in sequence "A, B, D, Z", both action B and action pair (B, D) appear. Although both B and Z appear and Z appears later than B, they are not contiguous in the sequence. Therefore, action pair (B, Z) does not appear in the sequence. We do not consider indicators for actions and action pairs that appears fewer than $0.05n$ times or more than $0.95n$ times in a dataset. Logistic regression is used to predict the derived variables from extracted

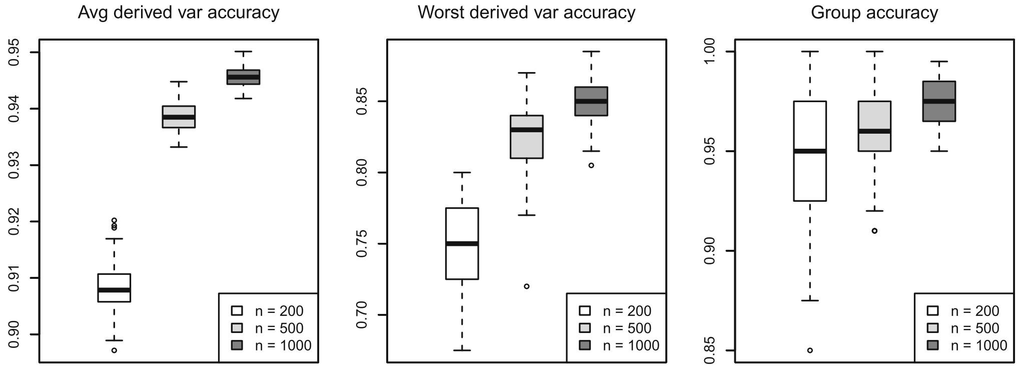

FIGURE 4.
Simulation results for datasets generated from strategy I.

features. For each data set, n sequences are split into training and test sets in the ratio of 4:1. A logistic regression model is estimated for each derived variable on the training set, and its prediction performance is evaluated on the test set. The average prediction accuracy and the worst prediction accuracy among all the derived variables are recorded for each dataset.

To inspect the ability of the extracted features in unveiling the latent structures in action sequences, we build a logistic regression model to identify the group structure from the extracted features for datasets generated from strategy I and a linear regression model of θ_0 on the extracted features for datasets generated from strategy II. The models are fitted on the training set. The logistic model of group identity is evaluated by the prediction accuracy on the test set, while the linear regression model of θ_0 is evaluated by out-of-sample R^2 (OSR^2), the square of the correlation between the predicted and true values. As an analogy to the in-sample R^2 in linear regression, a higher OSR^2 indicates a better prediction performance.

3.3. Results

Figures 4 and 5 display the results for datasets generated by strategies I and II, respectively. The left and middle panels of both figures present the average and worst prediction accuracy for derived variables. Under all the settings, for almost all datasets, the averaged prediction accuracy is greater than 0.9 and the worst prediction accuracy is greater than 0.7. These results demonstrate that the derived variables can be predicted well and imply that a significant amount of information in action sequences is compressed into the extracted features.

The right panel of Figure 4 presents the prediction accuracy for group identity. For most of the datasets, the prediction accuracy is higher than 0.9, indicating that group structures in action sequences can be identified very accurately by extracted features. The right panel of Figure 5 gives the OSR^2 for predicting θ_0. It reflects that continuous latent characteristics in action sequences can be captured well by features extracted from Procedure 1 as the correlation between the predicted and true values is higher than 0.8 for most of the datasets.

To take a closer look at how the extracted features reveal the latent structure of action sequences, in Figure 6, we plot the first two principal features for one dataset of 1000 sequences under each strategy. For the dataset generated from strategy I (left panel of Figure 6), the group structure is clearly shown in the figure and the two groups can be roughly separated by a horizontal line at zero. The data shown in the right panel of Figure 6 are generated from strategy II. It is evident that sequences located closer have similar latent characteristics.

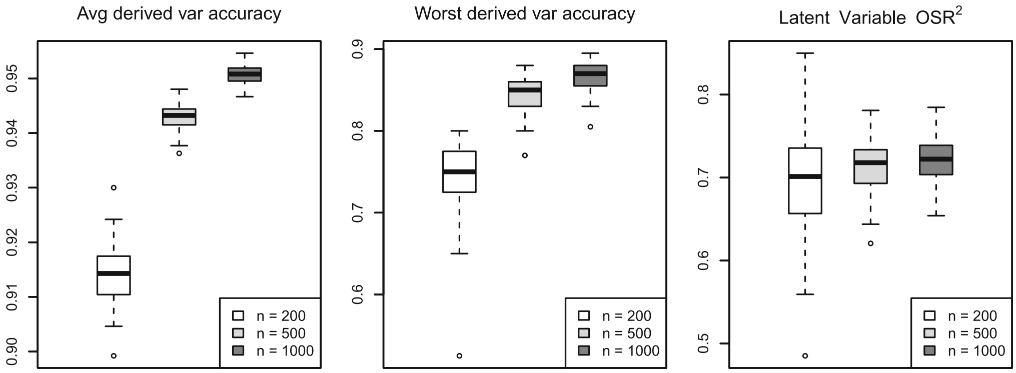

FIGURE 5.
Simulation results for datasets generated from strategy II.

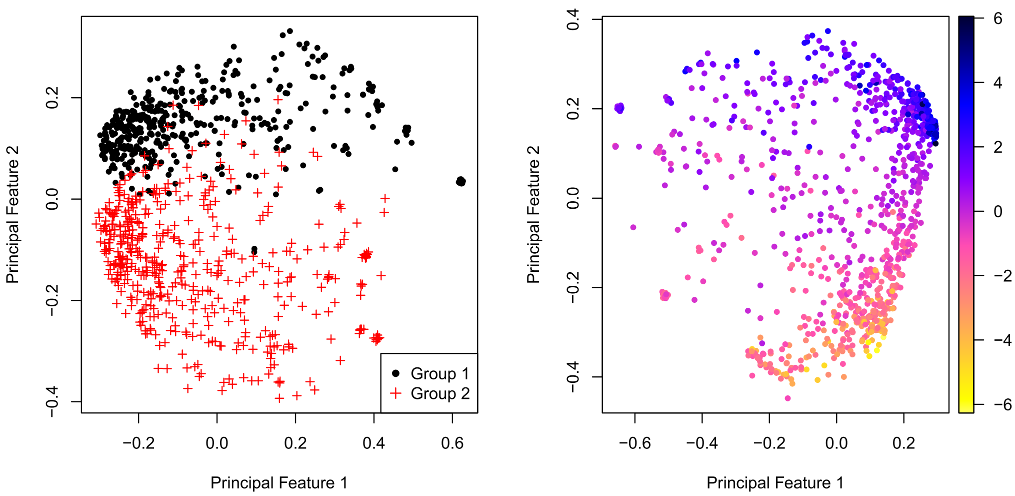

FIGURE 6.
First two principal features for one dataset with 1000 sequences generated from strategy I (left) or strategy II (right). The data points in the right panel are colored according to the value of the latent variable θ_0.

4. Case Study

4.1. Data

The data considered in this study come from the PIAAC 2012 survey from five countries: the UK, Ireland, Japan, the Netherlands, and the USA. There are 14 PSTRE items and 11,464 respondents in the dataset in total. Each person responded to all or a subset of the 14 items. There are 7620 respondents who answered 7 items and 3645 respondents who answered all 14 items. For each item, there were around 7500 respondents. Altogether there are 106,096 respondent–item pairs. Both the response process and the response outcome (correct or incorrect) were recorded for each pair.

Table 1 summarizes some basic descriptive statistics of the dataset by item, where n denotes the number of respondents, N is the number of possible actions, $\bar{L}$ stands for the average process length, and Correct% is the percentage of correct responses. For items with partial credits, we treat the responses with full credits as correct and all other responses as incorrect. The 14 items

TABLE 1.
Descriptive statistics of 14 PIAAC problem-solving items.

ID	Description	n	N	$\bar{L}$	Correct%
U01a	Party Invitations—Can/Cannot Come	7620	207	24.8	54.5
U01b	Party Invitations—Accommodations	7670	249	52.9	49.3
U02	Meeting Rooms	7537	328	54.1	12.8
U03a	CD Tally	7613	280	13.7	37.9
U04a	Class Attendance	7617	986	44.3	11.9
U06a	Sprained Ankle—Site Evaluation Table	7622	47	10.8	26.4
U06b	Sprained Ankle—Reliable/Trustworthy Site	7612	98	16.0	52.3
U07	Digital Photography Book Purchase	7549	125	18.6	46.0
U11b	Locate E-mail—File 3 E-mails	7528	236	30.9	20.1
U16	Reply All	7531	257	96.9	57.0
U19a	Club Membership—Member ID	7556	373	26.9	69.4
U19b	Club Membership—Eligibility for Club President	7558	458	21.3	46.3
U21	Tickets	7606	252	23.4	38.2
U23	Lamp Return	7540	303	28.6	34.3

n, number of respondents; N, number of possible actions; $\bar{L}$, average process length; Correct%, percentage of correct responses.

vary in content, task complexity, and difficulty. Items U02 and U04a are the most difficult items as only around 10% of respondents had the correct answer. The tasks of these two items are also relatively complicated, requiring more than 40 actions on average and having a large number of possible actions. U06a is the simplest item in terms of task complexity since respondents took only 10.8 actions on average to finish the task and the item has the fewest possible actions. Despite the simplicity, less than 30% of respondents answered U06a correctly. The variety of items necessitates automatic methods to extract features from process data and to avoid identifying important actions and patterns manually, which is time-consuming and requires extra work if coding is changed.

4.2. *Feature Interpretation*

We extracted features for each of the 14 items by Procedure 1. The number of features is chosen from $\{10, 20, \ldots, 100\}$ by fivefold cross-validation, and the selected number for each item is given in the second column of Table 2.

Many of the principal features, especially the first several ones, have clear interpretations. We find the interpretation of a feature by examining the characteristics of the action sequences corresponding to the two extremes of the feature and then confirm it by calculating the correlation between the feature and a variable constructed according to the interpretation. Table 2 lists the interpretation of the first three principal features for each item.

The first principal feature of each item usually indicates attentiveness. An inattentive respondent often tries to skip a task directly or submits an answer by guessing randomly without meaningful interactions with the simulated environment, while an attentive respondent usually tries to understand and to complete the task by exploring the environment, thus taking more actions. Attentiveness in response process can be reflected in the process length. In Table 2, the numbers in the parentheses after the interpretation of the first principal feature of each item give the absolute value of the correlation between the first principal feature and the logarithm of the process length. For 13 out of 14 items, the absolute correlation is higher than 0.85. To explore the relation between the 14 first principal features, we multiply the features by the sign of their correlation with the corresponding process length. With the redirection, a higher first principal feature indicates a

TABLE 2.
Interpretation of first three principal features.

Item	K	Feature	Interpretation
U01a	50	1	Attentiveness in item response process (0.68)
		2	Intensity of mail and folder viewing actions
		3	Intensity of mail moving actions
U01b	30	1	Attentiveness in item response process (0.96)
		2	Intensity of creating new folders actions
		3	Intensity of mail moving actions
U02	50	1	Attentiveness in item response process (0.94)
		2	Intensity of mail moving actions
		3	Intensity of mail viewing actions
U03a	70	1	Attentiveness in item response process (0.86)
		2	Intensity of search and sort actions
		3	Times of answer submission
U04a	70	1	Attentiveness in item response process (0.98)
		2	Intensity of switching environments
		3	Intensity of arranging tables actions
U06a	60	1	Attentiveness in item response process (0.91)
		2	Intensity of clicking radio buttons
		3	Chance of classifying a website as useful
U06b	20	1	Attentiveness in item response process (0.94)
		2	Intensity of selecting answers
		3	Intensity of choosing website 2 against choosing website 4
U07	100	1	Attentiveness in item response process (0.96)
		2	Intensity of actions related to website 6
		3	Intensity of actions related to website 3
U11b	40	1	Attentiveness in item response process (0.94)
		2	Intensity of actions related to email in save folder
		3	Intensity of mail moving actions
U16	70	1	Attentiveness in item response process (0.95)
		2	Intensity of "Other_Keypress"
		3	Intensity of email viewing against email replying
U19a	40	1	Attentiveness in item response process (0.91)
		2	Intensity of typing emails
		3	Intensity of ticking and clicking email environment button
U19b	50	1	Attentiveness in item response process (0.89)
		2	Intensity of sorting actions
		3	Number of checked boxes
U21	50	1	Attentiveness in item response process (0.92)
		2	Intensity of making reservations
		3	Number of games selected
U23	40	1	Attentiveness in item response process (0.87)
		2	Click customer service against clicking not needed links
		3	Obtain Authorization number or not

Number in parentheses represents absolute value of correlation between first principal feature and logarithm of sequence length.

more attentive respondent. For a given pair of items, we calculate the correlation between their first principal features among the respondents who responded to both items. These correlations range from 0.36 to 0.74, implying that the respondents who tend to skip one item are likely to skip other items as well.

Some other features reveal whether the respondent understands the requirements of items. For example, item U11b requires respondents to classify emails in the "Save" folder. The second feature of U11b reflects if a respondent was working on the correct folder. Similarly, item U01b requires creating a new folder. The second feature of this item is related to whether this requirement is followed.

There are also features related to respondents' information and computer technology skills. Examples include the second feature of U03a, indicating whether search or sort tools are used, and the second feature of U04a, reflecting whether window split is used to avoid frequent switching between windows.

4.3. *Reconstruction of Derived Variables*

In this subsection, we demonstrate that the extracted features contain a substantial amount of information of the action sequences by showing that some key variables derived directly from the action sequences can be accurately reconstructed from the features.

Derived variables are binary variables indicating whether certain actions or patterns appear in the action sequences. For the example item described in the introduction, whether the first link is clicked is a derived variable. Item response outcomes (correct or incorrect) can also be treated as derived variables since they are entirely determined by the action sequences. In PIAAC data, besides the item response outcomes, 79 derived variables are recorded for the 14 items. These variables were derived during the item development process for each item to better track whether the test takers follow the pre-defined strategies. For instance, in the email-related environment (e.g., U01a), a binary variable is defined for each email to indicate whether the respondent opened the email. The following experiment examines how well the 93 (79 + 14) derived variables can be reconstructed from the features extracted from Procedure 1 by predicting the derived variables from the extracted features. A higher prediction accuracy indicates the variable can be reconstructed accurately.

For a given item, let Y denote a generic binary derived variable and $\boldsymbol{x}$ be a vector of principal features extracted from its response process. We consider the logistic regression model for each derived variable

$$\log\left(\frac{p}{1-p}\right) = \boldsymbol{\eta}^T\boldsymbol{\beta}, \tag{7}$$

where p is the probability of $Y = 1$ and $\boldsymbol{\eta}^T = (1, \boldsymbol{x}^T)$. For each derived variable, the respondents with the variable are randomly divided into a training set and test set in the ratio 4:1. The logistic regression model (7) is fit on the training set, and the value of derived variable in the test set is predicted as 1 if the fitted probability is greater than 0.5, and 0 otherwise. The prediction performance is evaluated by prediction accuracy.

Figure 7 presents a histogram of the prediction accuracy for the 93 derived variables. For most of the variables, the model constructed from the extracted features has more than 90% accuracy. This result confirms that the feature extracted by Procedure 1 is a comprehensive summary of the response processes.

Given that the features contain information about action sequences, a natural question is whether these features are useful for assessing respondents' competency and understanding their behavior. We will try to answer this question in the remainder of this section.

4.4. *Cross-Item Outcome Prediction*

In this section, we explore if the features obtained from the process data of one item are helpful to predict the outcomes of another item. Intuitively, if the extracted features characterize

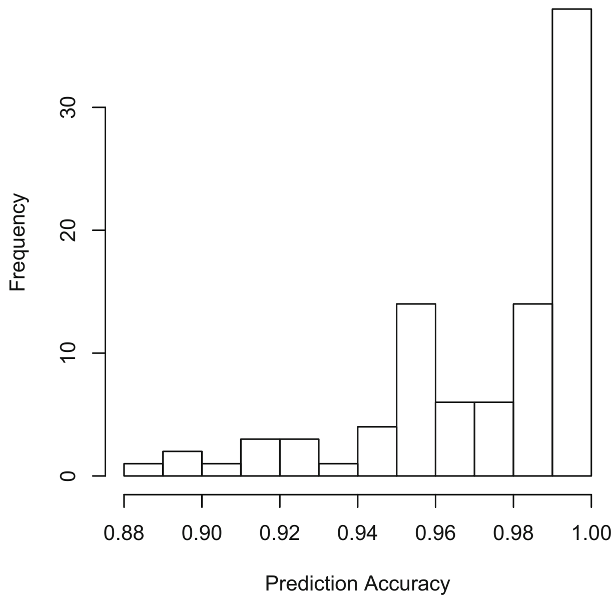

FIGURE 7.
Histogram of the prediction accuracy of derived variables.

the behavioral patterns and/or intellectual levels of respondents, which affect their performance in general, then these features should be able to tell more about whether the respondents can answer other items correctly than a single binary outcome.

Let Y_j denote the outcome of item j and $\boldsymbol{x}_j \in \mathbb{R}^{K_j}$ denote the features extracted from item j, $j = 1, \ldots, 14$. We model the relation between the outcome of item j and the outcome and the features of item $j' \neq j$ by a logistic regression

$$\log\left(\frac{p_j}{1-p_j}\right) = \boldsymbol{\eta}_{j'}^T \boldsymbol{\beta}, \tag{8}$$

where p_j is the probability of $Y_j = 1$ and $\boldsymbol{\eta}_{j'}$ is a vector of covariates of item j'. If process data are not taken into account, only $Y_{j'}$ provides information about Y_j and $\boldsymbol{\eta}_{j'}^T = (1, Y_{j'})$. In this case, available information for telling the outcome of item j is very limited, especially when the correct rate of item j' is close to 0 or 1. If process data are collected, then the features extracted according to Procedure 1 provide another source of information and we could use $\boldsymbol{\eta}_{j'}^T = (1, Y_{j'}, \boldsymbol{x}_{j'}^T, Y_{j'}\boldsymbol{x}_{j'}^T)$ as the covariates from item j'. Note that the interaction term $Y_{j'}\boldsymbol{x}_{j'}^T$ is included to increase the flexibility of the model. We call a model the baseline model if it only incorporates the outcome in $\boldsymbol{\eta}_{j'}$ and the process model if it utilizes the features extracted from process data.

Given that we want to model the outcome of item j based on the information provided in item j', respondents who responded to both items are randomly split into training, validation, and test sets in the ratio 4:1:1. Both the baseline model and the process model are fit on the training set. To avoid overfitting in the process model, L_2 penalties on the coefficients are incorporated. The process model is fitted on the training set for a grid of penalty parameters. The fitted process model that corresponds to the penalty parameter producing the highest prediction accuracy on the validation set is chosen to compare with the baseline model. The prediction accuracy of the process model for all combinations of j and j' is plotted against that of the corresponding baseline

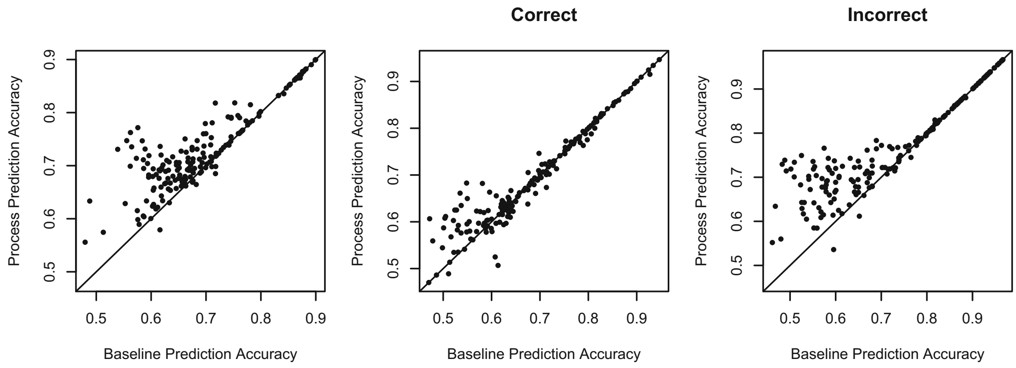

FIGURE 8.
Left: Prediction accuracy of the process model against the baseline model; middle: prediction accuracy of the process model against the baseline model for respondents who answered the predictor item correctly; right: prediction accuracy of the process model against the baseline model for respondents who answered the predictor item incorrectly.

model in the left panel of Figure 8. For most of the item pairs, the prediction accuracy is improved when the features extracted from process data are utilized, implying that the information in the process data is helpful in predicting the performance of respondents.

To take a closer look at the results, the middle and right panels of Figure 8 compare prediction accuracy separately for those who answered item j' correctly and incorrectly. The improvement in prediction accuracy is more obvious for the "incorrect" group. The main reason is that the action sequences corresponding to the incorrect responses usually provide more information about the respondents. There are usually more ways to answer a question incorrectly than correctly. An incorrect response may be the consequence of misunderstanding the item requirements or lack of basic computer skills. It may also result from the respondents' carelessness or inattentiveness. These varieties are reflected in the response processes, and thus, in the extracted features. As an illustration, the histograms of the first principal feature of item U01a stratified by the respondents' outcomes of U01a and U01b are plotted in Figure 9. In the U01a incorrect group, there is a significant difference in the feature distributions for those who answered U01b correctly and incorrectly, while the two distributions are almost identical in the U01a correct group. Recall that the first principal feature describes the respondents' attentiveness. Among the respondents who answer U01a incorrectly, those with lower feature values lack attentiveness. By including the features in the model, we are able to identify them and know that they are unlikely to answer U01b and other items correctly.

4.5. Score Prediction

The 14 interactive items in PIAAC were designed to study the PSTRE skills. The respondents' competency in literacy and numeracy were measured using items designed specifically for these two scales. We will show in this subsection that the process data from problem-solving items can cast light on respondents' proficiency in other scales. Let Z denote the score of a specific scale. We consider a linear model to explore the relation between Z and problem-solving items

$$Z = \boldsymbol{\eta}^T \boldsymbol{\beta} + \varepsilon, \tag{9}$$

where ε is a Gaussian random noise and $\boldsymbol{\eta}$ is a vector of predictors related to one or more problem-solving items and will be specified later.

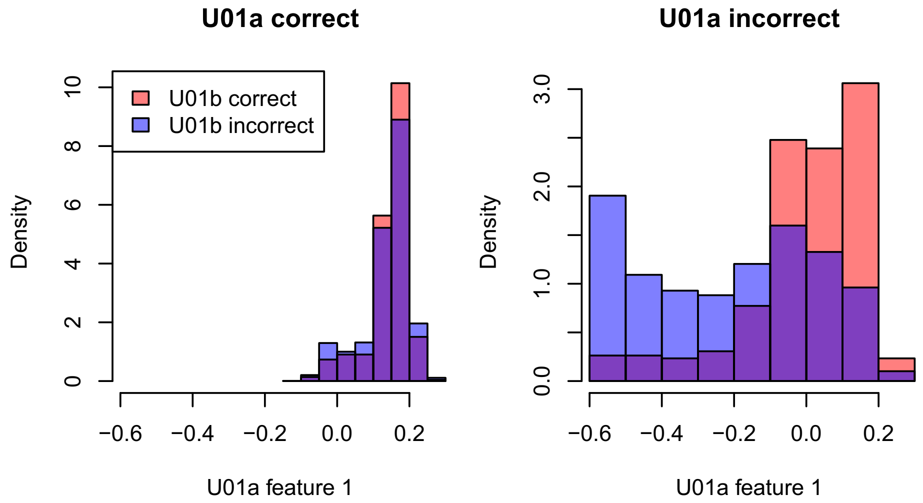

FIGURE 9.
Histograms of the first principal feature of U01a stratified by the outcomes of U01a and U01b.

4.5.1. Score Prediction Using a Single Item In the first experiment, we model the scores based on the information provided in a single item. In the model that only incorporates the binary outcome, namely the baseline model, the linear predictor is $\boldsymbol{\eta}^T = (1, Y_j)$. In the process model, we use $\boldsymbol{\eta}^T = (1, Y_j, \boldsymbol{x}_j^T, Y_j\boldsymbol{x}_j^T)$. For each of the 14 problem-solving items, the respondents are randomly split into training, validation, and test sets in the ratio 4:1:1. Both the baseline and the process model are fitted on the training set for literacy and numeracy scores separately. To avoid overfitting, L_2 penalties are placed on the coefficients in the process model for a grid of penalty parameters. The penalty parameter that produces the best prediction performance on the validation set is selected to obtain the final estimated process model. The prediction performance is evaluated by OSR^2.

The left panel of Figure 10 presents the OSR^2 of the baseline model and the process model for all combinations of score and item. For both literacy and numeracy scores, including information from process data is beneficial to score prediction. Although the problem-solving items are not designed to measure numeracy and literacy in PIAAC, process data can provide information leading to substantial improvements in these two scales.

The right panel of Figure 10 presents OSR^2 of the process model stratified by the outcome of an item. Similar to the outcome prediction in the previous subsection, the prediction performance for the respondents who answered an item incorrectly is usually much better than that for those who answered correctly since action sequences corresponding to incorrect answers often have more information than those corresponding to correct answers.

4.5.2. Score Prediction Using Multiple Items In the second experiment, we will examine how the improvement in score prediction brought by process data changes as the number of the items incorporated in the analysis increases. We only consider the 3645 respondents who responded to all 14 problem-solving items in this experiment. Among these respondents, 2645 are randomly assigned to the training set, 500 to the validation set and 500 to the test set. For each score, two models, a baseline model and a process model, are considered for a given set of items. For the baseline model, the linear predictor consists of the binary outcomes of the available items. For the process model, in addition to the binary outcomes, the linear predictor includes the first 20 principal features for each available item. Let $S_m = \{j_1, \ldots, j_m\}$ be a set containing the indices of

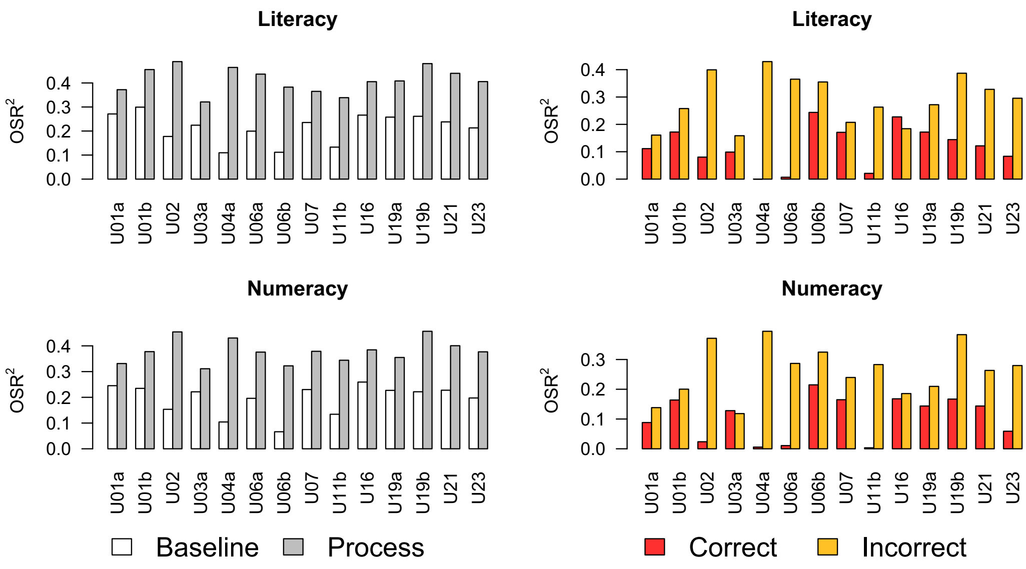

FIGURE 10.
Left: OSR^2 of the baseline and process model on the test set. Right: OSR^2 of the process model stratified by outcomes.

the items to be incorporated, where m denotes the number of indices in the set and ranges from 1 to 14 in our analysis. Then, the linear predictor for the baseline model is $\boldsymbol{\eta}^T = (1, Y_{j_1}, \ldots, Y_{j_m})$, while the linear predictor of the process model is $\boldsymbol{\eta}^T = (1, Y_{j_1}, \ldots, Y_{j_m}, \boldsymbol{x}_{j_1}, \ldots, \boldsymbol{x}_{j_m})$ where $\boldsymbol{x}_j \in \mathbb{R}^{20}$ is the first 20 principal features for item j. The set of available items is determined by forward Akaike information criterion (AIC) selection of the outcomes on the training set. Specifically, for a given m, S_m contains the items whose outcomes are the first m outcomes selected by the forward AIC selection among all 14 outcomes $Y_1, \ldots, Y_{14}$. For a given score, a sequence of baseline models and the process models are fitted on the training set. Similar to the previous subsection, L_2 penalty is added on the coefficients of the process models to avoid overfitting, and the penalty parameter is selected based on the OSR^2 on the validation set.

Figure 11 presents the OSR^2 of the baseline model and the selected process model on the test set. Regardless of the number of items available, the process model outperforms the baseline model in both literacy and numeracy score prediction. The improvement is more significant for literacy. The OSR^2 of the process model with only five items is comparable to the OSR^2 of the baseline model with all 14 items. In the process of completing the task in the problem-solving item, respondents need to comprehend the item description and provided materials, so the outcomes and the action sequences of problem-solving items can reflect respondents' literacy competency to some extent. Our experiment shows that process data can provide more information than binary outcomes. Properly incorporating process data in data analysis can exploit the information from items more efficiently.

5. Concluding Remarks

In this article, we present a method to extract informative latent variables from process data and illustrate the method via simulation studies and a case study of PIAAC 2012 data. The latent variables in the process data are extracted by an automatic procedure involving MDS of the dissimilarity matrix among response processes. The dissimilarity measure used in this article

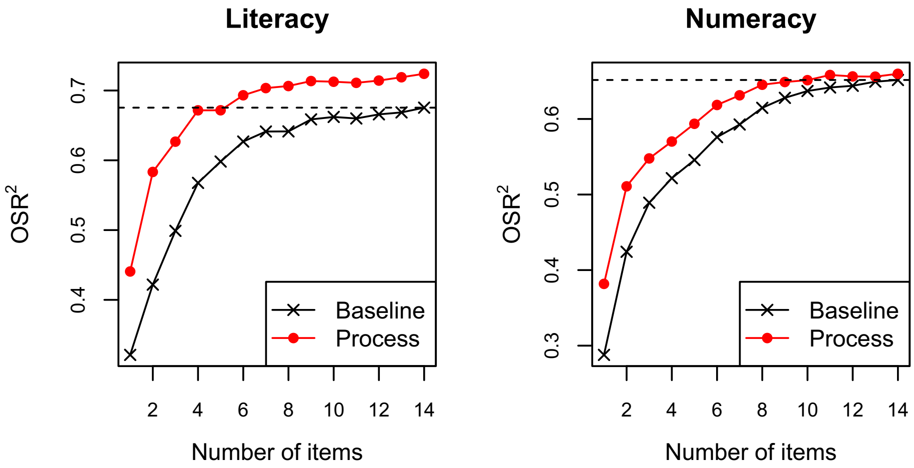

FIGURE 11.
OSR^2 of the baseline and process model with various number of items.

is just one of the possible choices. Other choices such as Levenshtein distance (Levenshtein, 1966) and optimal symbol alignment distance (Herranz, Nin, & Sole, 2011) can also be used, and similar observations can be made as shown in "Appendix." However, these measures are often more computationally demanding. The Levenshtein distance between two sequences can be computed by dynamic programming with time complexity $O(L_1 L_2)$, where L_1 and L_2 denote the lengths of the two sequences. The time complexity for calculating the dissimilarity measure used in this article is $O(L_1 + L_2)$. Another thing to be noted is that both the Levenshtein distance and the symbol alignment distance are genuine distance, whereas the dissimilarity used in this article is not since it does not satisfy the triangle inequality.

The proposed feature extraction method together with the prediction procedure in this article can be used to study the relationship between response processes and other variables of interest. For example, the respondents of our process data came from five different countries and they varied in age, gender, and many other demographic variables. We can study the difference among demographic groups in problem solving strategies. These results will help us identify the reasons for poor performance in different groups so that tailored suggestions can be provided for performance improvement. Another potential application is to provide career suggestions. If extracted features can accurately predict the level of various job-related skills, then we can make recommendations on suitable jobs by comparing the skill profile obtained from the response processes and the skill requirement for different jobs.

There are at least two directions along which the current method can be generalized to incorporate more information of response processes. In the current dissimilarity measure, two actions are either the same or not. Sometimes more delicate information about actions such as an $N \times N$ matrix of action similarity is available. How to incorporate such information in measuring the discrepancy between response processes needs further exploration. In this article, we only consider the action sequences in response processes. In many cases, time stamps of actions are also available in process data. The time elapsed between two consecutive actions may provide additional information about respondents and can be useful in cognitive assessments. Generalizing the current dissimilarity measure to incorporate response time or reaction time in the analysis of process data is another future direction.

The proposed method is implemented in `R` package `ProcData` available at http://scientifichpc.com/processdata/procdata.html. The code for producing the simulation study is included in the online supplementary materials.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Appendix

To compare the prediction performance of features extracted from different dissimilarity measures, we compute the Levenshtein distance matrix of the action sequences for each item and extracted features using Procedure 1 with the number of features K chosen by fivefold cross-validation. With these newly extracted features, we repeat the experiment of score prediction using multiple items (Section 4.5.2). All the settings are the same as before. A comparison of the prediction performance with the results in the main text is presented in Figure 12. Although the OSR^2 for the Levenshtein distance features is lower than that for the features extracted previously, it is still higher than that from the baseline model and the general trend of OSR^2 as the number of items increases is similar.

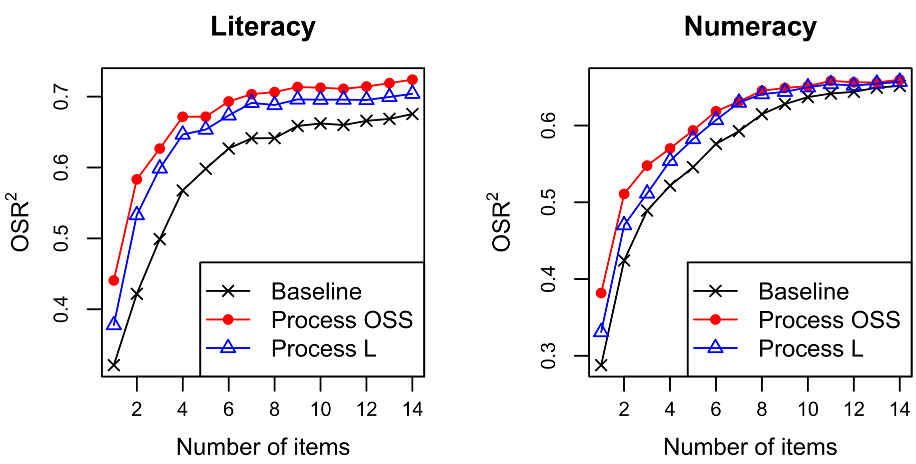

FIGURE 12.
Comparison of score prediction for features extracted based on different dissimilarity measures. "OSS" and "L" stand for the measure used in the main text and the Levenshtein distance, respectively.

References

Borg, I., & Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications*. New York, NY: Springer. https://doi.org/10.1007/0-387-28981-X.

Gómez-Alonso, C., & Valls, A. (2008). A similarity measure for sequences of categorical data based on the ordering of common elements. In V. Torra & Y. Narukawa (Eds.), *Modeling decisions for artificial intelligence* (pp. 134–145). Berlin: Springer. https://doi.org/10.1007/978-3-540-88269-5_13.

Greiff, S., Niepel, C., Scherer, R., & Martin, R. (2016). Understanding students' performance in a computer-based assessment of complex problem solving: An analysis of behavioral data from computer-generated log files. *Computers in Human Behavior*, *61*, 36–46. https://doi.org/10.1016/j.chb.2016.02.095.

He, Q., & von Davier, M. (2015). Identifying feature sequences from process data in problem-solving items with n-grams. In L. A. van der Ark, D. M. Bolt, W.-C. Wang, J. A. Douglas, & S.-M. Chow (Eds.), *Quantitative psychology research* (pp. 173–190). Cham: Springer. https://doi.org/10.1007/978-3-319-19977-1_13.

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. https://doi.org/10.4018/978-1-4666-9441-5.ch029.

Herranz, J., Nin, J., & Sole, M. (2011). Optimal symbol alignment distance: A new distance for sequences of symbols. *IEEE Transactions on Knowledge and Data Engineering*, *23*(10), 1541–1554. https://doi.org/10.1109/TKDE.2010.190.

Karni, E. S., & Levin, J. (1972). The use of smallest space analysis in studying scale structure: An application to the California psychological inventory. *Journal of Applied Psychology*, *56*(4), 341. https://doi.org/10.1037/h0032934.

Klein Entink, R., Fox, J.-P., & van der Linden, W. J. (2009). A multivariate multilevel approach to the modeling of accuracy and speed of test takers. *Psychometrika*, *74*(1), 21. https://doi.org/10.1007/s11336-008-9075-y.

Kroehne, U., & Goldhammer, F. (2018). How to conceptualize, represent, and analyze log data from technology-based assessments? A generic framework and an application to questionnaire items. *Behaviormetrika*, *45*(2), 527–563. https://doi.org/10.1007/s41237-018-0063-y.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, *10*(8), 707–710.

Lord, F. M. (1980). *Applications of item response theory to practical testing problems*. New York, NY: Routledge.

Meyer, E. M., & Reynolds, M. R. (2018). Scores in space: Multidimensional scaling of the WISC-V. *Journal of Psychoeducational Assessment*, *36*(6), 562–575. https://doi.org/10.1177/0734282917696935.

Qian, H., Staniewska, D., Reckase, M., & Woo, A. (2016). Using response time to detect item preknowledge in computer-based licensure examinations. *Educational Measurement: Issues and Practice*, *35*(1), 38–47. https://doi.org/10.1111/emip.12102.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, *22*(3), 400–407. https://doi.org/10.1214/aoms/1177729586.

Rupp, A. A., Templin, J., & Henson, R. A. (2010). *Diagnostic measurement: Theory, methods, and applications*. New York, NY: Guilford Press.

Shoben, E. J. (1983). Applications of multidimensional scaling in cognitive psychology. *Applied Psychological Measurement*, *7*(4), 473–490. https://doi.org/10.1177/014662168300700406.

Skager, R. W., Schultz, C. B., & Klein, S. P. (1966). The multidimensional scaling of a set of artistic drawings: Perceived structure and scale correlates. *Multivariate Behavioral Research*, *1*(4), 425–436. https://doi.org/10.1207/s15327906mbr0104_2.

Subkoviak, M. J. (1975). The use of multidimensional scaling in educational research. *Review of Educational Research*, *45*(3), 387–423. https://doi.org/10.3102/00346543045003387.

Takane, Y. (2006). 11 applications of multidimensional scaling in psychometrics. *Handbook of Statistics*, *26*, 359–400. https://doi.org/10.1016/S0169-7161(06)26011-5.

van der Linden, W. J. (2008). Using response times for item selection in adaptive testing. *Journal of Educational and Behavioral Statistics*, *33*(1), 5–20. https://doi.org/10.3102/1076998607302626.

Wang, S., Zhang, S., Douglas, J., & Culpepper, S. (2018). Using response times to assess learning progress: A joint model for responses and response times. *Measurement: Interdisciplinary Research and Perspectives*, *16*(1), 45–58. https://doi.org/10.1080/15366367.2018.1435105.

Zhan, P., Jiao, H., & Liao, D. (2018). Cognitive diagnosis modelling incorporating item response times. *British Journal of Mathematical and Statistical Psychology*, *71*(2), 262–286. https://doi.org/10.1111/bmsp.12114.

Manuscript Received: 11 APR 2019
Final Version Received: 13 MAY 2020