Supplementary Materials

Psychometric and machine learning approaches to reduce the length of scales

Item stems for all items in the analysis

SSRQ01. I don't notice the effects of my actions until it's too late.
SSRQ02. I put off making decisions.
SSRQ03. It's hard for me to notice when I've 'had enough' (alcohol, food, sweets).
SSRQ04. I have trouble following through with things once I've made up my mind to do something.
SSRQ05. I don't seem to learn from my mistakes.
SSRQ06. I usually only have to make a mistake one time in order to learn from it.
SSRQ07. I can usually find several different possibilities when I want to change something.
SSRQ08. Often I don't notice what I'm doing until someone calls it to my attention.
SSRQ09. I usually think before I act.
SSRQ10. I learn from my mistakes.
SSRQ11. I give up quickly.
SSRQ12. I usually keep track of my progress toward my goals.
SSRQ13. I am able to accomplish goals for myself.
SSRQ14. I have personal standards, and try to live up to them.
SSRQ15. As soon as I see a problem or challenge, I start looking for possible solutions.
SSRQ16. I have a hard time setting goals for myself.
SSRQ17. When I'm trying to change something, I pay a lot of attention to how I'm doing.
SSRQ18. I have trouble making plans to help me reach my goals.
SSRQ19. I set goals for myself and keep track of my progress.
SSRQ20. If I make a resolution to change something, I pay a lot of attention to how I'm doing.
SSRQ21. I know how I want to be.
SSRQ22. I have trouble making up my mind about things.
SSRQ23. When it comes to deciding about a change, I feel overwhelmed by the choices.
SSRQ24. I tend to keep doing the same things, even when it doesn’t work.
SSRQ25. Once I have a goal, I can usually plan how to reach it.
SSRQ26. If I wanted to change, I am confident that I could do it.
SSRQ27. I can stick to a plan that's working well.
SSRQ28. I have a lot of willpower.
SSRQ29. I am able to resist temptation.

	ssrq1	ssrq2	ssrq3	ssrq4	ssrq5	ssrq6	ssrq7	ssrq8	ssrq9	ssrq10	ssrq11	ssrq12	ssrq13	ssrq14	ssrq15	ssrq16	ssrq17	ssrq18	ssrq19	ssrq20	ssrq21	ssrq22	ssrq23	ssrq24	ssrq25	ssrq26	ssrq27	ssrq28	ssrq29
ssrq1	NA		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ssrq2	1	NA		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
ssrq3	1	1	NA		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ssrq4	1	1	1	NA		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ssrq5	1	1	1	1	NA		1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ssrq6	1	1	1	1	1	NA		1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ssrq7	1	1	1	1	1	1	NA		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ssrq8	1	1	1	1	1	1	1	NA		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ssrq9	1	1	1	1	1	1	1	1	NA		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ssrq10	1	1	1	1	0	0	1	1	1	NA		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ssrq11	1	1	1	1	1	1	1	1	1	1	NA		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ssrq12	1	1	1	1	1	1	1	1	1	1	1	NA		1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
ssrq13	1	1	1	1	1	1	1	1	1	1	1	1	NA		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ssrq14	1	1	1	1	1	1	1	1	1	1	1	1	1	NA		1	1	1	1	1	1	1	1	1	1	1	1	1	1
ssrq15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA		1	1	1	1	1	1	1	1	1	1	1	1	1
ssrq16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA		1	0	1	1	1	1	1	1	1	1	1	1
ssrq17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA		1	0	1	1	1	1	1	1	1	1	1
ssrq18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	NA		1	1	1	1	1	1	1	1	1	1
ssrq19	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	NA		1	1	1	1	1	0	1	1	1
ssrq20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	NA		1	1	1	1	1	1	1	1
ssrq21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA		1	1	1	1	1	1	1
ssrq22	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA	0	1	1	1	1	1	1
ssrq23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	NA		1	1	1	1	1
ssrq24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA		1	1	1	1
ssrq25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	NA		1	1	1
ssrq26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA		1	1
ssrq27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA		1
ssrq28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA	0
ssrq29	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	NA

Figure A. Items flagged by the Jackknife Slope Index (JSI) as exhibiting local dependence. 0=flagged, 1=not flagged. JSI bounds were -1.357 and 1.334.

##LD resolution – Items highlighted were included for analyses

Considerations: double-barrel items, conditionals, long prompts, or the number of doublets that the item was involved in

```
[1,]  22   2
[2,]  23   2
[24,]  23  22
# SSRQ02    I put off making decisions.
# SSRQ22    I have trouble making up my mind about things.
# SSRQ23 When it comes to deciding about a change, I feel
 overwhelmed by the choices.

 [3,]   6   5
 [4,]  10   5
 [5,]   5   6
# SSRQ05    I don't seem to learn from my mistakes.
# SSRQ06    I usually only have to make a mistake one time in
 order to learn from it.
# SSRQ10   I learn from my mistakes.

[9,]  19  12
[10,]  19  25
[11,]  25  19

# SSRQ12   I usually keep track of my progress toward my goals.
# SSRQ19   I set goals for myself and keep track of my progress.
# SSRQ25   Once I have a goal, I can usually plan how to reach
 it.

[14,]  20  17
# SSRQ20 If I make a resolution to change something, I pay a lot
 of attention to how I'm doing.
# SSRQ17   When I'm trying to change something, I pay a lot of
 attention to how I'm doing.

[12,]  18  16
[13,]  18  16
# SSRQ18   I have trouble making plans to help me reach my goals.
# SSRQ16   I have a hard time setting goals for myself.

[29,]  29  28
# SSRQ28   I have a lot of willpower.
# SSRQ29   I am able to resist temptation.
```

Table A. *Correlations between estimated scores of the SSRQ, summed scores of the SSRQ, and outcomes in the testing dataset.*

	EAP[θ]	Summed Score	IRT SF EAP[θ]	CAT EAP[θ]	GA score	Tree score
EAP[θ]	1.000					
Summed Score	.972	1.000				
IRT SF EAP[θ]	.967	.949	1.000			
CAT EAP[θ]	.980	.937	.958	1.000		
GA score	.951	.978	.916	.926	1.000	
Tree score	.896	.905	.907	.899	.874	1.000
Nervousness	-.209	-.203	-.220	-.253	-.198	-.270
Hopelessness	-.346	-.354	-.344	-.389	-.356	-.390
Alcohol Frequency	-.067	-.028	-.071	-.049	-.040	-.042
Coffee per Day	.068	.091	.056	.068	.053	.184
BSCS	.746	.763	.779	.726	.728	.699
Grit-S	.769	.759	.792	.784	.734	.756

Note: [a] is for correlations significantly different from the summed score after a Bonferroni correction (p=.002), and in this case are none of them. EAP[θ]=expected a posteriori θ estimate; IRT SF=item response theory short-form; CAT= Computerized Adaptive Testing (maximize Fisher's information); GA=genetic algorithm; BSCS=Brief Self-control Scale; Grit-S=Short Grit Scale

Item response theory short-forms (selecting items with maximum information, no hand-selecting the two items that represent the construct).

In the reduced set of 20 SSRQ items, if the IRT approach were to use only items that maximize information and not include items to define the construct, the correlation between the two sets of scores would be *r*=.978. The relations were largely maintained. The items (different from manuscript are bolded), correlations with other scores, and correlations with outcomes are shown below. There is not a clear pattern in the change between the updated correlations between the IRT EAP[θ] scores and the other scores/outcomes and those shown in the manuscript.

SSRQ02 I put off making decisions.
SSRQ11 I give up quickly.
SSRQ13 I am able to accomplish goals for myself.
 # SSRQ14 I have personal standards, and try to live up to them.
SSRQ15 As soon as I see a problem or challenge, I start looking for possible solutions.
SSRQ16 I have a hard time setting goals for myself.
 # SSRQ26 If I wanted to change, I am confident that I could do it.
SSRQ29 I am able to resist temptation.

Table B. *Updated correlations between estimated scores of the SSRQ, summed scores of the SSRQ, and outcomes (see bolded)*

	EAP[θ]	Summed Score	IRT SF EAP[θ]	CAT EAP[θ]	GA score	Tree score
EAP[θ]	1.000					
Summed Score	.972	1.000				
IRT SF EAP[θ]	**.965**	**.936**	1.000			
CAT EAP[θ]	.980	.937	**.970**	1.000		
GA score	.951	.978	**.907**	.926	1.000	
Tree score	.896	.905	**.898**	.899	.874	1.000
Nervousness	-.209	-.203	**-.234**	-.253	-.198	-.270
Hopelessness	-.346	-.354	**-.381**	-.389	-.356	-.390
Alcohol Frequency	-.067	-.028	**-.043**	-.049	-.040	-.042
Coffee per Day	.068	.091	**.070**	.068	.053	.184
BSCS	.746	.763	**.743**	.726	.728	.699
Grit-S	.769	.759	**.774**	.784	.734	.756

Note: EAP[θ]=expected a posteriori θ estimate; IRT SF=item response theory short-form; CAT= Computerized Adaptive Testing (maximize Fisher's information); GA=genetic algorithm; BSCS=Brief Self-control Scale; Grit-S=Short Grit Scale

Genetic Algorithm short-form with all of the data (no collapsing, no item elimination).

With 29 items, the items selected by the genetic algorithm were mostly items that had been eliminated. The correlation between the scores from the genetic algorithm in the supplemental materials and the one from the manuscript is *r*=.948, so the relations were largely maintained. The items, correlations with other scores, and correlations with outcomes are shown below. The updated correlations between the genetic algorithm scores and the other scores are smaller, but the correlations with outcomes are a bit larger than in the manuscript.

SSRQ01 I don't notice the effects of my actions until it's too late.
SSRQ05 I don't seem to learn from my mistakes.
SSRQ13 I am able to accomplish goals for myself.
SSRQ15 As soon as I see a problem or challenge, I start looking for possible solutions.
SSRQ18 I have trouble making plans to help me reach my goals.
SSRQ22 I have trouble making up my mind about things.
SSRQ25 Once I have a goal, I can usually plan how to reach it.
SSRQ28 I have a lot of willpower.

Table C. *Updated correlations between estimated scores of the SSRQ, summed scores of the SSRQ, and outcomes (see bolded)*

	EAP[θ]	Summed Score	IRT SF EAP[θ]	CAT EAP[θ]	GA score	Tree score
EAP[θ]	1.000					
Summed Score	.972	1.000				
IRT SF EAP[θ]	.967	.949	1.000			
CAT EAP[θ]	.980	.937	.958	1.000		
GA score	**.930**	**.951**	**.931**	**.919**	1.000	
Tree score	.896	.905	.907	.899	**.895**	1.000
Nervousness	-.209	-.203	-.220	-.253	**-.248**	-.270
Hopelessness	-.346	-.354	-.344	-.389	**-.381**	-.390
Alcohol Frequency	-.067	-.028	-.071	-.049	**-.008**	-.042
Coffee per Day	.068	.091	.056	.068	**.100**	.184
BSCS	.746	.763	.779	.726	**.746**	.699
Grit-S	.769	.759	.792	.784	**.763**	.756

Note: EAP[θ]=expected a posteriori θ estimate; IRT SF=item response theory short-form; CAT= Computerized Adaptive Testing (maximize Fisher's information); GA=genetic algorithm; BSCS=Brief Self-control Scale; Grit-S=Short Grit Scale

Trees for tailored testing with all of the data (no collapsing, no item elimination).

With 29 items and a p-value threshold of .05, the first few items selected by the conditional tree were largely the same as in the manuscript. The tree yields higher scores because the range of the sum of 29 SSRQ items (used here) is larger than the range of 20 SSRQ items (used in the manuscript; note that the trees are different). The correlation between the scores from the tree presented below and the tree from the manuscript is r=.953. The items, correlations with other scores, and correlations with outcomes are shown below. The updated correlations between the tree scores and other scores/outcomes are a bit larger than in the manuscript.

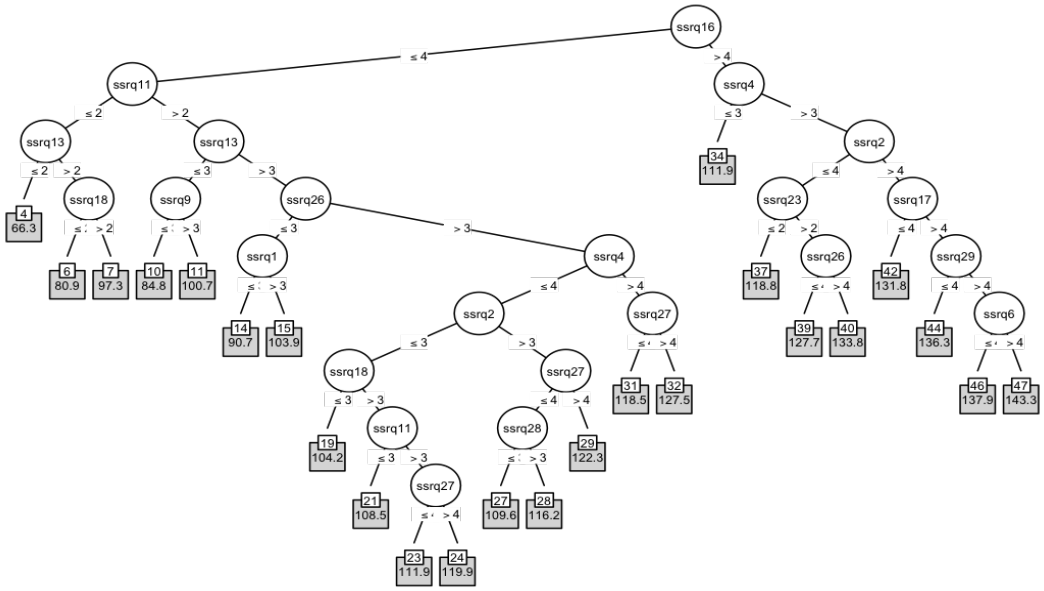


Table D. *Updated correlations between estimated scores of the SSRQ, summed scores of the SSRQ, and outcomes (see bolded)*

	EAP[θ]	Summed Score	IRT SF EAP[θ]	CAT EAP[θ]	GA score	Tree score
EAP[θ]	1.000					
Summed Score	.972	1.000				
IRT SF EAP[θ]	.967	.949	1.000			
CAT EAP[θ]	.980	.937	.958	1.000		
GA score	.951	.978	.916	.926	1.000	
Tree score	**.882**	**.891**	**.893**	**.898**	**.862**	1.000
Nervousness	-.209	-.203	-.220	-.253	-.198	**-.342**
Hopelessness	-.346	-.354	-.344	-.389	-.356	**-.439**
Alcohol Frequency	-.067	-.028	-.071	-.049	-.040	**.009**
Coffee per Day	.068	.091	.056	.068	.053	**.162**
BSCS	.746	.763	.779	.726	.728[a]	**.708**
Grit-S	.769	.759	.792	.784	.734	**.748**

Note: EAP[θ]=expected a posteriori θ estimate; IRT SF=item response theory short-form; CAT= Computerized Adaptive Testing (maximize Fisher's information); GA=genetic algorithm; BSCS=Brief Self-control Scale; Grit-S=Short Grit Scale

Supplementary materials: Selecting items from scales using the Lasso

In this appendix, we discuss Lasso regularization to select items that predict the summed score of a measure. We start by introducing linear regression with regularization and the Lasso penalty. Then, we discuss a common implementation of the Lasso and some conceptual challenges of using the Lasso to predict the summed score of individual items. Next, we apply the Lasso to select items that predict the summed score of the data from the main text. Finally, we explain why a modification of the common loss function for the Lasso is needed for this specific application and why the genetic algorithm with OLS regression and regression trees do not have this limitation.

What is regularization via the Lasso?

The goal of regularization is to improve the generalizability of a linear model. When regression coefficients are estimated via OLS regression, the loss function is trying to minimize the residual sums of squares between what we observe in our data and what the model predicts. Regularization extends the loss function of a linear model by introducing a penalty that controls for the size of the estimated coefficients, which in turn shrinks the estimated coefficients toward zero. Although coefficients estimated with regularization are not optimal for a specific sample (i.e., there is some bias in our coefficient estimates), it is expected that the model is more likely to generalize to other samples because our estimated coefficients do not overfit the training dataset. That is, coefficients do not learn the idiosyncrasies of the training dataset. There are several types of regularizing penalties that could extend the loss function of linear regression, but in this case we focus on regularization using the Lasso penalty (here referred to as *the Lasso*). The loss function for the Lasso is,

$$Lasso = RSS + \lambda \sum |\beta_j| \,,$$

where the first part, the RSS, is the residual sums of squares (which is the loss function of OLS regression) and the second part is the Lasso penalty. In this case, the Lasso controls of the sum of the absolute size of the coefficients. The influence of the penalty is controlled by λ, which is a tuning parameter commonly found through k-fold cross-validation to minimize prediction error in the left out fold. A property of the Lasso is that it tends to favor models that are sparse – the Lasso penalty can shrink coefficients to exactly zero, so a predictor falls out of the model, acting as a proxy of variable selection.

Conceptual challenges to select items by the Lasso for this problem

Items are selected by predicting the total score of the items using individual item responses. If we fit an OLS regression model to predict the total score from item responses, the true model would keep all of the items and give them a coefficient of 1. That is because the total score is a linear combination of item responses with unit weights, and regression recovers that true model. In this case, the variance explained is 1 and the residual sums of squares (RSS) is 0.

Now, consider using the lasso to predict the summed score from individual items. As mentioned above, we know that when all of the items are in the model (no regularization), the RSS is zero, so the lasso loss function is already at a low value.

a. If the tuning parameter is determined by 10-fold cross-validation to minimize prediction error in a left-out fold, then a model that keeps all of the items is the one that minimizes the prediction error (because there is no prediction error).
b. The tuning parameter, and in turn the lasso penalty, would have to be rather high to start dropping items from the model. Empirically, however, the tuning parameter would be rather small.

We expect for lasso to have a different performance when items are selected to reduce prediction error of an external outcome. However, in this case the outcome for the lasso model (summed score) is fully determined by the predictors (items). Therefore, given the nature of the problem we do not think that lasso would be a good candidate to select items with this common implementation to choose λ.

Empirical illustration

Below, we use the Lasso predict the summed score of the 20 SSRQ items from the individual item responses. The model was fit using the glmnet R-package. Code is shown below:

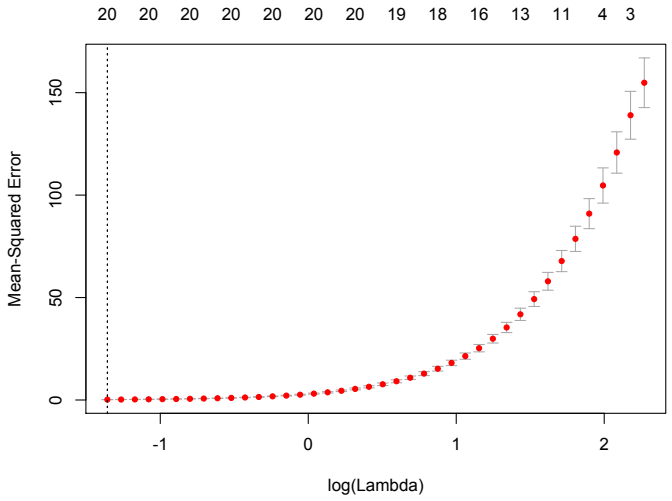
```
library(glmnet)
lassum=rowSums(i2)
x_tr=model.matrix(lassum~.,data=as.data.frame(i2))[,-1] #save
 predictors
y_tr=lassum #save outcomes

p2=cv.glmnet(y=y_tr,x=x_tr,alpha=1,nfolds=10)
p2
plot(p2) #MSE vs tuning parameter
p2$lambda.1se #best penalty value
coef(p2,s=p2$lambda.1se) #coefficients remaining in the model
```

The output below shows the relation between the number of items in the model, variance explained, and the tuning parameter ("lambda") value. This relation is also plotted below:

```
      Df   %Dev Lambda
 [1,]  0 0.0000 9.6880
 [2,]  3 0.1172 8.8270
 [3,]  4 0.2396 8.0430
 [4,]  4 0.3416 7.3280
 [5,]  6 0.4303 6.6770
 [6,]  7 0.5085 6.0840
 [7,] 11 0.5812 5.5440
 [8,] 11 0.6468 5.0510
 [9,] 11 0.7011 4.6020
[10,] 13 0.7477 4.1940
[11,] 14 0.7873 3.8210
[12,] 16 0.8210 3.4820
[13,] 16 0.8496 3.1720
[14,] 16 0.8732 2.8900
[15,] 18 0.8934 2.6340
[16,] 18 0.9104 2.4000
```

```
[17,] 18 0.9245 2.1860
[18,] 19 0.9364 1.9920
[19,] 19 0.9468 1.8150
[20,] 19 0.9554 1.6540
[21,] 19 0.9625 1.5070
[22,] 20 0.9687 1.3730
[23,] 20 0.9740 1.2510
[24,] 20 0.9784 1.1400
[25,] 20 0.9821 1.0390
[26,] 20 0.9851 0.9465
[27,] 20 0.9876 0.8624
[28,] 20 0.9897 0.7858
[29,] 20 0.9915 0.7160
[30,] 20 0.9929 0.6524
[31,] 20 0.9941 0.5944
[32,] 20 0.9951 0.5416
[33,] 20 0.9959 0.4935
[34,] 20 0.9966 0.4497
[35,] 20 0.9972 0.4097
[36,] 20 0.9977 0.3733
[37,] 20 0.9981 0.3401
[38,] 20 0.9984 0.3099
[39,] 20 0.9987 0.2824
[40,] 20 0.9989 0.2573
[41,] 20 0.9991 0.2345
```



The output suggests that there is an optimal tuning parameter value for the Lasso that leads to a mean squared error close to zero (λ = .2573).

Below are the estimated regression coefficients in the Lasso model using the optimal lambda value (λ = .2573) determined via 10-fold cross-validation.

```
ssrq1       1.0158033
ssrq2       0.9551188
ssrq3       0.9379576
```

```
ssrq4       1.0075417
ssrq7       0.8206864
ssrq8       0.9306273
ssrq9       0.8726878
ssrq10      1.0136442
ssrq11      0.9946258
ssrq12      0.9120342
ssrq13      1.0414334
ssrq14      0.9541322
ssrq15      1.0279655
ssrq16      1.0266523
ssrq17      0.9692923
ssrq21      0.9095152
ssrq24      0.9991401
ssrq26      0.9822395
ssrq27      0.8971796
ssrq29      0.9854285
```

As expected, there are 20 coefficients, so the Lasso did not favor a sparser model. Given that the Lasso did not select less items to predict the summed score. Also, note that the estimated regression coefficients are all relatively close to 1, which is what we expect from a model estimated via OLS without regularization. Perhaps one could modify the typical lasso procedure to force some regularization by choosing a tuning parameter that does not minimize prediction error in the left-out fold. Two options to do this would be to 1) choose a λ value associated with a desired number of items to keep, or 2) include a fit function (similar to the genetic algorithm) that balances the prediction error and the number of items in the short form.

Illustration

From the object created above, we can extract information about the number of nonzero elements, the mean cv error, and associated lambda values.

```
cbind(p2$nzero,p2$lambda,p2$cvm,p2$cvsd)
```

```
   nzero    lambda     cv mean       cv sd
s0     0 9.6875409 156.7744276 11.38395398
s1     3 8.8269265 139.8251540 11.05446313
s2     4 8.0427667 121.4459838 10.04214686
s3     4 7.3282694 105.3576109  9.00474022
s4     6 6.6772462  91.5777366  8.03800019
s5     7 6.0840580  79.0798465  7.13664259
s6    11 5.5435671  67.9036439  6.35749492
s7    11 5.0510919  57.7903727  5.60695834
s8    11 4.6023668  49.0131124  4.80710997
s9    13 4.1935052  41.5507043  4.09341031
s10   14 3.8209658  35.0817133  3.50130915
s11   16 3.4815217  29.6047790  2.99719577
s12   16 3.1722329  24.9642020  2.57574987
s13   16 2.8904205  21.0716168  2.20516631
s14   18 2.6336435  17.8101256  1.89191271
s15   18 2.3996779  15.0552024  1.62423419
s16   18 2.1864971  12.7262114  1.38934232
s17   19 1.9922547  10.7317742  1.16745216
s18   19 1.8152683   8.9972873  0.98027649
```

```
s19   19 1.6540049   7.5472956  0.82425951
s20   19 1.5070676   6.3421190  0.69339849
s21   20 1.3731839   5.3193524  0.58261313
s22   20 1.2511940   4.4230225  0.48651707
s23   20 1.1400414   3.6728738  0.40366798
s24   20 1.0387633   3.0497738  0.33528530
s25   20 0.9464824   2.5325527  0.27858054
s26   20 0.8623995   2.1032141  0.23124356
s27   20 0.7857863   1.7465764  0.19205615
s28   20 0.7159792   1.4509530  0.15951319
s29   20 0.6523736   1.2048225  0.13237728
s30   20 0.5944185   1.0009051  0.11010268
s31   20 0.5416120   0.8313905  0.09146760
s32   20 0.4934967   0.6908809  0.07595102
s33   20 0.4496558   0.5738513  0.06312573
s34   20 0.4097096   0.4770213  0.05256561
s35   20 0.3733122   0.3962001  0.04367215
s36   20 0.3401481   0.3291146  0.03627593
s37   20 0.3099303   0.2734176  0.03016940
s38   20 0.2823970   0.2270631  0.02506387
s39   20 0.2573096   0.1887132  0.02089119
```

In this case, suppose that we wanted to have a short-form with 11 items, so we can obtain the estimates from the coefficient estimates from the Lasso associated with a lambda value of ~4.602, which in this case is the 9th lambda entry.

```
coef(p2,s=p2$lambda[9])
                      1
ssrq1        0.4623873
ssrq2        0.1872801
ssrq3        .
ssrq4        1.1398723
ssrq7        .
ssrq8        .
ssrq9        .
ssrq10       0.3848897
ssrq11       0.8644495
ssrq12       .
ssrq13       1.4892782
ssrq14       .
ssrq15       0.5555975
ssrq16       1.4147531
ssrq17       .
ssrq21       .
ssrq24       0.7323111
ssrq26       0.4824611
ssrq27       .
ssrq29       0.4444281
```

As shown above, the items with nonzero coefficients are the items that are retained in the short-form. For the measure that selected eleven items, the MSE in the testing dataset was 51.963, and the correlation between the predicted scores and the scores on the full measure was .966. When those eleven items were used in an OLS regression, the MSE in the testing dataset was 9.086, and the correlation between the predicted scores and the scores on the full measure was .973. It is

important to note that the regression coefficients from the Lasso procedure were substantially smaller than the coefficients for OLS regression.

Similarly, we could follow the same steps as the genetic algorithm by having a loss function that balances prediction error and the number of items, with a maximum number of items. In this case, a proposed loss function might be,

$$Loss = RMSE_{lasso} + l * \#items\ ,$$

with a maximum of ten items. Below is code to carry out this procedure.

```
p2=cv.glmnet(y=y_tr,x=x_tr,alpha=1,nfolds=10,nlambda=500)
```

The 30th entry had the minimum penalty that kept 10 items. We can estimate RMSE and penalize the number of items remaining with a cost of 0.50 with the code below.

```
(sqrt(p2$cvm)+p2$nzero*.5)[1:30]
```

```
      s0       s1       s2       s3       s4       s5       s6       s7
12.43520 12.84796 12.73220 13.08947 13.43916 13.27996 13.61666 13.45138
      s8       s9      s10      s11      s12      s13      s14      s15
13.28594 13.12196 12.96148 12.80456 12.65057 12.49948 12.35153 12.20656
     s16      s17      s18      s19      s20      s21      s22      s23
12.56461 12.42551 12.28821 12.65164 12.51482 12.37925 12.24543 12.11353
     s24      s25      s26      s27      s28      s29
11.98342 12.35534 12.72837 12.60160 12.47463 13.34694
```

The lambda value associated with the bolded estimate kept 6 items in the short-form.

```
ssrq1        .
ssrq2        .
ssrq3        .
ssrq4        0.9731469
ssrq7        .
ssrq8        .
ssrq9        .
ssrq10       .
ssrq11       0.7113622
ssrq12       .
ssrq13       1.3239144
ssrq14       .
ssrq15       .
ssrq16       1.3053623
ssrq17       .
ssrq21       .
ssrq24       0.3033323
ssrq26       0.2818106
ssrq27       .
ssrq29       .
```

For the measure that selected six items, the MSE in the testing dataset was 89.627, and the correlation between the predicted scores and the scores on the full measure was .937. When those six items were used in an OLS regression, the MSE in the testing dataset was 18.550, and the correlation between the predicted scores and the scores on the full measure was .944. Similar to above, the regression coefficients from the Lasso procedure were smaller than the coefficients for OLS regression.

It is important to note that for the genetic algorithm with eight items there was a slightly higher correlation between the predicted and the observed summed scores (r=.978) than either of the lasso methods in the testing dataset. Although these proposed extensions of the Lasso to select items relative to the total score are interesting, future directions include exploring the properties of these extensions even further.

Supplementary materials: Selecting items from scales that assess formative constructs

In this appendix, we discuss several challenges to select less items from a measure that assesses a formative construct. We start by defining formative constructs and how they are different from reflective constructs. Then, we discuss several challenges for constructing formative scales. Finally, we use two simulated datasets to illustrate the methodology assuming that items from a formative measure are interchangeable and content-valid. The discussion of this topic relies heavily on Bollen (2011) and Murray and Booth (2019).

What are formative constructs?

A formative construct can be thought of as the outcome of a combined influence of a set of items, or as a prediction model in which the items predict the outcome. For example, consider the construct of stress, which could be measured by asking respondents if they have experienced a recent death in the family, loss of retirement savings in the stock market, high demand in the workplace, etc. Traditional measurement models would have a hard time accommodating the construct of stress with said indicators because most psychometric theory deals with reflective constructs, which are constructs thought to cause the item responses. Consider a measure of extraversion, where as extraversion increases, a respondent is more likely to endorse that they feel comfortable giving a talk in front of large audiences. However, those relations are not likely to hold in our stress measure - as stress increases, is the death of a family member more likely? Or, more likely to lose retirement savings in the stock market? Theory and mental experiments could help us determine if our measure conceptualizes a construct as formative or reflective (Bollen & Lennox, 1991).

Formative constructs are measured by causal indicators, which as mentioned before, are thought the cause the construct. In the rest of this appendix, the terms formative constructs and causal indicator models are used interchangeably. There are a lot of controversies in the use of these models, and many of these controversies are thought to stem from the lack of development of psychometric theory in this area (Murray & Booth, 2019). For example, it is unlikely that a formative scale could be developed and evaluated using the traditional psychometric models that assume a reflective measure. This supplement does not seek to solve these theoretical or empirical controversies, but rather focuses on introducing how causal indicator models are conceptualized and the assumptions of why selecting items from formative measures might not always be feasible.

Models. It is important to introduce structural models to draw a stronger distinction between a measure that assumes a reflective construct and a measure that assumes a formative construct (Bollen, 2011). Below is the structural model for a measure that assumes a reflective construct,

$$\boldsymbol{x} = \boldsymbol{\tau} + \boldsymbol{\lambda}\eta + \boldsymbol{\epsilon}\,.$$

In this case, **x** is a vector of item responses, η is the respondent's standing on the measured construct, τ is a vector of item intercepts, λ is a vector of factor loadings representing the strength of the relation between the construct and the item responses, and ϵ is the unique factor score that also determines the item responses. Note that for a measure that assesses a reflective construct,

the construct appears on the right-hand side of the equation and is thought to cause the item responses. Now, below is the structural model for a measure that assumes a formative construct,

$$\eta = i_f + \boldsymbol{\gamma}'\boldsymbol{x} + \zeta.$$

In this case, **x** is a vector of item responses, η is the respondent's standing on the measured construct, γ is a vector of coefficients that represent the strength of the relation between the item responses and the construct, i_f is the intercept, and ζ is the error disturbance of the construct. There are three things to note about this model. First, the construct η appears on the left-hand side of the equation and is thought to be the outcome of the item responses. Second, the model above is not identified, but two reflective indicators could be included to identify the model, which is often referred to as a MIMIC model. Third, the η is not perfectly determined by the item responses because there is an error term ζ in the equation. When the item responses combine linearly to perfectly predict the construct, then the outcome is referred to as a composite, defined with the following structural model,

$$C = i_c + \boldsymbol{\beta}'\boldsymbol{x}\,.$$

It is important to consider the structural model for composites because the main goal of the paper is the select items that predict the summed score of the whole item set. The summed score is a composite made up of a perfect linear combination of the items with unit-weights, and as such the equation for the composite structural model does not have an error term. According to Bollen (2011), the term *formative indicator* (perhaps alluding to a measure of a formative construct) has been used in previous literature to refer to indicators of composites, so the distinction between causal indicators and composite indicators has not always been clear.

Challenges in selecting items from formative measures

When researchers assess constructs, a fundamental property of the construct indicators is that they should have conceptual unity. That is, the observed indicators must match the theoretical definition of the construct (Bollen, 2011). Reflective constructs, formative constructs, and composites differ in the extent to which their indicators have conceptual unity.

- Reflective constructs assume that there is conceptual unity of the indicators. However, a feature of the model is conceptual parallelism, which is that all indicators are conceptually interchangeable with one another, and thus the meaning of the latent variable is not changed if items are dropped or swapped (Murray & Booth, 2019). Also, note that items/indicators are correlated because they have a common cause: the reflective construct (e.g. we expect a correlation between talking to strangers and being comfortable with addressing large audiences in our extraversion measure).
- Formative constructs assume that there is conceptual unity of the indicators. However, the indicators do not have conceptual parallelism (i.e. items are not interchangeable) because they do not reflect the same common cause (Murray & Booth, 2019). Given that causal indicators largely determine the formative construct, removing, adding, or changing items could fundamentally change the nature of the construct that is measured (Bollen, 2011). Therefore, it is recommended to include more items than less in order to be comprehensive and over-inclusive. Also, note that items/indicators are not necessarily correlated, as they could be independent causes of the formative construct (e.g. we do not

expect a correlation between the death of a family member and high demand at workplace in our stress measure). Empirically, it poses some challenges to select items when the items are not correlated – it is unlikely that they will account for the same variance in the outcome when the items do not correlate.

- Composites do not require conceptual unity of the indicators or for items to be correlated. However, the main goal of the paper is to predict the summed score composite of the measure from individual items. Given that the measure assesses a construct, conceptual unity in the indicators is desired.

In essence, it is easier to justify item selection from reflective measures and composites than from formative measures. Bollen (2011) mentions that, "decisions on whether to eliminate indicators must be made taking into account of the theoretical appropriateness of the indicator and its empirical performance in the researcher's and the studies of others," as in:

1. If there is strong a priori theory that the item is important to define the construct, then the item should be kept in the scale.
2. If the theory does not match repeated empirical tests about the relation between the item and the construct, then the item is candidate to be removed. For example, if the regression coefficient between the item and the construct is not significant or of the opposite sign than hypothesized, then the item might be considered for removal.
3. If causal indicators were to be highly correlated, estimating the unique influence of the indicator on the formative construct might be difficult due to multicollinearity.

Murray and Booth (2019) seem to be more conservative when it comes to item selection for formative constructs: "while there are many ways in which a given indicator could be considered good or bad, alignment to the theoretical nature of the proposed construct (classically referred to as content validity) is of the upmost importance in causal indicator models." That is because removing an indicator might be removing a noncompensable piece of the whole scale.

Overall, there are three aspects of formative constructs that make item selection challenging. First, item selection should be done by theory so that the nature of the items does not change. Second, items are not exchangeable, so remaining items might not cover the content span of the construct. Third, indicators might not be correlated, so empirically the variance that the indicator explains in the construct might not be substituted by another indicator. Note also that psychometric approaches for item selection that rely on latent variable models might not work when items do not correlate – there is nothing to factor-analyze. With these limitations in mind, if researchers wanted to select items from a formative measure, there might be one way to proceed: assume that any combination of causal indicators cover the content span and that indicators are interchangeable.

Empirical examples that <u>assume</u> that items are interchangeable and cover the content span
In a hypothetical situation where causal indicators were interchangeable and content-valid in a certain measure, then it is hypothesized that selecting items with machine learning methods (i.e. genetic algorithms and regression trees) might be more useful than psychometric methods introduced in the main text of the manuscript (i.e. item response theory short-forms and computerized adaptive testing). The two reasons why these psychometric models might fail for formative measures are that 1) they assume a reflective measure and 2) they assume that items

correlate. Causal indicator models do not match neither the theoretical or the empirical structure of psychometric models, so perhaps machine learning algorithms are expected to do better in this case (although including two reflective indicators and fitting a MIMIC model might be an option to use psychometric methods for formative measures; Bollen, 2011).

Below we present two examples using simulated datasets into how each of these approaches might perform.

Scenario 1. Scale is strongly formative (no correlation between indicators). A dataset of N=2,000 was simulated from a 20x20 diagonal correlation matrix, so items were not correlated. Simulated indicators come out as standard normal variables, so thresholds were imposed at -1.5, -.5, .5, and 1.5 to create discrete variables that ranged from 0 to 4. Items per respondent were then summed and a random standard normal variable was added to emulate the error term in a causal indicator model. In theory, the true model is 20 variables, with a regression coefficient of 1, predicting the summed score.

The first 1,000 simulated cases were used to train the model, and the last 1,000 were used to evaluate the model. Below, we start by empirically showing why the psychometric approaches might not be appropriate (besides that the data-generating model does not match the assumptions by the model). Then, we show results from the genetic algorithm and the regression trees as estimated in the main text.

Psychometric Models: Below we show the estimates of the item parameters of the IRT model. Even though there might not be estimation issues, the item parameter estimates are nonsensical – the threshold parameters are outside conventional bounds and the slope parameters are very low. Ideally, we would want to see all of the slope parameters above a values of 1 and thresholds within -3 and 3. As expected, it does not appear that the items are providing much information.

	a	b1	b2	b3	b4
par	0.037	-70.374	-20.133	21.021	66.782
par	0.264	-10.832	-2.656	3.682	10.290
par	-0.072	37.260	10.663	-11.789	-36.590
par	0.291	-8.819	-2.857	3.231	9.920
par	-0.164	15.616	4.221	-5.660	-18.248
par	-0.054	49.219	16.107	-13.570	-51.414
par	-0.047	54.046	17.516	-16.819	-56.017
par	-0.315	8.191	2.382	-2.574	-8.647
par	0.277	-9.455	-3.024	2.805	9.926
par	0.166	-17.827	-4.602	4.468	15.582
par	0.389	-6.703	-2.208	2.293	6.745
par	0.113	-24.132	-7.156	7.874	24.761
par	-0.016	158.135	44.190	-63.353	-176.164
par	-0.016	162.546	51.693	-48.820	-158.705
par	-0.036	68.660	23.691	-21.856	-79.591
par	0.371	-7.126	-2.446	2.032	6.999
par	0.134	-19.264	-6.129	6.196	21.912
par	-0.114	24.191	7.584	-7.326	-24.188
par	0.471	-6.205	-1.634	1.791	5.863
par	-0.044	56.938	17.612	-19.740	-60.791

Furthermore, below are the estimates of the first ten factor scores and their standard errors, respectively, when all of the items are administered:

```
                F1     SE_F1
 [1,] -0.30394174 0.8861455
 [2,] -0.06354447 0.8689858
 [3,]  0.89028514 0.8829736
 [4,]  0.55180790 0.8713201
 [5,]  0.15552731 0.8799634
 [6,] -0.29415757 0.8581454
 [7,]  0.21436755 0.8846638
 [8,]  0.04824316 0.8704330
 [9,] -0.77361296 0.8681216
[10,]  0.05255194 0.8827381
```

The standard errors for the rest of the 990 cases are also around .88. One might be able to notice that the standard errors are very large relative to the estimate of the factor score, which has implications for adaptive testing. As a reminder, in computerized adaptive testing researchers can administer items conditional on a person's response. Items could be administered until the standard error of the factor score falls below a prespecified value. In the main text, the stopping point for the adaptive test was a standard error was .30, which is associated with a reliability of .90 for the score. If we wanted to keep a similar stopping point, then adaptive testing might not help us – factor scores estimated from less items have larger standard errors than factor scores estimated with all of the items. In conclusion, the lack of informative items and the large standard errors empirically suggests that short-forms built using item response functions and computerized adaptive testing should not be used.

Machine Learning models: As described in the text, the goal of the machine learning algorithms is to predict the summed score of the item responses. However, if items do not correlate, it is unlikely that some respondents have uniformly high or uniformly low scores across all items. Summary statistics suggest that the lowest observed summed score in the simulated data was 25, the mean was 40, and the max was 56, which might be the distribution expected if respondents were endorsing items at random. Below is the estimated tree structure using conditional inference trees using a p-value of .05 as the threshold for splitting:

```
 [1] root
|   [2] v16 <= 1
|   |   [3] v20 <= 1
|   |   |   [4] v20 <= 0: 35.878 (n = 36, err = 481.5)
|   |   |   [5] v20 > 0: 38.441 (n = 73, err = 975.6)
|   |   [6] v20 > 1
|   |   |   [7] v13 <= 2
|   |   |   |   [8] v17 <= 0: 35.803 (n = 12, err = 101.5)
|   |   |   |   [9] v17 > 0
|   |   |   |   |   [10] v9 <= 1: 38.191 (n = 42, err = 553.0)
|   |   |   |   |   [11] v9 > 1: 40.619 (n = 113, err = 1330.2)
|   |   |   [12] v13 > 2
|   |   |   |   [13] v11 <= 1: 38.209 (n = 12, err = 229.4)
|   |   |   |   [14] v11 > 1
|   |   |   |   |   [15] v18 <= 2: 42.639 (n = 36, err = 336.9)
|   |   |   |   |   [16] v18 > 2: 45.438 (n = 15, err = 163.0)
|   [17] v16 > 1
|   |   [18] v6 <= 2
|   |   |   [19] v3 <= 1
|   |   |   |   [20] v4 <= 1: 37.897 (n = 49, err = 1127.2)
|   |   |   |   [21] v4 > 1
|   |   |   |   |   [22] v11 <= 1: 38.251 (n = 25, err = 277.5)
|   |   |   |   |   [23] v11 > 1
|   |   |   |   |   |   [24] v18 <= 1: 38.848 (n = 19, err = 127.4)
|   |   |   |   |   |   [25] v18 > 1
|   |   |   |   |   |   |   [26] v9 <= 2: 40.677 (n = 37, err = 245.0)
|   |   |   |   |   |   |   [27] v9 > 2: 43.630 (n = 19, err = 203.0)
|   |   |   [28] v3 > 1
|   |   |   |   [29] v1 <= 1
|   |   |   |   |   [30] v10 <= 1: 37.826 (n = 23, err = 503.0)
|   |   |   |   |   [31] v10 > 1: 40.816 (n = 63, err = 740.3)
|   |   |   |   [32] v1 > 1
|   |   |   |   |   [33] v7 <= 2
|   |   |   |   |   |   [34] v5 <= 2
|   |   |   |   |   |   |   [35] v13 <= 1: 38.931 (n = 22, err = 274.5)
|   |   |   |   |   |   |   [36] v13 > 1: 41.179 (n = 78, err = 938.1)
|   |   |   |   |   |   [37] v5 > 2
|   |   |   |   |   |   |   [38] v12 <= 1: 40.350 (n = 19, err = 267.0)
|   |   |   |   |   |   |   [39] v12 > 1: 44.955 (n = 35, err = 361.6)
|   |   |   |   |   [40] v7 > 2
|   |   |   |   |   |   [41] v2 <= 1: 41.465 (n = 20, err = 386.8)
|   |   |   |   |   |   [42] v2 > 1: 45.707 (n = 53, err = 712.0)
|   |   [43] v6 > 2
|   |   |   [44] v17 <= 3
|   |   |   |   [45] v14 <= 2
|   |   |   |   |   [46] v7 <= 0: 37.948 (n = 12, err = 150.6)
|   |   |   |   |   [47] v7 > 0
|   |   |   |   |   |   [48] v19 <= 1
|   |   |   |   |   |   |   [49] v9 <= 1: 37.984 (n = 11, err = 181.5)
|   |   |   |   |   |   |   [50] v9 > 1: 42.090 (n = 22, err = 212.4)
|   |   |   |   |   |   [51] v19 > 1
|   |   |   |   |   |   |   [52] v14 <= 1: 42.101 (n = 26, err = 319.8)
|   |   |   |   |   |   |   [53] v14 > 1: 44.340 (n = 51, err = 613.4)
|   |   |   |   [54] v14 > 2
|   |   |   |   |   [55] v20 <= 2: 43.958 (n = 37, err = 447.1)
|   |   |   |   |   [56] v20 > 2: 46.968 (n = 23, err = 334.4)
|   |   |   [57] v17 > 3: 48.132 (n = 17, err = 226.3)
```

As we might notice, the tree is rather long and yields predictions (in bold) that range from 35 to 48, which are roughly around the mean of the observed responses. Participants could have received as few as 4 items and at most 7 items. The mean squared error and variance explained in the training dataset were MSE=12.820 and R^2=.394, and the mean squared error and variance explained in the testing dataset were MSE=19.232 and R^2=.164.

Also, below are the estimated coefficients for the items selected by the genetic algorithm. Note that, as in the main text, we specified a maximum of ten items in the short-form. In this case, the genetic algorithm selected a short-form with 10 items that had estimated regression coefficients close to 1.

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 19.25027    0.66467  28.962   <2e-16 ***
v1           1.21634    0.10266  11.848   <2e-16 ***
v2           1.04809    0.09794  10.701   <2e-16 ***
v4           1.11671    0.10061  11.100   <2e-16 ***
v5           1.12677    0.09620  11.713   <2e-16 ***
v6           1.07674    0.10061  10.702   <2e-16 ***
v7           1.01031    0.09862  10.245   <2e-16 ***
v8           1.13273    0.09947  11.388   <2e-16 ***
v10          0.97768    0.10083   9.696   <2e-16 ***
v13          1.01197    0.10047  10.072   <2e-16 ***
v14          1.15609    0.10022  11.536   <2e-16 ***
```

The mean squared error and variance explained in the training dataset were MSE=9.953 and R^2=.534, and the mean squared error and the variance explained in the testing dataset were MSE=10.484 and R^2=.472.

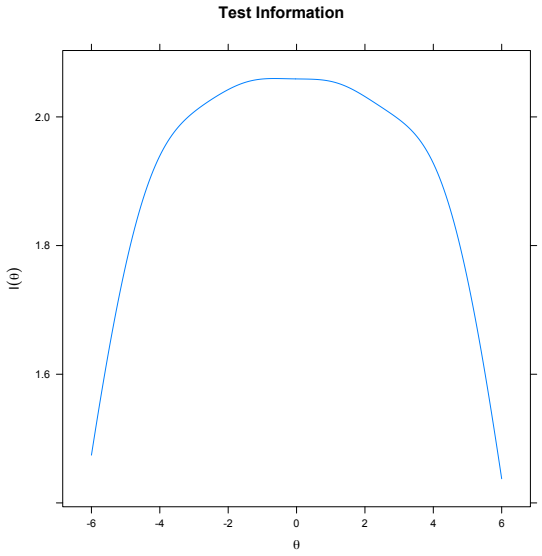
Scenario 2. Scale is weakly formative (correlation of .1 between indicators). Similar steps were followed as above in scenario 1, but now items were allowed to correlated r=.1. We start with a brief discussion about IRT estimation and then we demonstrate results by the genetic algorithm and regression trees.

Psychometric Models. Below are the item parameters estimates. Estimates seem to take values that are within typical ranges, although all of the items are still not very informative.

```
        a     b1     b2    b3    b4
par 0.533 -5.552 -1.658 1.485 5.196
par 0.729 -3.837 -1.311 1.129 3.791
par 0.646 -4.536 -1.216 1.377 4.367
par 0.555 -5.299 -1.628 1.504 4.607
par 0.516 -5.373 -1.769 1.569 5.279
par 0.482 -5.728 -2.083 1.569 5.528
par 0.622 -4.604 -1.546 1.315 4.654
par 0.595 -4.785 -1.584 1.319 4.896
par 0.560 -4.660 -1.634 1.387 5.098
par 0.530 -5.603 -1.643 1.542 5.301
par 0.541 -5.228 -1.619 1.490 4.924
par 0.614 -4.632 -1.600 1.347 4.768
par 0.730 -3.769 -1.278 0.991 3.758
```

```
par 0.544 -5.060 -1.683 1.539 4.867
par 0.487 -5.851 -1.878 1.721 5.351
par 0.590 -4.618 -1.595 1.365 4.542
par 0.467 -5.797 -2.006 1.557 5.162
par 0.728 -3.829 -1.252 1.243 3.831
par 0.490 -5.823 -1.692 1.704 5.572
par 0.618 -4.784 -1.665 1.240 4.405
```

Below is also the test information function, which suggests that by taking all of the items, the expected information around the average of the latent variable is I=2, which is associated with a standard error of 1/I = .70, and a reliability of .50.



Below are also the first ten factor scores and their associated standard errors (the standard errors for the other factor scores are similar):

```
               F1      SE_F1
 [1,]  0.67758078 0.5760443
 [2,]  0.47004271 0.5628637
 [3,] -1.10694852 0.5836182
 [4,] -0.73640215 0.5773445
 [5,] -0.04773171 0.5577663
 [6,] -0.15261788 0.5635032
 [7,] -0.32716215 0.5912569
 [8,]  1.77499463 0.6022286
 [9,] -0.40488132 0.5801024
[10,] -0.39562681 0.5406829
```

Although the standard errors are smaller than in scenario 1, they still might not fit the stopping criteria of an adaptive test of having standard errors of around .30 (associated with reliability of .90). So, in a scenario in which the items barely correlate (r=.1), short-forms built using IRT and adaptive testing still are not the best approach to select items.

Machine Learning models*:* The range of the observed summed scores is a bit wider, with a minimum of 15, a mean of 41, and a maximum of 63. Below is the estimated regression tree.

```
[1] root
|   [2] v20 <= 1
|   |   [3] v1 <= 1
|   |   |   [4] v18 <= 0: 26.551 (n = 11, err = 494.2)
|   |   |   [5] v18 > 0
|   |   |   |   [6] v11 <= 1: 31.746 (n = 42, err = 1186.5)
|   |   |   |   [7] v11 > 1: 36.680 (n = 50, err = 905.4)
|   |   [8] v1 > 1
|   |   |   [9] v6 <= 1
|   |   |   |   [10] v1 <= 2: 34.504 (n = 38, err = 817.6)
|   |   |   |   [11] v1 > 2: 39.473 (n = 30, err = 1097.6)
|   |   |   [12] v6 > 1
|   |   |   |   [13] v16 <= 1: 37.776 (n = 34, err = 927.2)
|   |   |   |   [14] v16 > 1
|   |   |   |   |   [15] v13 <= 2
|   |   |   |   |   |   [16] v3 <= 2: 39.545 (n = 33, err = 729.0)
|   |   |   |   |   |   [17] v3 > 2: 44.265 (n = 22, err = 412.9)
|   |   |   |   |   [18] v13 > 2
|   |   |   |   |   |   [19] v18 <= 2: 43.586 (n = 16, err = 241.0)
|   |   |   |   |   |   [20] v18 > 2: 49.803 (n = 11, err = 112.0)
|   [21] v20 > 1
|   |   [22] v6 <= 1
|   |   |   [23] v1 <= 2
|   |   |   |   [24] v16 <= 1
|   |   |   |   |   [25] v9 <= 2: 32.426 (n = 35, err = 836.2)
|   |   |   |   |   [26] v9 > 2: 37.217 (n = 14, err = 467.9)
|   |   |   |   [27] v16 > 1
|   |   |   |   |   [28] v5 <= 1
|   |   |   |   |   |   [29] v11 <= 2: 33.696 (n = 23, err = 449.3)
|   |   |   |   |   |   [30] v11 > 2: 39.633 (n = 14, err = 241.2)
|   |   |   |   |   [31] v5 > 1
|   |   |   |   |   |   [32] v8 <= 2
|   |   |   |   |   |   |   [33] v3 <= 2: 38.129 (n = 38, err = 909.6)
|   |   |   |   |   |   |   [34] v3 > 2: 44.749 (n = 13, err = 266.0)
|   |   |   |   |   |   [35] v8 > 2: 45.497 (n = 18, err = 411.9)
|   |   |   [36] v1 > 2
|   |   |   |   [37] v18 <= 1: 39.195 (n = 15, err = 479.8)
|   |   |   |   [38] v18 > 1: 45.217 (n = 43, err = 1148.6)
|   |   [39] v6 > 1
|   |   |   [40] v17 <= 2
|   |   |   |   [41] v4 <= 2
|   |   |   |   |   [42] v16 <= 2
|   |   |   |   |   |   [43] v13 <= 1
|   |   |   |   |   |   |   [44] v18 <= 1: 32.419 (n = 11, err = 287.1)
|   |   |   |   |   |   |   [45] v18 > 1: 37.343 (n = 24, err = 369.0)
|   |   |   |   |   |   [46] v13 > 1
|   |   |   |   |   |   |   [47] v5 <= 1
|   |   |   |   |   |   |   |   [48] v15 <= 1: 34.529 (n = 10, err = 269.3)
|   |   |   |   |   |   |   |   [49] v15 > 1
|   |   |   |   |   |   |   |   |   [50] v14 <= 2: 38.756 (n = 14, err =
147.3)
|   |   |   |   |   |   |   |   |   [51] v14 > 2: 42.897 (n = 8, err = 115.2)
|   |   |   |   |   |   |   [52] v5 > 1
|   |   |   |   |   |   |   |   [53] v12 <= 1
|   |   |   |   |   |   |   |   |   [54] v11 <= 2: 38.686 (n = 20, err =
257.2)
|   |   |   |   |   |   |   |   |   [55] v11 > 2: 44.915 (n = 7, err = 69.9)
```

```
|   |   |   |   |   |   |   |   [56] v12 > 1
|   |   |   |   |   |   |   |   |   [57] v19 <= 2: 42.974 (n = 33, err =
429.7)
|   |   |   |   |   |   |   |   |   [58] v19 > 2: 46.301 (n = 14, err = 78.4)
|   |   |   |   |   [59] v16 > 2
|   |   |   |   |   |   [60] v11 <= 2: 42.413 (n = 42, err = 888.1)
|   |   |   |   |   |   [61] v11 > 2: 46.955 (n = 19, err = 552.6)
|   |   |   |   [62] v4 > 2
|   |   |   |   |   [63] v11 <= 2
|   |   |   |   |   |   [64] v17 <= 1: 40.118 (n = 26, err = 473.1)
|   |   |   |   |   |   [65] v17 > 1
|   |   |   |   |   |   |   [66] v14 <= 2: 43.382 (n = 30, err = 485.0)
|   |   |   |   |   |   |   [67] v14 > 2: 48.095 (n = 17, err = 369.6)
|   |   |   |   |   [68] v11 > 2
|   |   |   |   |   |   [69] v19 <= 2: 47.118 (n = 28, err = 702.2)
|   |   |   |   |   |   [70] v19 > 2: 51.851 (n = 21, err = 349.5)
|   |   |   [71] v17 > 2
|   |   |   |   [72] v16 <= 2
|   |   |   |   |   [73] v11 <= 2
|   |   |   |   |   |   [74] v15 <= 1: 40.337 (n = 19, err = 626.0)
|   |   |   |   |   |   [75] v15 > 1: 45.901 (n = 48, err = 861.7)
|   |   |   |   |   [76] v11 > 2: 48.248 (n = 40, err = 768.1)
|   |   |   |   [77] v16 > 2
|   |   |   |   |   [78] v12 <= 1: 45.977 (n = 15, err = 322.8)
|   |   |   |   |   [79] v12 > 1
|   |   |   |   |   |   [80] v10 <= 2: 49.819 (n = 31, err = 936.1)
|   |   |   |   |   |   [81] v10 > 2: 54.168 (n = 23, err = 419.3)
```

As we might notice, the tree is longer than in scenario 1 and yields predictions (in bold) in a wider range, from 26 to 54. Participants could have received as few as 3 items and at most 9 items. The mean squared error and variance explained in the training dataset were MSE=18.384 and R^2=.651, and the mean squared error and the variance explained in the testing dataset were MSE=37.436 and R^2=.385. As we can notice, the mse went up from scenario 1, but this is expected given that the variability of the observed responses also increased. Furthermore, the variance explained also increased, which suggests that the correlation between the items increased helped predict the summed score better.

Also, below are the estimated coefficients for the items selected by the genetic algorithm:

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.1722     0.5609   18.13   <2e-16 ***
v1            1.8984     0.1151   16.49   <2e-16 ***
v2            1.5695     0.1153   13.62   <2e-16 ***
v3            1.5246     0.1185   12.86   <2e-16 ***
v6            1.6572     0.1135   14.60   <2e-16 ***
v9            1.5582     0.1146   13.60   <2e-16 ***
v10           1.4361     0.1176   12.21   <2e-16 ***
v11           1.4631     0.1159   12.62   <2e-16 ***
v12           1.3084     0.1147   11.41   <2e-16 ***
v19           1.3934     0.1189   11.72   <2e-16 ***
v20           1.6747     0.1177   14.22   <2e-16 ***
```

In this case, the genetic algorithm also selected a short-form with 10 items that had estimated regression coefficients higher than 1. The mean squared error and variance explained in the

training dataset were MSE=12.806 and R^2=.758, and the mean squared error and variance explained in the testing dataset were MSE=15.260 and R^2 = .718.

Overall, our simulated example suggests that the genetic algorithm and regression trees perform better than the psychometric models to select items in formative measures mainly because the psychometric models are misspecified (i.e., the items are not reflective indicators of the construct). Again, these results would only be meaningful if the assumptions were to hold – that the remaining items are interchangeable and cover the whole content span.

Supplemental Materials: Simulations to examine the stability of item selection procedures

Below, we present results of two small Monte Carlo simulation studies that examine some of the properties of the item selection procedures presented in the main text. First, we examine the stability of the items that are selected using bootstrapped datasets from the the training data (N=400). Then, we examine how sample size affects the variability in prediction accuracy of the summed score or latent variable score, correspondingly, by simulating from the model implied in the training dataset. Finally, we discuss the general conclusions of the small simulation studies.

Stability of the item selection solutions

There were 1,000 bootstrapped samples drawn from the training dataset, and the models described in the main text were fitted. Estimates of interest across approaches were:

1. Static short-forms with IRT: Distribution of the slope parameter
2. Computerized adaptive testing: Average item exposure for each of the items
3. Genetic algorithm: Proportion of times that each item was selected
4. Regression trees: The variability in the first split and cut point selected

Static short-forms with IRT: The mean and empirical standard deviation of the estimate of the slope parameters are presented in the third column of Table A. The results suggest that the item parameter estimation was relatively stable. The empirical standard deviation of the estimates is about $1/10^{th}$ of the mean estimates. Thus, it is likely that highly informative items would remain highly informative, and in turn the items selected for the short-form, after data fluctuations.

Computerized adaptive testing: The mean and empirical standard deviation of the item exposure estimate are presented in the fourth column of Table A. The variability in individual item exposure varied as a function of how often respondents received the item – there was less variability across bootstrapped datasets from items that were rarely or almost always administered than those that were administered between 20% to 40% of the time.

Genetic algorithm: The proportion of times that items were selected in models across bootstrapped datasets is presented in the fifth column of Table A. The results suggest that there is a lot of variability in item selection – only six out of the twenty items had slighlty higher than a 50% chance to appear in the short-form. In other words, the genetic algorithm might not have uniformly found the same items in the short-form. One could consider item SSRQ16, which has the second highest IRT slope parameter. For GA, item SSRQ16 was the most commonly selected item, appearing in 72.2% of the short-forms. It is important to highlight that the genetic algorithm is a heuristic search procedure, so it does not guarantee that even the same solutions were explored in each bootstrap sample.

Regression trees: The stability of the first split (i.e., the predictor selected and the split point) is presented in Table B. Results suggest that there was substantial instability in the first split. There were seven different items that appeared as the first split, although item SSRQ16 was the first split in almost half of the trees, and SSRQ13 in an extra 20% of the trees. Furthermore, there was also a lot of variability in the split point selection – splits on item SSRQ16 could have happened on response 1, 2, or 3, with, roughly, a .20-.40-.40 chance, respectively.

Table E. *Mean and empirical standard deviation of the estimates for the slope IRT item parameters, mean exposure rate and empirical standard deviation during the adaptive test for the items, and proportion of times selected by the genetic algorithm across bootstrapped replications.*

Labels	Items	**IRT mean estimate**	**CAT % exposure**	**GA % selected**
SSRQ01	I don't notice the effects of my actions until it's too late.	1.690 (.185)	.185 (.057)	.542
SSRQ02	I put off making decisions.	1.899 (.173)	.482 (.226)	.448
SSRQ03	It's hard for me to notice when I've 'had enough' (alcohol, food, sweets).	1.585 (.164)	.152 (.069)	.397
SSRQ04	I have trouble following through with things once I've made up my mind to do something.	2.423 (.238)	.948 (.099)	.565
SSRQ07	I can usually find several different possibilities when I want to change something.	1.141 (.148)	.129 (.016)	.130
SSRQ08	Often I don't notice what I'm doing until someone calls it to my attention.	1.628 (.176)	.150 (.050)	.274
SSRQ09	I usually think before I act.	1.219 (.154)	.112 (.013)	.243
SSRQ10	I learn from my mistakes.	2.256 (.253)	.516 (.271)	.485
SSRQ11	I give up quickly.	2.592 (.262)	.976 (.085)	.311
SSRQ12	I usually keep track of my progress toward my goals.	2.062 (.213)	.461 (.240)	.313
SSRQ13	I am able to accomplish goals for myself.	3.038 (.305)	.982 (.048)	.409
SSRQ14	I have personal standards, and try to live up to them.	2.056 (.231)	.266 (.190)	.295
SSRQ15	As soon as I see a problem or challenge, I start looking for possible solutions.	2.150 (.206)	.450 (.198)	.513
SSRQ16	I have a hard time setting goals for myself.	2.879 (.265)	1.000 (.004)	.722
SSRQ17	When I'm trying to change something, I pay a lot of attention to how I'm doing.	1.851 (.186)	.244 (.071)	.443
SSRQ21	I know how I want to be.	1.687 (.200)	.165 (.104)	.243
SSRQ24	I tend to keep doing the same thing, even when it doesn't work.	2.004 (.181)	.399 (.274)	.561
SSRQ26	If I wanted to change, I am confident that I could do it.	2.363 (.221)	.825 (.211)	.418
SSRQ27	I can stick to a plan that's working well.	2.073 (.221)	.279 (.195)	.168
SSRQ29	I am able to resist temptation.	1.788 (.160)	.285 (.110)	.576

Table F. Rate and split point at which each item was used as the first split when growing a conditional inference tree across bootstrapped datasets

Item	**First split rate**	**Split point**
SSRQ16	479 (47.9%)	1 (16.7%) 2(42.4%) 3(40.9%)
SSRQ13	235 (23.5%)	1 (9.4%), 2 (90.6%)
SSRQ4	132 (13.2%)	2 (97.8%) 3(2.2%)
SSRQ11	130 (13%)	1 (5.4%) 2 (56.1%) 3(38.5%)
SSRQ26	22 (2.2%)	2 (100%)
SSRQ2	1 (.1%)	2 (100%)
SSRQ10	1 (.1%)	2 (100%)

Prediction accuracy across sample size

For this simulation, a population of *ss*=100,500 was simulated from the implied item response model in the dataset (use the `simdata()` function form the `mirt` r-package). Then, sample *N* cases from the *ss*=100,000, carry out the item selection procedures as described in the main text, estimate prediction accuracy in the *ss*=500 cases, and repeat the procedure 100 times for each sample size. It is expected that sample size would affect the estimate of prediction accuracy, along with the variability, in the following ways:

1. Stability of the slope parameter used for IRT short-forms and adaptive testing
2. Estimation of the regression coefficients used in an intermediate step of the genetic algorithm
3. Stability of the regression trees

Note that the simulated data comes from the implied item response model, which would heavily favor the psychometric methods. However, we chose this approach to facilitate the generation of item response data (see supplementary materials on item selection in formative measures where we discuss that data generated to favor machine learning, e.g., where items that do not correlate could make the step of fitting a psychometric model trivial). Therefore, it is expected that the mean estimate of prediction accuracy will be constant – we are just refitting the data-generating model with less data. Along the same lines, given the expected constant estimates of the prediction accuracy of the IRT short-forms, the previous findings that the slope parameter across bootstrapped datasets is stable, and computational complexity, simulation results for computerized adaptive testing are not presented – they are expected to be similar to the IRT approaches: constant and stable. Lastly, it is expected that the prediction accuracy by the genetic algorithm and regression trees and their precision would increase as sample size increases. The sample sizes studied would be from 200 to 1,000 in 100 increments.

The results are presented in Table C, and suggest that the estimate of prediction accuracy (e.g., the MSE and variance explained) of the summed score from the items selected by the genetic algorithm and the trees increased as sample size increased, and its variability decreased. As expected, the prediction accuracy by the IRT short-form to predict the factor score remained constant across sample size.

Table G. *Estimates of mean square error and variance explained of prediction accuracy and its variability across item selection approaches and sample sizes.*

N	IRT MSE	IRT R^2	GA MSE	GA R^2	Tree MSE	Tree R^2
200	.058 (.006)	.936 (.004)	11.865 (2.037)	.924 (.013)	39.053 (3.813)	.866 (.012)
300	.057 (.005)	.936 (.003)	10.447 (1.868)	.933 (.012)	33.786 (2.780)	.883 (.009)
400	.057 (.004)	.936 (.003)	10.071 (1.394)	.935 (.009)	31.518 (2.446)	.890 (.008)
500	.057 (.003)	.936 (.002)	9.599 (1.447)	.938 (.009)	29.070 (2.276)	.899 (.008)
600	.057 (.003)	.936 (.002)	9.284 (1.253)	.940 (.008)	28.143 (1.960)	.901 (.007)
700	.057 (.003)	.936 (.002)	9.466 (1.430)	.939 (.009)	26.914 (1.766)	.905 (.006)
800	.057 (.003)	.936 (.002)	9.142 (1.115)	.941 (.007)	25.732 (1.955)	.909 (.006)
900	.057 (.003)	.936 (.002)	8.864 (1.031)	.943 (.007)	24.887 (1.732)	.912 (.006)
1000	.057 (.003)	.936 (.002)	8.972 (1.091)	.942 (.007)	24.218 (1.707)	.914 (.006)

Note: IRT estimates are in a different metric than the GA and Tree estimates. MSE is mean squared error; R^2 is variance explained; IRT is item response theory; GA is genetic algorithm; N is sample size.

General conclusions

Results from these small simulation studies suggest that the psychometric item selection procedures are more consistent in selecting the same items across bootstrapped datasets than the machine learning item selection procedures. The estimation of the slope parameter in the item response model, which are used to build the short-form and the adaptive test, seems to be stable. On the other hand, there was a lot of variability in terms of which items were selected by the genetic algorithm and which item was first selected by the tree to split on. Future directions entail extending these simulations to determine the sample size at which item selection by machine learning models stabilizes.

Furthermore, the simulation results suggested that prediction accuracy of the summed score by the genetic algorithm and trees increased and was more precise as sample size increased. On the other hand, as expected, the precision accuracy of the factor score by the IRT short-form was constant across sample size (and, in turn, we expect the same findings for the computerized adaptive test). Therefore, if the true data-generating model is a psychometric item response model, it would be difficult to recover the summed score using machine learning algorithms – the estimates would depend on sample size. Future directions entail extending these simulations to understand the conditions that affect the recovery of the summed score by machine learning algorithms.