SESUG Paper 91-2023

Using PROC PSMATCH for Assembling Parallel Test Forms

Tsung-hsun Tsai, Research League, LLC; Yung-chen Hsu, Pearson, Inc,

ABSTRACT

In testing organizations, a pivotal undertaking within the test development cycle involves constructing multiple parallel forms, each comprising items with comparable characteristics and qualities. Drawing from the perspective of equivalence as akin joint distributions of test item traits, we leverage the matching capabilities offered by the PSMATCH procedure to establish links between item selections through matching, thereby enabling the construction of uniform parallel tests. The goal of assembling psychometrically equivalent tests transforms into the task of generating joint distributions of item characteristics resembling a known distribution. We elucidate the principle and essential concepts underlying this process, demonstrating how PROC PSMATCH automates the creation of parallel tests. Intended for an audience with a basic understanding of SAS/STAT® software, this work offers insights into streamlining test development procedures.

INTRODUCTION

A propensity score is the probability of treatment assignment being assigned to a particular group conditional on a set of observed baseline covariates. Propensity score analysis is commonly employed to mitigate potential biases in effect estimates derived from observational studies, allowing for comparison of outcomes between treated and control groups with the aim of attributing observed effects to the treatment itself. Proper implementation of propensity score adjustments entails several steps: (1) select covariates, (2) create propensity score/matching statistics, (3) select a matching algorithm, (4) generate matches, (5) diagnose matches, and (6) evaluate the effect of the treatment.

This process offers researchers a plethora of options in terms of estimation and conditioning methods. PROC PSMATCH streamlines the calculation of propensity scores or matching statistics, incorporating various propensity score matching (PSM) algorithms. These algorithms empower practitioners to mitigate systematic biases stemming from self-selection by matching treated individuals to controls, thereby ameliorating imbalances in pre-treatment covariate distributions. PROC PSMATCH facilitates both numerical and graphical assessments of variable balance between treated and control groups. It offers the flexibility to apply propensity score weighting, stratification, or matching methods as needed. Furthermore, researchers can leverage the propensity score data output from PROC PSMATCH for subsequent outcome analyses.

Traditional test assembly methods typically rely on optimization strategies grounded in mathematical programming approaches, focusing on point-wise matching of test information by minimizing fitting errors, as indicated by differences between the test information functions of constructed test forms and a reference form. To overcome scalability and feasibility challenges inherent in traditional mathematical programming approaches in constructing parallel test forms, we employ PSM techniques provided by PROC PSMATCH. Rather than pursuing a direct optimization path, we adopt a perspective that regards equivalence as akin to similar joint distributions of item characteristics. Leveraging PSM techniques, we perform item-by-item matching to select sets of items that exhibit psychometric equivalence.

AN OVERVIEW OF AUTOMATED TEST ASSEMBLY

Current test assembly practices primarily rely on item response theory (IRT), a modern psychometric test theory that assumes a mathematical model for the probability that a test-taker will respond correctly to a specific test item, given the test-taker’s ability and item characteristics of the test item. During scaling stage of test development cycle, IRT parameters of the models are estimated. There are various IRT models and estimation methods. The procedure for IRT (PROC IRT) in SAS/STAT® provides useful features for conducting various IRT analyses of dichotomous and polytomous datasets that are unidimensional or multidimensional. For instance, the two-parameter logistic model (2PL) is a unidimensional IRT model for dichotomous responses used in various large-scale testing programs, such as National Assessment of Educational Progress, a congressionally mandated program administered by the National Center for Education Statistics, within the U.S. Department of Education. The model can be expressed as

$$P_i(\theta, a_i, b_i) = \frac{1}{1 + \exp[-Da_i(\theta - b_i)]},$$

where P represents the probability of correctly answering a particular dichotomously scored item given the proficiency level θ. The parameters a_i and b_i are characteristics of item i, and D is the commonly used scaling constant of 1.7. Items that are functioning poorly on a scale are identified early during pilot tests and dropped from the scale. The collection of suitable assessment items are organized and catalogued as an item bank to serve as a repository of all test content. The idea is that the items can be selected as required to make up a particular test.

Constructing multiple parallel test forms is a crucial task for most testing organizations. Parallel test forms are necessary when tests are administered at different times or in specific situations, such as pretest-posttest designs for program evaluation. The goal is to select items to construct multiple forms, ensuring that each form comprises items with approximately equivalent psychometric characteristics and qualities, yet consists of a different set of items. This ensures that test-takers using distinct test forms can be evaluated objectively on the same scale.

ILLUSTRATIVE EXAMPLE

We utilize simulated statistics to elucidate the matching approach. For descriptive purposes, we opt for the 2PL model in the example, facilitating a clearer geometrical interpretation within a two-dimensional graph. Drawing from log-normal distributions and normal distributions with varying means, variances, and upper and lower bounds, we generate a set of 1000 a- and b-parameters, respectively, to simulate an item pool and a 50-item reference test. The objective is to assemble three parallel test forms.

The input parameters for the matching program are:

- `ID` = item ID
- `a` = a-parameter of 2PL
- `b` = b-parameter of 2PL
- `ref`= Binary indicator denoting whether the item is a reference item (Y for reference item)
- `lga`= Logarithm of the a-parameter

The input data set represents processed data queried from an item bank, encapsulating item characteristics. The PROC PSMATCH matching code and the following steps to separate reference test form items and parallel tests item and are as follows:

```
* PS match;
proc psmatch data=itempool region=allobs;
   class ref;
   psmodel ref(Treated='Y')=loga b;
   match method=optimal(k=3)
         stat=mah(var=(loga b) /cov=identity );
   assess ps var=(loga b) /plots=all weight=none;
   output out(obs=match)=psOut matchid=_MatchID;
run;

* Split reference test and parallel tests items;
data pllTestSet refTestSet;
  set psOut (drop=_PS_ _MATCHWGT_);
  if ref='N' then output pllTestSet;
  else output refTestSet;
```

```
run;
proc sort data=refTestSet out=refTestSet;
  by _MatchID;
run;
proc sort data=pllTestSet out=pllTestSet;
  by _MatchID;
run;
```

MATCHING PROCESS

CLASS Statement

The CLASS statement specifies the classification variables, which must precede the PSMODEL statement.

PSMODEL Statement

In the PSMODEL statement, the `ref` variable serves as the binary treatment indicator, denoting whether an item is a reference item, with Treated='Y' identifing the treated group. In this example, the PSMODEL statement includes two predictors, `loga` and `b`. The MATCH statement specifies the criteria for matching.

MATCH Statement

Various matching strategies, which differ in the selection of matching algorithms, influence both the quantity and quality of matches. Ultimately, the choice rests on the researcher's discretion (Jacovidis, 2017). PROC PSMATCH offers three distinct matching strategies:

- **Greedy nearest neighbor matching:** This method sequentially selects control units without replacement, choosing those whose propensity scores are closest to the propensity score of each treated unit.
- **Replacement Matching:** Control units are selected with replacement, opting for those with propensity scores closest to those of the treated units.
- **Optimal matching:** This strategy simultaneously selects all matches without replacement, aiming to minimize the total absolute difference in propensity scores across all matches. It encompasses the following methods:
  - Fixed Ratio Matching: Matches a predetermined number of control units to each treated unit.
  - Variable Ratio Matching: Matches one or more control units to each treated unit.
  - Full Matching: Matches each treated unit to one or more control units and vice versa.
- **Matching with replacement:** Control units are selected with replacement, opting for those with propensity scores closest to those of the treated units.

For most typical test assembly tasks, optimal matching with a fixed ratio is the preferred method. Since we are constructing three parallel test forms, the METHOD=OPTIMAL(K=3) option requests the optimal matching of one control unit to three units in the treated group.

The PSMATCH procedure provides the following types of statistics for matching observations in the treated group with those in the control group:

- the absolute difference in the logit of the propensity score
- the Mahalanobis distance between sets of continuous variables
- the absolute difference in the propensity score

To better elucidate the concept, we define the distance measure for a pair of items i and j as:

$$d(i,j)=\sqrt{(b_i-b_j)^2+(\log(a_i)-\log(a_j))^2}$$

where a and b are the item parameters of an item defined in 2PL. The distance measure is based on Euclidean distance, a special case of Mahalanobis distance. The metric symbolizes the proximity of two items within the item bank and facilitating a geometric interpretation. The STAT=MAH option requests the use of Mahalanobis distance as the matching criterion for comparing pairs of observations. The COV=IDENTITY suboption specifies the use of the identity matrix to calculate distance.

ASSESS Statement

There are several methods to measure the degree of matching between the treated and control groups. These include standardized mean differences, which compare the means of covariates, and the Kolmogorov-Smirnov test, which measures distributional difference. Additionally, certain machine learning approaches can be applied. Graphical methods, such as histograms and box plots, allow visual inspection of the balance of covariates between the treated and control groups, both before and after matching.

The PSMATCH procedure offers multiple methods to evaluate the balance of variable distributions between the treated and control groups. The ASSESS statement generates numeric measures and plots to diagnose differences between these groups. These metrics and visualizations help users iteratively refine the propensity score model, using different methods and variables, until they achieve a satisfactory fit (Bergstra et al., 2019).

OUTPUT Statement

The OUTPUT statement produces a dataset. The OUT(OBS=MATCH)=psOut creates an output data set, psOUT, containing only the matched data. The option MATCHID=_MatchID provides identification numbers for the matched treated and control units.

DIAGNOSTIC ASSESSMENT

Figures 1 to 3 illustrate some examples of standardized differences plots, box plots, and cloud plots.

While there is no universally accepted threshold for the standardized difference to indicate covariate imbalance, a value below 10% suggests negligible mean or prevalence differences between treatment groups (Normand et al., 2001). Figure 1 shows the results before and after matching adjustment, indicating no statistically significant difference between the reference test and matched parallel tests item sets.

Figure 2 presents boxplots of the propensity scores, revealing substantial overlap between the reference items and matched items. The boxplots depict the interquartile range and whiskers extending to the highest and lowest values.

Figure 3 displays cloud plots depicting the distribution of propensity scores for items. These plots differentiate between the reference test form (treated group) and item pool (control groups) across three categories: all items, items within the common support region, and matched items. Points are jittered vertically to prevent overlap. Green dots represent items successfully matched to the reference test based on similar propensity scores, while unmatched items (red circles) tend to concentrate in the tails of the distribution, indicating a scarcity of suitable matches in these regions.

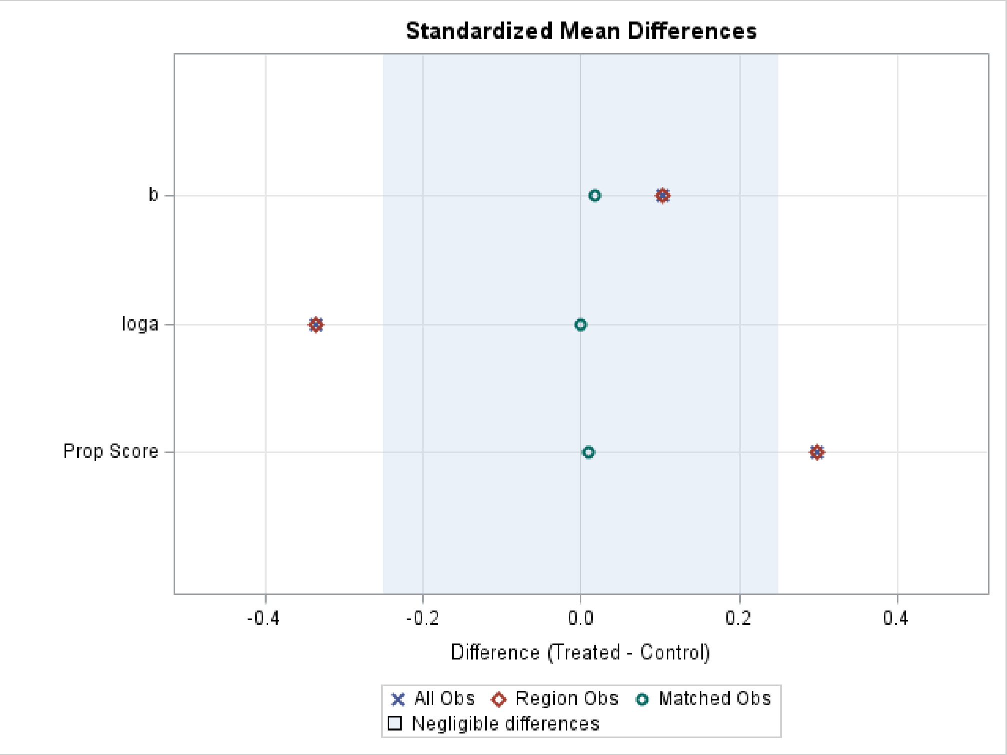

Figure 1. Standardized Differences Plot

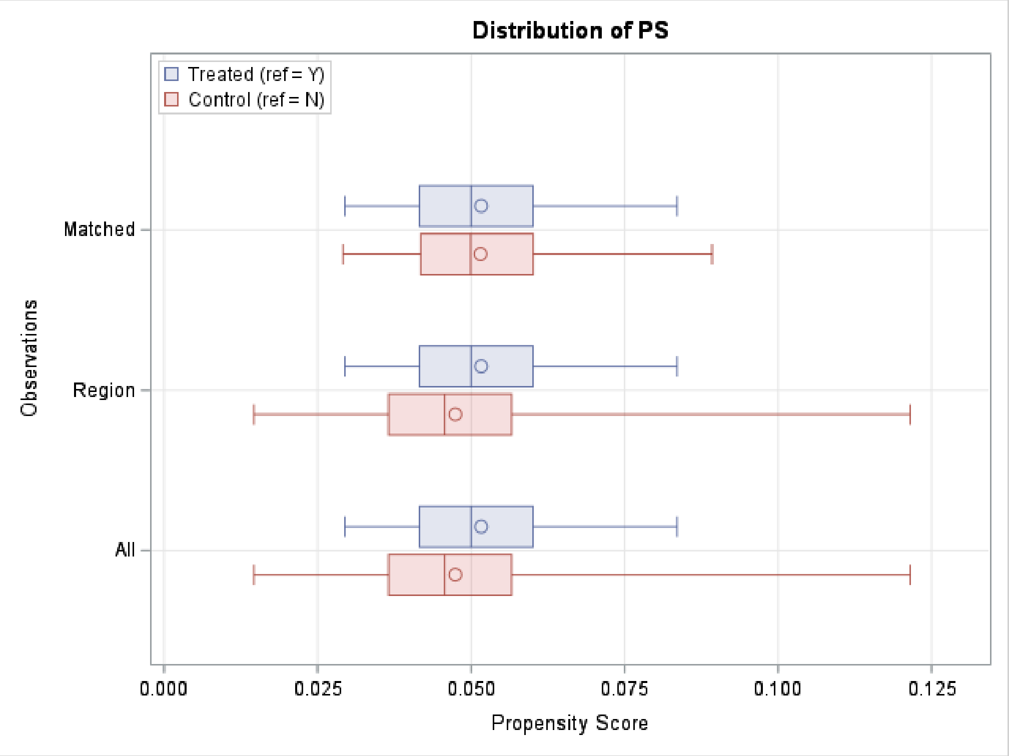

Figure 2. Box Plots

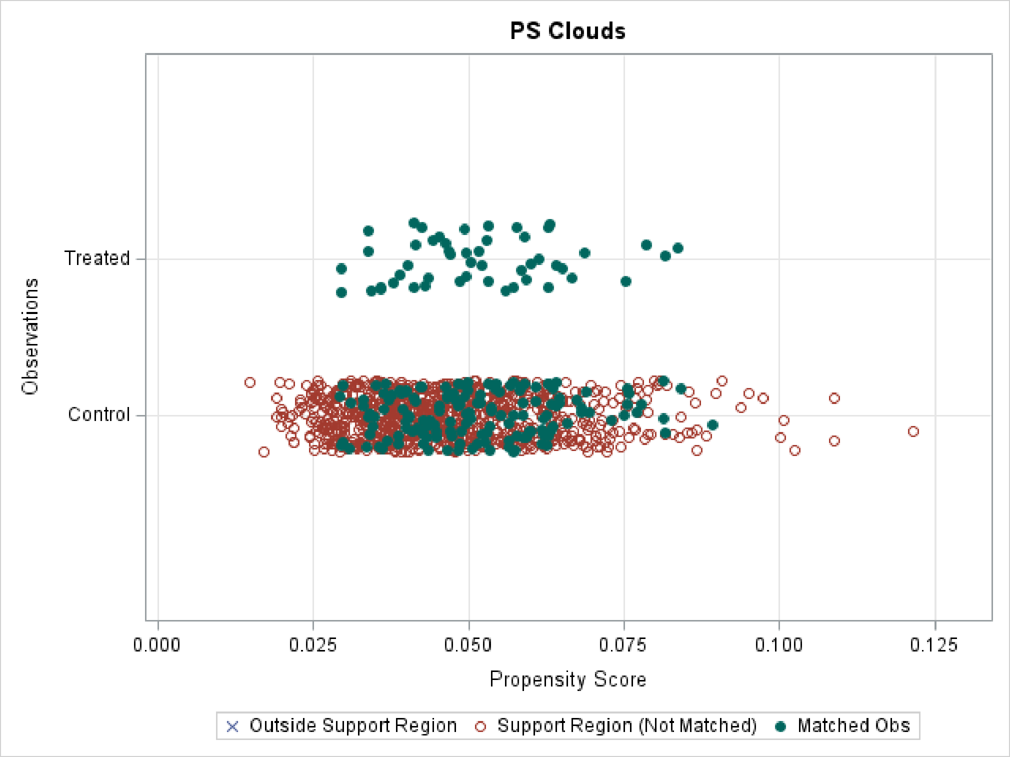

Figure 3. Cloud Plots

TEST FORM ASSIGNMENT

The code to assign test form numbers is shown below:

```
* Assign test form numbers;
data refTestSet;  set refTestSet;  testNum=0;  run;
data pllTestSet;
  set pllTestSet;
  by _MatchID;
  testNum+1;
  if first._MatchID then testNum=1;
run;
```

To assign test form numbers, we leverage the inherent order within the matched ID groups generated by the PSMATCH procedure. While further randomization could be implemented, we posit that the existing sequence provides sufficient randomness for practical purposes. For enhanced control, Fisher-Yates shuffle (Dustenfeld, 1964), a special case of reservoir sampling (Vitter, 1985), could be employed. This would adjust the selection probabilities when choosing a parallel test form to assign an item from the remaining unmatched items.

IMPLICATION OF EQUIVALENCY

The mathematical programming approach interprets the *equivalency* of parallel test forms as having *similar test information curves*. The item information of item i pertaining to the proficiency level θ, which is derived from Fisher information, is given by the formula (Yen & Fitzpatrick, 2006):

$$I_i(\theta) = \frac{[P_i^{'}(\theta)]^2}{P_i(\theta)(1-P_i(\theta))} \text{, where } P_i'(\theta) = \frac{\partial P_i(\theta)}{\partial \theta}.$$

A test is a combination of items. Hence, the test information of a test with n items is simply the sum of the item information of all the items:

$$I(\theta) = \sum_{i=1}^{n} I_i(\theta).$$

Figure 4 provides additional assessment by comparing the test information cures of the reference test form and the three parallel test forms, showing the relative information values at different ability levels. The reference test is a classification test with two cut-off points, which is why the curve has two peaks. The code to produce the graph is provided in Appendix A.

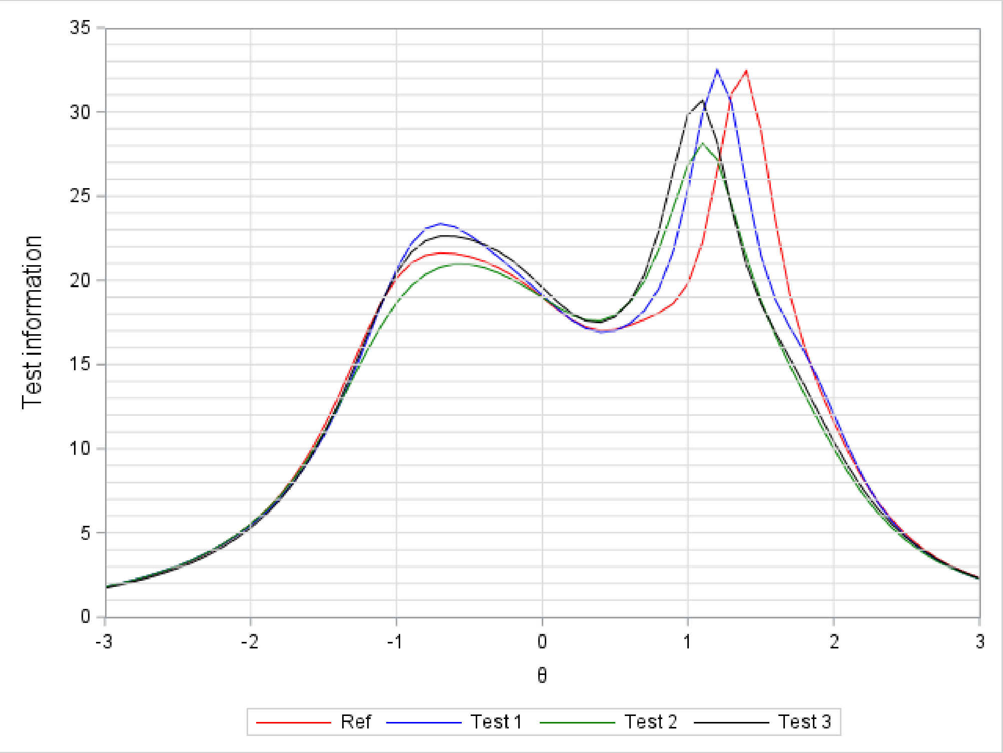

Figure 4. Test Information Curves

Chen’s (2005) view of equivalency is similar joint distributions of the item characteristics, aiming to select items to form a distribution that is approximately equivalent in a multidimensional space. Figure 5 visually depicts this by clustering reference items (red dots) with their matched counterparts (green dots). The code used to generate this plot is provided in Appendix B. However, to quantify the level of equivalence achieved, we still require appropriate statistical measures or similarity metrics.

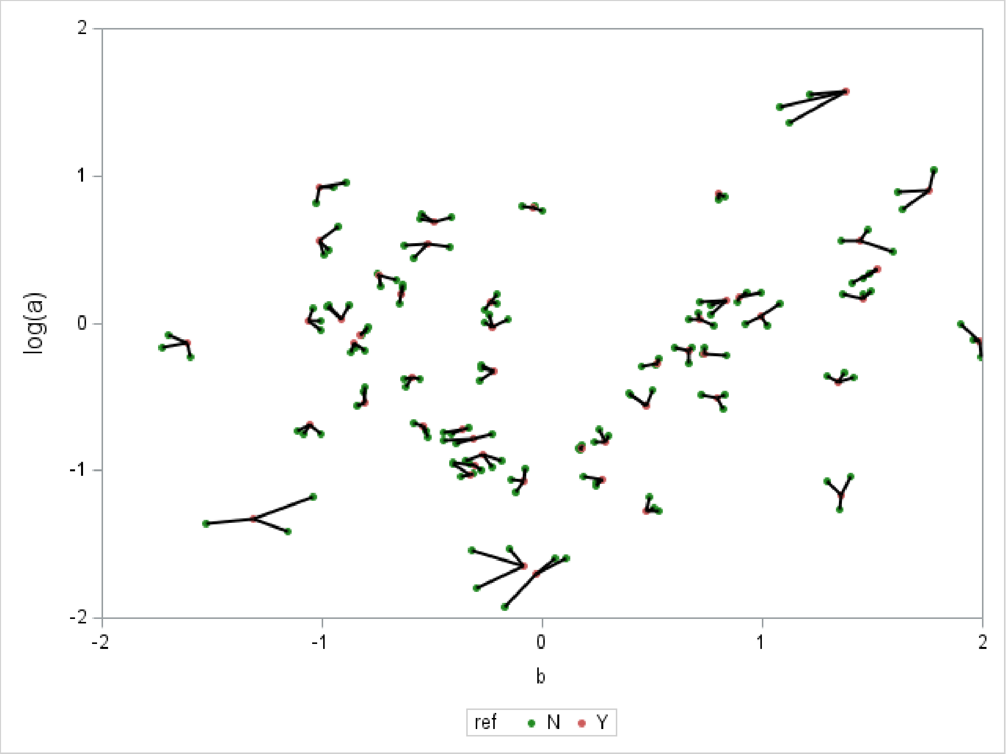

Figure 5. Scatter plot of reference items and matched items.

PRAGMATIC PERSPECTIVES AND OBJECTIVES

For test information curves, an important concern is the closeness of these curves between two forms. Researchers commonly use the mean squared deviation (MSD) as a measure of similarity between two test information curves. In the mathematical programming model, the objective is to *minimize MSD*.

However, testing service organizations prioritize other practical aspects beyond just the closeness of two test information curves. First, while the overall shape of the information curve remains important, point-to-point matching of test information is less essential in practical settings. Second, having test forms with similar information curves does not necessarily imply that the test items share similar characteristics. Although adding more constraints could improve the model fit, this would significantly exacerbate infeasibility and inefficiency issues.

In Chen's (2005) proposal the selection is balanced through randomization to produce more uniform forms, which approximately guarantees form quality. This can be achieved easily by PSMATCH procedure. Later Chen (2012) introduced several preliminary stratification matching strategies. The general idea is to divide the pool, viewed as a multidimensional space, into smaller cells, each containing a subset of items or points. Each item in the reference form is mapped to a corresponding cell. To create a parallel test form, each reference item is matched from its designated cell. The selection is balanced through randomization to ensure a more uniform quality without explicitly specifying an objective function.

Chen (2014) explicitly discussed the objective of *minimizing the variation of mean distances* among test forms, with a mean distance representing the average of accumulated distances between reference items and their corresponding selections. Realizing this goal might necessitate resturing the program block for assigning test form numbers. Additionally, other pragmatic objectives, like *minimizing the variation of standard deviations* of distances among test forms, could be considered. While these objectives are not necessarily conflicting, they often compete and are interrelated.

CONCLUSION

Assembling multiple parallel test forms is a crucial task in the test development cycle for most testing organizations. This involves selecting items to build test forms with approximately equivalent psychometric characteristics and qualities. While a combinatorial optimization method may be used during the initial form construction, it is often not employed in the later phases of assembling or revising parallel forms. This task is typically intensive, time-consuming, and costly.

In many real-world scenarios, effectively performing the test assembly task is a practical issue rather than an aesthetic one. Testing organizations frequently face decision-making challenges with conflicting goals. It is essential to recognize that no single method is universally appropriate for all situations; the choice of methods or objectives depends on the specific context. A satisfactory outcome is one that complies with policy and meets immediate objectives through an admissible decision-making strategy. Given the tight schedules for building parallel forms, finding viable strategies to streamline this process is a common necessity.

We explore tangible solutions for automating parts of the parallel test assembly process using the PSMATCH procedure to address practical issues. Implementing the matching process with PROC PSMATCH is relatively straightforward, and the results demonstrate its feasibility and efficiency in an operational environment.

REFERENCES

Bergstra SA, Sepriano A, Ramiro S, Landewél R. (2019). Three handy tips and a practical guide to improve your propensity score models. *RMD Open, 5* (1). https://doi.org/10.1136/rmdopen-2019-000953

Chen, P.-H. (2005). *IRT-based automated test assembly: A sampling and stratification perspective* (Doctoral Dissertation). University of Texas, Austin, TX.

Chen, P.-H. , Chang, H.-H., & Wu, H. (2012). Item selection for the development of parallel forms from an IRT-based seed test using a sampling and classification approach. *Educational and Psychological Measurement*, *72* (6), 933–953.

Chen, P.-H. (2014, April). *Three-Element Item Selection Procedures for Multiple Forms Assembly: An Item Matching Approach.* Paper presented at the 2014 Annual Meeting of the American Educational Research Association, Philadelphia, PA.

Durstenfeld, R. (1964). Algorithm 235: Random permutation. *Communications of the ACM , 7* (7), 420.

Jacovidis. J. N. (2017). Evaluating the performance of propensity score matching methods: A simulation study (Doctoral Dissertation). James Madison University, Harrisonberg, VA.

Normand, S. L. T., Landrum, M. B., Guadagnoli, E., Ayanian, J. Z., Ryan, T. J., Cleary, P. D., & McNeil, B. J. (2001) Validating recommendations for coronary angiography following an acute myocardial infarction in the elderly: A matched analysis using propensity scores. *Journal of Clinical Epidemiology, 54*, 387–398.

Vitter, J. S. (1985). Random Sampling with a Reservoir. *ACM Transactions on Mathematical Software, 11* (1), 37–57.

Yen, W. M., & Fitzpatrick, A. R. (2006). Item Response Theory. InR. L. Brennan (Eds.), Educational Measurement (4th ed., pp. 126-127). Praeger.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Tsung-hsun Tsai
Research League, LLC
ttsai@researchleague.org

APPENDIX A

```
* 61-point item information;
%macro i61pt;
%let D=1.7;
%let D2=%sysevalf(&D*&D);
  array ri{61} r1 - r61;
  a2=a*a;
  do i=1 to 61;
    P=1/(1+exp(-&D*a*(((i-31)/10.0)-b)));
   Q=1-P;
    ri{i}=&D2*a2*p*q;
  end;
  drop a2 a b P Q i;
%mend;

* 61-point test information;
data testSet;
  set refTestSet pllTestSet;
run;

data itemInf;
  set testSet(keep=a b testNum);
  %i61pt;
run;

proc means data=itemInf noprint;
  class testNum;
  var r1-r61;
  output out=testInf sum(r1-r61)=sr1-sr61;
run;

data testInf;
  set testInf (firstobs=2);
  drop _TYPE_ _FREQ_;
run;

proc transpose data=testInf out=testInfPlot prefix=t; run;

data testInfPlot;
  set testInfPlot (drop=_name_ firstobs=2);
  rename t1=ref t2=test1 t3=test2 t4=test3;
  theta=(_n_-31)/10.0;
run;

* Test information curves;
proc sgplot data=testInfPlot;
  series x=theta y=ref
         / lineattrs=(color=red) name='ref' legendlabel='Ref';
  series x=theta y=test1
         / lineattrs=(color=blue) name='test1' legendlabel='Test 1';
  series x=theta y=test2
         / lineattrs=(color=green) name='test2' legendlabel='Test 2';
  series x=theta y=test3
         / lineattrs=(color=black) name='test3' legendlabel='Test 3';
  xaxis label="(*ESC*){unicode theta}" values=(-3 to 3 by 1)
        offsetmin=0 offsetmax=0;
  yaxis label="Test information" labelattrs=(size=12) values=(0 to 35 by 5)
        offsetmin=0 offsetmax=0;
  refline -3 to 3 / axis=x lineattrs=(color=graydd);
  refline 0 to 35 / axis=y lineattrs=(color=graydd);
run;
```

APPENDIX B

```
* Annotation;
data refPoint;
  set refTestSet(drop=a ref id testNum);
  rename _MatchID=node0 b=b0 loga=loga0;
run;

data pllPoint;
  set pllTestSet(drop=a ref id testNum);
  rename _MatchID=node1 b=b1 loga=loga1;
run;

proc sql;
  create table anno as
  select * from refPoint, pllPoint where node0=node1;
quit;

data anno;
  set anno;
  retain function 'line' drawspace 'datavalue' linecolor 'black';
  x1=b0;  y1=loga0;  x2=b1;  y2=loga1;
run;

* Cluster plot;
proc sgplot data=psOut sganno=anno;
  scatter x=b y=loga / group=ref markerattrs=(size=5 symbol=CircleFilled);
  styleattrs
    datacontrastcolors=(ForestGreen IndianRed);
  xaxis label="b" values=(-2 to 2 by 1) offsetmin=0 offsetmax=0;
  yaxis label="log(a)" labelattrs=(size=12) values=(-2 to 2 by 1)
        offsetmin=0 offsetmax=0;
run;
```