(Cheung, S. F., & Pesigan, I. J. A. (2023). *semlbci*: An R package for forming likelihood-based confidence intervals for parameter estimates, correlations, indirect effects, and other derived parameters. *Structural Equation Modeling: A Multidisciplinary Journal*. Advance online publication. https://doi.org/10.1080/10705511.2023.2183860. This is the version accepted, not the final published version..

The package is available at CRAN and can be installed: `install.package("semlbci")`

The package website at GitHub: https://sfcheung.github.io/semlbci/index.html)

***semlbci*: An R package for Forming Likelihood-Based Confidence Intervals for Parameter Estimates, Correlations, Indirect Effects, and Other Derived Parameters**

Shu Fai Cheung and Ivan Jacob Agaloos Pesigan

University of Macau

Author Note

Shu Fai Cheung (ORCiD: https://orcid.org/0000-0002-9871-9448), Department of Psychology; Ivan Jacob Agaloos Pesigan (ORCiD: https://orcid.org/0000-0003-4818-8420), Department of Psychology. This research was supported by the Multi-Year Research Grant (MYRG2020-00242-FSS) from the University of Macau, Macao SAR, China. We thank Wai Chan for his valuable comments on an earlier draft of this article. Correspondence concerning this article should be addressed to Shu Fai Cheung, Department of Psychology, Faculty of Social Sciences, University of Macau, Avenida da Universidade, Taipa, Macao SAR, China, or by email (sfcheung@um.edu.mo).

Abstract

There are three common types of confidence interval (CI) in structural equation modeling (SEM): Wald-type CI, bootstrapping CI, and likelihood-based CI (LBCI). LBCI has the following advantages: 1) it has better coverage probabilities and Type I error rate compared to Wald-type CI when the sample size is finite; 2) it correctly tests the null hypothesis of a parameter based on likelihood ratio chi-square difference test; 3) it is less computationally intensive than bootstrapping CI; and 4) it is invariant to transformations. However, LBCI is not available in many popular SEM software packages. We developed an R package, *semlbci*, for forming LBCI for parameters in models fitted by *lavaan*, a popular open-source SEM package, such that researchers have more options in forming CIs for parameters in SEM. The package supports both unstandardized and standardized estimates, derived parameters such as indirect effect, multisample models, and the robust LBCI proposed by Falk (2018).

Keywords: likelihood-based confidence interval, structural equation modeling, confidence interval, robust method

***semlbci*: An R package for Forming Likelihood-Based Confidence Intervals for Parameter Estimates, Correlations, Indirect Effects, and Other Derived Parameters**

Forming confidence intervals (CIs) is an important task in structural equation modeling (SEM). In addition to providing an interval estimate of a parameter that takes into account the sampling variation, it can also be used to test a hypothesis when a *z*-test or a *t*-test is not available, such as testing the indirect effect in a mediation model. The most popular type of CI is the Wald-type CI (WCI, Pek & Wu, 2015). Another popular type of CI is the bootstrapping CI (BCI, Efron & Hastie, 2016), which has several variants that differ in the resampling method (e.g., nonparametric bootstrapping, parametric bootstrapping) and the CI formation procedure (e.g., percentile CI and bias-corrected accelerated CI). They are available in most commonly used SEM software packages. The third type of CI, likelihood-based confidence interval (LBCI), is less popular in social sciences and available only in a limited number of SEM software packages (e.g., *OpenMx*, Neale, Hunter, Pritikin, Zahery, Brick, Kirkpatrick, Estabrook, Bates, Maes, & Boker, 2016), despite its advantages over the other two types in some situations. To give researchers the option to use LBCI, we developed *semlbci*, an R package, for forming LBCIs for parameters in models fitted by *lavaan* (Rosseel, 2012), a popular R package for SEM. We first briefly introduce LBCI, including its advantages and disadvantages. We then illustrate how to use *semlbci* to form LBCIs for parameters and functions of parameters, such as indirect effects and standardized coefficients (e.g., correlations). Last, we discuss current limitations and possible future directions for the package.

A Brief Introduction to CI, LBCI, WCI, and BCI

To illustrate the advantages of LBCI over WCI, we first present a brief introduction to CI in general. We then present two aspects of LBCI's performance that highlight its substantial difference from WCI. LBCI, WCI, and BCI have been discussed and compared in detail by others in this journal (e.g., Falk & Biesanz, 2015; Pritikin, Rappaport, & Neale, 2017; Falk, 2018). We will only give a brief introduction to them, focusing on the test inversion approach presented below when comparing LBCI and WCI. Interested readers can consult Falk (2018) for a more comprehensive introduction to WCI and LBCI in SEM, and Meeker and Escobar (1995) for a general introduction of LBCI in other contexts.

Confidence Interval as Inverting a Test

For reasons to be presented later, we first use a sample mean to illustrate how CI is formed and used. Suppose a sample of size *n* on a variable *x* is drawn from a population: $\{x_1, x_2, \dots, x_i, \dots, x_n\}$, x_i the value of the *i*th case, $\bar{x}$ the sample mean, s_x the sample standard deviation. The standard error (SE) of $\bar{x}$, $s_{\bar{x}}$, is $s_x/\sqrt{n}$. Assuming the population distribution of *x* is normal, to test whether $\bar{x}$ is significantly different from x_0, with level of significance α, two-tailed, the critical *t*, $t_0 = t_{n-1,1-\alpha/2}$, is computed, $n - 1$ the degrees of freedom (*df*) and $1 - \alpha/2$ the area to the left of t_0. If the sample *t* statistics, $t_{\bar{x}} = |\bar{x} - x_0|/s_{\bar{x}}$, is greater than t_0, $\bar{x}$ is declared to be significantly different from x_0 at *α* (two-tailed).

Instead of specifying x_0, we can also find the *interval of nonsignificance* by inverting the test (Casella & Berger, 2001): $[\bar{x} \pm t_0 s_{\bar{x}}]$. All values on the left of the interval are less than $\bar{x} - t_0 s_{\bar{x}}$, and all values on the right of the interval are greater than $\bar{x} + t_0 s_{\bar{x}}$. Equivalently, $\bar{x}$ is significantly different from all values outside the interval. If $\bar{x}$ is tested against $\bar{x} \pm t_0 s_{\bar{x}}$, the two bounds of the interval, the two-tailed *p*-values of the *t*-test are exactly α. If tested against values outside the interval, the two-tailed *p*-values are less than α. If tested against values inside the interval, the two-tailed *p*-values are greater than α. That is, $\bar{x}$ is not significantly different from all values in this interval at α, two-tailed. This interval is called the $100(1 - \alpha)\%$ confidence interval (CI) of $\bar{x}$ based on the *t*-test.

Using the Confidence Intervals: Hypothesis Testing and Interval Estimation

There are two common uses of CI. If a CI is formed by inverting a known statistical test, then the CI can be used for hypothesis testing based on this test, as shown above. Even if it is formed by methods like bootstrapping, it is still usually used this way. It is probably the most popular method for testing an indirect effect (e.g., Hayes, 2022), by checking whether zero is outside a $100(1-\alpha)\%$ CI. If yes, an indirect effect is significantly different from zero at α (two-tailed). Nevertheless, as shown above, the CI also shows the range of values from which an estimate is not significantly different at α (two-tailed), as long as the test is valid for this range of values. Using CI to test against zero is only a special case. If used this way, then the validity of a CI can be evaluated by checking the *p*-values of an estimate when tested against the two bounds of the interval. For a $100(1-\alpha)\%$ CI, the two-tailed *p*-values should be equal to α, which is necessarily the case for the *t*-based CI for sample means and CIs formed in a similar way.

Another common use of CI is as an interval estimate of a parameter. A common frequentist interpretation of the probability that a sample $100(1-\alpha)\%$ CI includes the population value of the parameter being estimated is $100(1-\alpha)\%$. Note that, for any particular sample CI, it either includes or excludes the population value. Therefore, in assessing the validity of a CI procedure, one common way is to estimate its coverage probabilities across situations and see how close they are to the expected value. However, a CI can have the expected coverage probability but cannot be used for doing a two-tailed test. For example, suppose we form the CI of a sample mean this way: $[\bar{x} - t_{(n,1-\alpha/2-.02)} s_{\bar{x}}, \bar{x} - t_{(n,1-\alpha/2+.02)} s_{\bar{x}}]$. This confidence interval is asymmetric about $\bar{x}$, with the lower bound farther away from the sample mean and the upper bound closer to the sample mean, having different Type I error rates for the two "tails." However, its coverage probability of the population value is still 95%. A CI formed this way is valid in the coverage probability sense but not valid in the hypothesis testing sense (unless different weights are placed on the tails, or the test is one-tailed). In the present paper, we focus on using CI as a tool to identify values from which a sample estimate is significantly different.

Likelihood Ratio Test and LBCI

In SEM, there are several ways to test a parameter estimate. Suppose an arbitrary structural equation model, M_1, with q parameters, $\boldsymbol{\theta}$, is fitted to the data by maximum likelihood (ML), with estimates $\hat{\boldsymbol{\theta}}$. To test whether the estimate of a parameter, $\hat{\theta}_j$, is significantly different from zero at α = .05 (two-tailed), a likelihood ratio test (LR test, a χ^2 test with *df* = 1) can be conducted by comparing M_1 to a more restricted model, M_0, with θ_j fixed to zero. This test is also called the χ^2 difference test because it is a general procedure to test the difference between two models, one nested within another, in goodness of fit. Although not common, the LR test can be used to test an estimate against other values. Moreover, the LR test can also be used to compare two models, one with an equality constraint imposed (Bollen, 1989). Therefore, the LR test can also be used to test functions of parameters, such as standardized regression coefficients, correlations, and indirect effects (e.g., Cheung, 2009a, 2009b; Falk, 2018; Falk & Biesanz, 2015; Pesigan & Cheung, 2020), comparing M_1 to M_0, with the function of relevant parameters (e.g., an indirect effect) fixed to zero in M_0.

If the LR test is appropriate, then a CI of a parameter can be formed by inverting the LR test. The CI formed this way is called likelihood-based CI (LBCI, also called profile likelihood CI in SEM, see Pritikin et al., 2017). A $100(1-\alpha)\%$ LBCI is formed by inverting the LR test to find two values, θ_{jL} and θ_{jU}, such that the LR tests when fixing the parameter to these two values have *p*-values equal to α.

The LBCI has the advantage that it can be interpreted as in the simple case of *t*-based CI for a sample mean: Values inside the interval are values from which $\hat{\theta}_j$ is not significantly different based on the LR test, while $\hat{\theta}_j$ is significantly different from all values outside the interval. If used to test against zero, the LBCI also tells exactly what a researcher wants to know when reading the *p*-value of an estimate: Whether the χ^2 difference test is significant if a path or covariance is "removed" (the parameter fixed to zero).

Wald CI and Delta-Method CI

Although LBCI is easy to interpret, two other CIs, Wald CI and delta-method CI, are much more popular in the applications of SEM. Following Pek and Wu (2015), we use Wald-Type CI (WCI) to refer to both Wald CI and delta-method CI. Most popular SEM programs report WCIs for parameters and derived parameters (functions of parameters), such as standardized coefficients and correlations. Using the example above, a 95% Wald CI is formed using the standard error of $\hat{\theta}_j$, $s_{\hat{\theta}_j}$, and the value in the standard normal distribution with $(1 - \alpha/2)$ of the area to the left, 1.96: $[\hat{\theta}_j \pm 1.96 s_{\hat{\theta}_j}]$. A 95% delta-method CI for a derived parameter, $h(\theta_*)$, θ_* being a subset of θ (or θ if all parameters are involved), is formed by $[h(\widehat{\boldsymbol{\theta}}_*) \pm 1.96 s_h]$, where $s_h = \sqrt{\dot{h}(\widehat{\boldsymbol{\theta}}_*)\hat{\Sigma}_{\boldsymbol{\theta}_*}\dot{h}(\widehat{\boldsymbol{\theta}}_*)'}$ is the approximated standard error of $h(\widehat{\boldsymbol{\theta}}_*)$, $\hat{\Sigma}_{\boldsymbol{\theta}_*}$ is the estimated sampling variances and covariances of $\widehat{\boldsymbol{\theta}}_*$, and $\dot{h}(\boldsymbol{\theta}_*) = \partial h(\boldsymbol{\theta}_*)/\partial \boldsymbol{\theta}'_*$ (Rao, 1973). Wald CI and delta-method CI are also formed by inverting a test. Wald CI inverts the Wald test (Wald, 1943), which tests whether an estimate is significantly different from θ_0 using $z_{\hat{\theta}_j} = \left|\hat{\theta}_j - \theta_0\right|/s_{\hat{\theta}_j}$ and the critical value from a standard normal distribution (1.96 when $\alpha = .05$). The case is the same for the delta-method CI, using $z_{h(\hat{\theta}_*)} = \left|h(\hat{\theta}_*) - h_0\right|/s_h$. Therefore, like LBCI, WCI is also formed by inverting a test, and so the interval are values from which a parameter is not significantly different, although the tests being inverted are different.

Comparing LBCI and WCI in a Real Data Set

Although WCI and LBCI are asymptotically equivalent (Cox & Hinkley, 1974), they can be different in finite samples, sometimes substantially. We can assess the similarity between a Wald CI and an LBCI without actually finding the LBCI, but by checking the LR test *p*-values when fixing a parameter to one of its Wald CI bounds. We use the classic data set by Holzinger and Swineford (provided in *lavaan*) as an example, fitting a three-factor model on the nine variables, *x*1 to *x*9 (Figure 1) using ML estimation (source file *cfa_pvalues.Rmd*)[1]:

Table 1 shows the LR test *p*-values when a parameter is fixed to a bound of its 95% Wald CI. The columns *p*-values are the LR test *p*-values comparing the fitted model to a model with a parameter fixed to the lower bound or upper bound of its 95% Wald CIs. None of the Wald CI bounds have *p*-values equal to .05, that is, none of the intervals coincide with the corresponding LBCIs if they are formed. Some of the *p*-values are substantially higher or lower than .05. For example, the upper bound of the factor loadings of *x*9 on speed ability is 1.378, with $p = .229$. This interval is too narrow and optimistic, excluding a wide range of values above 1.378 from which the estimate 1.082 is not significantly different based on the LR test. In sum, although formed by inverting the Wald test, the Wald CI cannot be interpreted as such if we use LR test *p*-value as the criterion.

To illustrate why a delta-method CI may also lead to incorrect conclusions on the LR test, we used as an example the data set from Tal-Or, Cohen, Tsfati, and Gunther (2010, available in the *psych* package),[2] which is a popular dataset for illustrating simple mediation (e.g., Hayes, 2022). They conducted an experiment to investigate whether a treatment would influence reaction to a news story through presumed media influence (PMI). A simple mediation model is fitted, with condition (where the stimulus was said to be presented in a newspaper: front page = 1, interior page = 0) as the independent variable, PMI as the mediator, and reaction as the outcome variable. The indirect effect (*ab*) is the product of the path coefficient from condition to presumed media influence (*a*) and that from PMI to

[1] All source and output files are available at the OSF project for this manuscript: https://osf.io/b9a2p/.
[2] We thank the corresponding author for the permission to use this dataset for illustration.

reaction (*b*). The ML estimate is 0.241, and the 95% delta-method CI (default of *lavaan*) is -.007 to .490. The indirect effect is not significantly different from zero by this CI. However, if this model is compared to a model with the indirect effect fixed to zero, the p-value of the LR test is .043. That is, the indirect effect is actually significantly different from zero using the LR test (source file *Tal_Or_pvalue.Rmd*).

In sum, WCI can yield conclusions different from those by the LR test. A WCI can include or exclude values from which the estimate is not significantly based on the LR test, and the significant test conclusions by WCI and LR test can be different.

Why LBCI Should Be Used (If Available) Instead of WCI

WCI as an Approximation of LBCI

One may argue that Wald CI is a correct CI if we trust the Wald test because it is formed by inverting the Wald test. If we test an estimate against the bounds of a 95% Wald CI, then the Wald test p-values are necessarily .05. However, Pawitan (2001) argued that, if LBCI and Wald CI are not similar, as in the examples above, LBCI is preferred, partly because Wald CI and delta-method CI can be considered approximations of the LBCI.

We use the Holzinger-Swineford data set for illustration again. Suppose we want to form the CI for the covariance between visual ability and speed ability, $\widehat{\sigma}_{VS}$. When estimated by ML, the log-likelihood function, $\log L(\boldsymbol{\theta})$, is maximized when evaluated at $\widehat{\boldsymbol{\theta}}$, the ML estimates (MLEs) of all free parameters, and $\widehat{\sigma}_{VS} = 0.262$ (Table 1). Following the suggestion by Pawitan (2001), we scale the log-likelihood to be zero at MLEs when visualizing the log-likelihood. We then plot the log-likelihood by fixing $\widehat{\sigma}_{VS}$ to values near its MLE and maximize the log-likelihood with respect to other parameters (Figure 2), $\log L(\sigma_{VS}) = \log L\left(\sigma_{VS}, \boldsymbol{\theta}_q\right)$, where $\boldsymbol{\theta}_q$ are other parameters in $\boldsymbol{\theta}$ (Pawitan, 2001). This is called the log profile likelihood of σ_{VS} (source file *cfa.Rmd*).

The solid blue line is the plot of the log profile likelihood of σ_{VS} from 0.162 to 0.382, peaking at 0.262 which is the MLE of σ_{VS}. The decrease in log profile likelihood is 1.92 at these two points, half of 3.84, the critical value of χ^2 with *df* = 1. The LR test statistic when testing σ_{VS} against an arbitrary value σ'_{VS} is given by $2[\log L(\hat{\boldsymbol{\theta}}) - \log L(\sigma'_{VS})]$. Therefore, the *p*-values of the LR tests fixing σ_{VS} to 0.162 or 0.382 is .05. In other words, the estimated covariance of 0.262 is just significant from 0.162 and 0.382, not significantly different from 0.162 to 0.382, and significantly different from values less than 0.162 or greater than 0.382. This interval is the 95% LBCI of $\hat{\sigma}_{VS}$. Note that the log profile likelihood is curved and slightly asymmetric for σ_{VS} in this example.

Instead of fitting the model many times to plot the log profile likelihood in a range of values, a quadratic approximation of the log profile likelihood of σ_{VS} can also be formed by $\log L(\sigma_{VS}) \approx \log L(\hat{\boldsymbol{\theta}}) - 0.5(s^2_{\hat{\sigma}_{VS}})^{-1}(\sigma_{VS} - \hat{\sigma}_{VS})^2$ using Taylor's expansion, where $s^2_{\hat{\sigma}_{VS}}$ is the diagonal element corresponding to σ_{VS} in the inverse of Fisher information matrix $I(\hat{\boldsymbol{\theta}})$, and $s_{\hat{\sigma}_{VS}}$ is the standard error of $\hat{\sigma}_{VS}$ (Pawitan, 2001). In practice, whether the expected or observed information is used depends on the data and other factors (e.g., whether missing data is present or not, see Savalei, 2010). In the above example, with complete data, the expected information matrix is used, the default option in *lavaan*. The LR test is conducted by using $2[\log L(\hat{\boldsymbol{\theta}}) - \log L(\sigma'_{VS})]$ and a 1-*df* χ^2 distribution. Therefore, $(s^2_{\hat{\sigma}_{VS}})^{-1}(\sigma_{VS} - \hat{\sigma}_{VS})^2$ is the approximated test statistic, and the test is equivalent to finding the *p*-value of $|\hat{\sigma}_{VS} - \sigma_{VS}|/s_{\hat{\sigma}_{VS}}$ in a standard normal distribution, which is the Wald test used to form the Wald CI. The dotted red line in Figure 2 is a plot of this quadratic approximation, and the interval at log profile likelihood = -1.92, 0.152 to 0.373, is the Wald CI of $\hat{\sigma}_{VS}$.

Numerically, the two intervals may look similar. However, their differences are in the *p*-values of the bounds. The LR test *p*-values of the LBCI is .05 at its bounds because it is formed by inverting the LR test. This is not necessarily the case for Wald CI because Wald test is an approximation of the LR test. In this example, the Wald CI is shifted to the left. Based on the LR test, the sample estimate $\widehat{\sigma}_{VS}$ is not significantly different from some values within the left end of the Wald CI (from 0.152 to 0.162) and some values outside the right end of it (from 0.373 to 0.382). Different from how we want a CI behaves, we cannot say that $\widehat{\sigma}_{VS}$ is not significantly from all values within the Wald CI based on the LR test. Although the conclusion is the same if we only want to test whether $\widehat{\sigma}_{VS}$ is significantly different from zero, when one bound of a Wald CI is close to zero, which is not uncommon, LBCI and Wald CI may lead to different conclusions because one interval includes zero while the other does not. In this case, LBCI, if available, should be used to test a null hypothesis.

The same phenomenon applies to delta-method CI because it can also be treated as an approximation of the log profile likelihood when a function of parameters, such as a correlation, is fixed to a value (Pawitan, 2001). The log profile likelihood of the indirect effect and its quadratic approximation in Tar-Or et al. (2010) are plotted in Figure 3 (source file *Tal_Or_no_boot.Rmd*). The asymmetry of the log profile likelihood is more apparent, with the LBCI wider but did not include zero, showing that the delta-method CI, the interval defined by the blue dotted line, may not approximate the log profile likelihood well, leading to a conclusion different from that by the LR test and LBCI.

WCI is Not Invariant to Transformation and Reparameterization, while LBCI is

Another problem with WCI and Wald test is their dependence on transformation and reparameterization (Gregory & Veall, 1985). For example, Gonzalez and Griffin (2001) showed that the Wald test p-value of a factor covariance may depend on which loading of the indicators of a factor is fixed to one. Different ways to formulate a hypothesis can result in different p-values for the Wald test, and consequently different WCI because it is formed by inverting the Wald test. The LR test and LBCI are invariant to reparameterization and transformation (Pawitan, 2001).

Comparing LBCI with BCI

Although not used as the default CI, a popular CI method for some parameters, such as derived parameters like indirect effects presented above, is bootstrapping CI (BCI). There are many variants, and we focus on nonparametric percentile bootstrapping (NPCI), a common method for forming CI for an indirect effect (e.g., in the PROCESS macro by Hayes, 2022). In NPCI for SEM, for a sample of size n, B bootstrap samples of the same size are drawn with replacement. The original model is then fitted to each of the bootstrap samples, and the parameter of concern is estimated in each bootstrap sample, yielding B bootstrap estimates. The $100(1-\alpha)\%$ NPCI is formed by finding the $100(\alpha/2)^{\text{th}}$ and $100(1-\alpha/2)^{\text{th}}$ percentiles of the bootstrap estimates. NPCI is usually used when (a) the distributional assumption of the estimators (e.g., multivariate normality for ML) may be violated, or (b) the sampling distribution of a parameter is unknown or complicated.

NPCI is similar to LBCI because both methods do not assume symmetry in the uncertainty about the point estimate, making them appropriate choices for derived parameters such as indirect effect and standardized parameters like correlations. NPCI has the added advantage that it makes no distributional assumption on the raw data. LBCI is valid only if the likelihood function is not misspecified, and so multivariate normality is still needed to use LBCI for ML estimates (but see Falk, 2018, on a robust version of LBCI, discussed later).

Despite the advantages of NPCI over LBCI, it has two disadvantages. First, it is computationally intensive, requiring fitting a model a large number of times (B must be large, at least 2000 while 5000 is common). Although repeated model fitting is also required in forming LBCI, as implied in the plot of log profile likelihood above, it usually requires much less computational time. Second, NPCI is a resampling method and so the CI will change as the set of bootstrap samples changes, adding one more level of sampling variability. This variability can be decreased but at the cost of increasing B. Moreover, if a bound is close to zero, a large *B* is required to achieve results that depend little on the set of bootstrap samples. In the Tal-Or et al. example above, we formed NPCI for the indirect effect 100 times, with *B* = 5000, 10000, and 20000 (source file *Tal_Or_boot.Rmd*). When *B* = 5000, 9% of the NPCIs included 0. When *B* increased to 10000, 3% of the NPCIs included 0. All NPCIs agree (do not include 0) when *B* is 20000. In practice, researchers may stop at *B* = 5000, not being aware that a larger number of *B* is needed. LBCI, though without a closed-form solution, is fixed and there is no additional source of uncertainty due to resampling, unlike NPCI, , other than what is due to optimization.

Robust WCI and Robust LBCI

When the assumption of multivariate normality is not tenable, several robust versions of WCI (R-WCI) are available. The principle is similar in all common approaches: the estimated variance-covariance matrix is adjusted for potential deviation from multivariate normality and the adjusted variance-covariance matrix is used to form the WCI (see Savalei, 2014, for an overview). This approach cannot be used for LBCI because it is not computed from the variance-covariance matrix. Falk (2018) proposed a simple approach to from a robust LBCI. Because LBCI is formed by inverting the LR test, robust LBCI can be formed by inverting the scaled χ^2 test developed by Satorra (2000). This test is readily available in *lavaan* through *lavTestLRTs()* with *method* set to *"satorra.2000"*. Therefore, the $100(1-\alpha)\%$ robust LBCI of a parameter is found by finding the two values that will result in Satorra-2000 LR test p-values equal to α if the parameter is fixed to one of these two values. The procedure is the same for derived parameters such as standardized coefficients and indirect effects. Falk empirically compared robust LBCI with other methods across a wide variety of conditions and found that robust LBCI is a viable CI procedure when violation of multivariate normality is suspected. In *semlbci*, this approach is adopted to form a robust LBCI (details on the implementation presented later).

Why LBCI is Rarely Used in SEM in Some Disciplines

We believe there are three major reasons that LBCI is rarely used in the applications of SEM in some disciplines. First, researchers may believe that LBCI and WCI are similar. We demonstrated that this is not the case even in a popular dataset. Second, when WCI is not suitable, researchers may resort to bootstrapping CI. Bootstrapping CI indeed performed well in many situations (e.g., Falk, 2018). However, LBCI is also a viable alternative in these situations, especially when the computational cost required to reduce the resampling error in bootstrapping CI is too great to use it repeatedly except in the final stage of a series of analyses. The third major reason is the lack of tools. LBCI is not available in most of the popular SEM software packages. *OpenMx* (Neale et al., 2016) has good support for the LBCI, and users can request them for virtually any parameters. However, at the time of writing, *Amos* (Arbuckle, 2021), *lavaan* (Rosseel, 2012), and *Mplus* (Muthén & Muthén, 2017) do not support the LBCI. Users need to find them manually (e.g., Asparouhov, 2020). To overcome this problem and let more researchers have the option to use LBCI, we developed *semlbci*, an R package for forming LBCI for parameters in a model fitted by SEM in *lavaan*. We first illustrate how to use *semlbci* which is relevant to most users, and then discuss major technical details on the implementation for interested readers. The package can be downloaded from the OSF page for this manuscript (https://osf.io/b9a2p/) or installed from GitHub (https://sfcheung.github.io/semlbci/).

How to Use *semlbci* to Form LBCI

Workflow

The package was designed to allow users to form the LBCI for model parameters without learning a new package to do SEM. It currently supports *lavaan*, one of the most popular SEM packages in R (R Core Team, 2022). The workflow is simple: (1) Researchers fit their models, as usual, using *lavaan*, and (2) pass the fit object from *lavaan* to *semlbci* and specify parameters for which LBCI are to be formed. The output is similar to the output of *parameterEstimates()* for original estimates and *standardizedSolution()* for standardized estimates, to reduce the need to learn reading a new output format. In the following sections, we illustrate how *semlbci* can be used in different scenarios. The source files and PDF files of the output of all the illustrations can be found in the folder *examples* of the OSF page (https://osf.io/b9a2p/files/osfstorage).

Free Parameters and Derived Parameters in a Simple Mediation Model

A simple mediation model is fitted to the dataset *simple_med*, included in *semlbci*:

```
fit <- sem(model = mod, data = simple_med, fixed.x = FALSE)
```

This is an excerpt of the original parameter estimates results from *lavaan*:

```
lhs op rhs label   est    se      z pvalue ci.lower ci.upper
  m  ~   x     a 1.676 0.431  3.891  0.000    0.832    2.520
  y  ~   m     b 0.470 0.074  6.354  0.000    0.325    0.616
  y  ~   x       1.540 0.468  3.291  0.001    0.623    2.457
 ab := a*b    ab 0.789 0.238  3.318  0.001    0.323    1.254
```

To form the LBCIs for all free parameters and the indirect effect *ab*, call the function *semlbci()* (source file *med.Rmd*):

```
fit_lbci <- semlbci(fit)
```

By default, LBCIs are formed for all free parameters and derived parameters, except for variances and error variances because these CIs are rarely reported, and researchers rarely test whether they are significant or not. To include them, add *remove_variances = FALSE*. The output is a *semlbci*-class object. This is an excerpt of the default printout:

```
lhs op rhs label   est lbci_lb lbci_ub    lb    ub cl_lb cl_ub
  m  ~   x     a 1.676   0.828   2.525 0.832 2.520 0.950 0.950
  y  ~   m     b 0.470   0.325   0.616 0.325 0.616 0.950 0.950
  y  ~   x       1.540   0.618   2.461 0.623 2.457 0.950 0.950
 ab := a*b    ab 0.789   0.370   1.313 0.323 1.254 0.950 0.950
```

The LBCIs are [*lbci_lb*, *lbci_ub*]. For comparison, the original CIs are also printed (Wald CIs for free parameters and delta-method CI for the indirect effect). The columns *cl_lb* and *cl_ub* are the *achieved levels of confidence of the bounds* (the LR test *p*-value if a parameter is fixed to a bound), which should be close to the level of confidence of the intervals (.95 for a 95% confidence interval). As expected, LBCIs and the original CIs are close to each other for the path coefficients, suggesting that the quadratic approximation is good. However, the lower bound of the indirect effect (.370) is closer to the point estimate and farther away from zero. As shown in the supplementary file with full output (*med.html*), if *ab* is fixed to the lower bound of LBCI (.370), the LR test *p*-value is .05. If fixed to the lower bound of Wald CI (.323), the *p*-value is .028, showing that the lower bound is farther away from the point estimate than it should be.

Standardized Estimates

LBCIs for parameter estimates in the standardized solution, such as standardized path coefficients, correlations, and standardized indirect effects can also be formed using *semlbci*. This is an excerpt of the standardized solution results:

```
lhs op rhs label est.std    se      z pvalue ci.lower ci.upper
  m  ~   x     a   0.265 0.066  4.035  0.000    0.136    0.394
  y  ~   m     b   0.403 0.059  6.863  0.000    0.288    0.518
  y  ~   x         0.209 0.062  3.346  0.001    0.087    0.331
 ab := a*b    ab   0.107 0.031  3.463  0.001    0.046    0.168
```

Note that the CIs in the standardized solution is delta-method CIs and are symmetric around the point estimates, including the standardized indirect effect. To form the LBCIs for the standardized solution, just add *standardized = TRUE*:

```
fit_lbci_std <- semlbci(fit, standardized = TRUE)
```

This is an excerpt of the output:

```
lhs op rhs id label est.std lbci_lb lbci_ub    lb    ub cl_lb cl_ub
  m  ~   x  1     a   0.265   0.132   0.389 0.136 0.394 0.950 0.950
  y  ~   m  2     b   0.403   0.283   0.512 0.288 0.518 0.950 0.950
  y  ~   x  3         0.209   0.084   0.328 0.087 0.331 0.950 0.950
 ab := a*b  7    ab   0.107   0.051   0.173 0.046 0.168 0.950 0.950
```

The sampling distribution of the standardized solution is not expected to be symmetric, and more so for the standardized indirect effect. Delta-method CIs are symmetric for all the standardized estimates, while the LBCIs take into account possible asymmetry. The lower bound of the LBCI of the indirect effect (.051) is again farther away from zero and closer to the point estimate, though the magnitude is smaller on the standardized metric. As in the case of the original estimates, all the bounds achieved the desired level of confidence. That is, if a standardized parameter is fixed to this bound, the LR test *p*-value is .05.

A Confirmatory Factor Analysis (CFA) Model

The function *semlbci()* can also be used to form the LBCIs for factor loadings and factor covariances in a CFA model. We use the Holzinger-Swineford dataset as an example again. First, fit the model as usual:

```
mod <-'visual  =~ x1 + x2 + x3
       textual =~ x4 + x5 + x6
       speed   =~ x7 + x8 + x9'
fit <- cfa(model = mod, data = HolzingerSwineford1939)
```

To form the LBCIs for all free parameters except for variances and error variances, simply pass the fit object to *semlbci()* as before. This model has 21 free parameters, and the search can be slow. To speed up the search, researchers can add *parallel = TRUE* to enable parallel processing, and use *ncpus* to set the number of cores to use (source file *cfa.Rmd*):

```
fit_lbci <- semlbci(fit, parallel = TRUE, ncpus = 6)
```

Running on a system with Intel Core i7-8700 with this setting took about 16 seconds. This is an excerpt of the results:

```
    lhs op     rhs   est lbci_lb lbci_ub    lb    ub ratio_l ratio_u
 visual =~      x2 0.554   0.356   0.793 0.358 0.749   1.013   1.224
 visual =~      x3 0.729   0.520   0.996 0.516 0.943   0.977   1.245
textual =~      x5 1.113   0.992   1.249 0.985 1.241   0.946   1.062
textual =~      x6 0.926   0.821   1.044 0.817 1.035   0.966   1.082
  speed =~      x8 1.180   0.923   1.536 0.857 1.503   0.795   1.101
  speed =~      x9 1.082   0.782   1.655 0.785 1.378   1.011   1.934
 visual ~~ textual 0.408   0.262   0.577 0.264 0.552   1.017   1.173
 visual ~~   speed 0.262   0.162   0.382 0.152 0.373   0.909   1.090
textual ~~   speed 0.173   0.083   0.281 0.077 0.270   0.932   1.112
```

The LBCIs for some of the loadings are substantially different from Wald CIs. If the ratio of the distance from the point estimate to an LBCI bound to that from the point estimate to the corresponding original CI bound is larger than a threshold (1.5 by default), two new columns will be displayed, *ratio_l* and *ratio_u*. The ratio is 1.934 for the upper bound of the factor loading of x9, (1.655 - 1.082) / (1.378 - 1.082) ≈ 1.934. As shown in Figure 4, the quadratic approximation is poor for this factor loading. The Wald CI is too narrow.

Standardized Factor Loadings and Factor Correlations

To form the LBCIs for standardized factor loadings and factor correlations, simply add *standardized = TRUE*. However, the search for the LBCIs of standardized estimates can be slow for a model with many free parameters. Therefore, parallel processing is recommended, or the LBCIs are only formed for parameters with which the magnitudes of the standardized estimates will be interpreted.

```
fit_lbci_std <- semlbci(fit, parallel = TRUE, ncpus = 6, standardized = TRUE)
```

Running on a system with Intel Core i7-8700 took about 13 seconds. Note that variances and error variances are removed by default. Therefore, only the LBCIs for 12 parameters (9 standardized factor loadings and 3 factor correlations) will be formed. This is the output for the factor correlations:

```
    lhs op     rhs id est.std lbci_lb lbci_ub    lb    ub cl_lb cl_ub
 visual ~~ textual 22   0.459   0.326   0.575 0.334 0.584 0.950 0.950
 visual ~~   speed 23   0.471   0.300   0.633 0.328 0.613 0.950 0.950
textual ~~   speed 24   0.283   0.139   0.418 0.148 0.418 0.950 0.950
```

The LBCIs for the three factor correlations are close to the delta-method CIs except for that of visual-speed correlation. The LBCI of visual-speed correlation is wider than the delta-method CI. Again, the LBCIs correctly reflect the expected asymmetry of the distribution of correlations, with the bounds away from zero closer to the point estimates, and the other bounds closer to zero.

Latent Level Mediation

Functions of parameters are also supported for a model with latent variables. We use the sample dataset *mediation_latent*, provided with *semlbci*, for illustration. It has nine variables loaded on three factors: *x*1 to *x*3 on *fx, x*4 to *x*6 on *fm*, and *x*7 to *x*9 on *fy*. The effect of *fx* on *fy* is mediated through *fm*, with a direct path from *fx* to *fy* included:

```
mod <- 'fx =~ x1 + x2 + x3
        fm =~ x4 + x5 + x6
        fy =~ x7 + x8 + x9
        fm ~ a * fx
        fy ~ b * fm + cp * fx
        ab := a*b'
fit <- sem(model = mod, data = mediation_latent)
```

The call to find the LBCI of the indirect effect is similar to that in previous examples.

To specify a derived parameter, use the same syntax in *lavaan* but omit the definition after ":=". The standardized latent indirect effect can be found by adding *standardized = TRUE* (source file *med_lav.Rmd*):

```
fit_lbci <- semlbci(fit, pars = "ab := ")
fit_lbci_std <- semlbci(fit, pars = "ab := ", standardized = TRUE)
```

These are the results:

```
Unstandardized
 lhs op rhs label     est lbci_lb lbci_ub    lb    ub cl_lb cl_ub
  ab := a*b    ab   0.115   0.016   0.247 0.007 0.222 0.950 0.950
Standardized
 lhs op rhs label est.std lbci_lb lbci_ub    lb    ub cl_lb cl_ub
  ab := a*b    ab   0.115   0.017   0.227 0.015 0.214 0.950 0.950
```

The latent indirect effect is .115, with 95% delta-method CI .007 to .222 and 95% LBCI .016 to .247. The standardized indirect effect is also .115, with 95% delta-method CI .015 to .214 and 95% LBCI .017 to .227. The log profile likelihood functions (Figure 5) are asymmetric as expected, with the delta-method CIs too narrow compared to the LBCI it approximates.

For comparison, 95% NPCIs were also formed, with 5000 bootstrap samples. The 95% NPCI of the latent indirect effect is .008 to .255, wider than both delta-method CI and LBCI. The 95% NPCI of the standardized latent indirect effect is .009 to 241, again wider than both other CIs. The sample dataset was generated from a multivariate normal distribution and so the LBCI based on ML is valid and preferred in this example.

A Multisample Model

Multisample models with or without between-group equality constraints are also supported. We use the Holzinger and Swineford dataset again as an example and use school as the grouping variable:

```
mod <- 'visual  =~ x1 + c(lambda2, lambda2)*x2 + c(lambda3, lambda3)*x3
        textual =~ x4 + c(lambda5, lambda5)*x5 + c(lambda6, lambda6)*x6
        speed   =~ x7 + c(lambda8, lambda8)*x8 + c(lambda9, lambda9)*x9'
fit <- cfa(model = mod, data = HolzingerSwineford1939,
           group = "school")
```

The factor loadings are constrained to be equal across groups and so LBCIs can be formed for only one set of them. To do this, we can specify the parameters for which LBCIs will be formed using the *pars* argument. Users can form a vector of strings using lavaan model syntax, and "multiply" the right-hand side by the group number:

```
free_loadings <- c("visual  =~ 1*x2", "visual  =~ 1*x3",
                   "textual =~ 1*x5", "textual =~ 1*x6",
                   "speed   =~ 1*x8", "speed   =~ 1*x9")
```

Alternatively, they can be specified as a usual lavaan model:

```
free_loadings <- "visual  =~ 1*x2 + 1*x3
                  textual =~ 1*x5 + 1*x6
                  speed   =~ 1*x8 + 1*x9"
```

The first approach is easier to read while the second approach is more compact.

```
fit_lbci_loadings <- semlbci(fit, pars = free_loadings, parallel = TRUE, ncpus = 6)
```

Running on a system with Intel Core i7-8700 took about 34 seconds. These are part of the results:

```
    lhs op rhs group   est lbci_lb lbci_ub    lb    ub cl_lb cl_ub
 visual =~  x2     1 0.599   0.396   0.847 0.402 0.795 0.950 0.950
 visual =~  x3     1 0.784   0.573   1.064 0.573 0.996 0.950 0.950
textual =~  x5     1 1.083   0.958   1.225 0.951 1.215 0.950 0.950
textual =~  x6     1 0.912   0.802   1.037 0.798 1.025 0.950 0.950
  speed =~  x8     1 1.201   0.953   1.536 0.897 1.506 0.950 0.950
  speed =~  x9     1 1.038   0.771   1.494 0.771 1.304 0.950 0.950
```

The achieved level of confidence is .95 for all confidence bounds.

Standardized Factor Correlations

Forming the LBCIs for parameters in the standardized solution for a multisample model with equality constraints can be substantially slower. Therefore, they can be formed just for parameters that will be interpreted, such as the factor correlations. Again, to select the factor correlations, we can use lavaan model syntax. For a multisample model, if the group number is not specified, then LBCIs will be formed for the parameter in all groups.

```
fit_lbci_fcor <- semlbci(fit,
  pars = c("visual ~~ textual", "visual ~~ speed", "textual ~~ speed"),
  parallel = TRUE, ncpus = 6, standardized = TRUE)
```

Running on a system with Intel Core i7-8700 took about 50 seconds. These are parts of the results:

```
    lhs op     rhs group id est.std lbci_lb lbci_ub    lb    ub cl_lb cl_ub
 visual ~~ textual     1 22   0.485   0.291   0.640 0.315 0.654 0.950 0.950
 visual ~~   speed     1 23   0.340   0.097   0.565 0.118 0.563 0.950 0.950
```

```
textual ~~   speed     1 24   0.333   0.127   0.519 0.138 0.529 0.950 0.950
 visual ~~ textual     2 58   0.540   0.357   0.692 0.373 0.708 0.950 0.950
 visual ~~   speed     2 59   0.536   0.319   0.725 0.352 0.719 0.950 0.950
textual ~~   speed     2 60   0.345   0.143   0.524 0.166 0.523 0.950 0.950
```

The sampling distribution of correlations is expected to be skewed toward zero. LBCIs correctly reflect this. All the lower bounds of LBCIs of the factor correlations are closer to zero, and less biased than the overoptimistic delta-method CIs which assume a symmetric distribution.

Robust LBCI

Robust LBCI using the method proposed by Falk (2018) is also supported in *semlbci*. We used the latent level mediation model again but used the dataset *mediation_latent_skewed* provided with *semlbci*. The variables were generated from a population with indicator error terms that are exponentially distributed. The latent level mediation model is fitted using MLR as the estimator in *lavaan* (source file *med_lav_nnorm.Rmd*):

```
fit <- cfa(model = mod, data = mediation_latent_skewed, estimator = "MLR")
```

To form the robust LBCIs, add *robust = "satorra.2000".* Currently, only the method proposed by Falk (2018) using the robust LR test by Satorra (2000) is supported.

```
fit_lbci <- semlbci(fit, pars = "ab := ", robust = "satorra.2000")
fit_lbci_std <- semlbci(fit, pars = "ab := ", robust = "satorra.2000",
                        standardized = TRUE)
```

This is an excerpt of the output:

```
Unstandadized:
  lhs op rhs label   est lbci_lb lbci_ub     lb    ub cl_lb cl_ub       robust
   ab := a*b    ab 0.057   0.004   0.128 -0.002 0.116 0.950 0.950 satorra.2000
Standardized:
lhs op rhs label est.std lbci_lb lbci_ub     lb    ub cl_lb cl_ub       robust
 ab := a*b    ab   0.071   0.009   0.146  0.005 0.138 0.950 0.950 satorra.2000
```

Note that the columns *lb* and *ub* are the confidence intervals based on robust standard errors in the original output. All columns are interpreted as before, except that the columns *lbci_lb* and *lbci_ub* are the bounds of robust LBCIs, and *cl_lb* and *cl_ub* are one minus the *p*-values of the χ^2 difference test by Satorra (2000) when fixing the parameters to their confidence bounds. Robust LBCI is also supported for all previous scenarios, such as multisample models and path models with only observed variables.

For comparison, 95% NPCIs were also formed for the latent indirect effect. The 95% NPCI of the unstandardized indirect effect is .008 to .138, and that of the standardized indirect effect is .010 to .152. Both LBCIs and NPCIs are close to each other and yield the same conclusion for both unstandardized and standardized latent indirect effects. However, ran on Intel Core i7-8700 with six cores, NPCIs took about 96 seconds while LBCI only took about six seconds with two cores (one for each bound). The delta-method CIs based on the robust variance-covariance matrix, on the other hand, are narrower than both LBCIs and NPCIs, and suggest the unstandardized latent indirect effect is nonsignificant.

Implementation

Algorithms

The 95% LBCI can be formed manually in any SEM software package by fixing a parameter or a function of parameters to different values until two values, one smaller and one larger than the point estimates are found that yield 1-*df* LR test *p*-values of .05. The challenge is to do this efficiently and automatically. As reviewed by Pek and Wu (2015), there are several algorithms to form the LBCI. In SEM, the algorithm proposed by Neale and Miller (1997) is a popular one, used as the default in *OpenMx* and in the simulation studies by Falk (2018). Another algorithm was proposed by Wu and Neale (2012), also implemented in *OpenMx*. Although Wu and Neale originally proposed the algorithm to form LBCI for bounded parameters (free parameters with attainable bounds), we did not include their steps for bounded parameters and only adopted the general algorithm illustrated by Pek and Wu (Equation 12), which is applicable to other parameters and functions of parameters. We call this algorithm the WNPW algorithm.

When developing an LBCI package for *lavaan*, we adopted two goals. First, we wanted to use only information exported by *lavaan* whenever possible, such that it is more likely to be compatible with future versions of *lavaan*. Second, we wanted to use as many functions exported by *lavaan* as possible, to ensure that the results are consistent with those by *lavaan*. This is because one of the checks of a bound is to check the *p*-value of the LR test between the original model and a model with the target parameter or function of parameters constrained to this bound, and this test is to be conducted by the *lavaan*. These two goals also allow for possible expansion to other SEM packages in the future if they export similar information and functions. We tried the approach by Neale and Miller (1997) in an earlier version of *semlbci*. However, we found that it is easier to program using the WNPW algorithm if we want to achieve these two goals. Therefore, we developed the package based on this algorithm (see the Online Appendix of Pek & Wu, 2015, for an example of this algorithm in *OpenMx*).

The implementation of robust LBCI was inspired by the work of Falk (https://github.com/falkcarl/lavaan). Different from his work, we developed a separate package instead of proposing changes in *lavaan*. Therefore, we adopted Falk's application of Satorra's (2000) LR test to form robust LBCI but implemented it using Pek and Wu's (2015) version of Wu and Neale (2012). For optimization, the package *nloptr* (Ypma et al., 2022, an R interface to NLopt by Johnson) was adopted, using the SLSQP algorithm (Kraft, 1994) as the default algorithm. We present briefly the implementation below. The full technical details can be found in technical appendices at the OSF project page for this manuscript (https://osf.io/b9a2p/).

LBCI

To find the lower bound θ_L of the $100(1-\alpha)\%$ LBCI of $\hat{\theta}_j$, θ_L is minimized with respect to all parameters $(\theta_L, \boldsymbol{\theta}_q)$, $\boldsymbol{\theta}_q$ being all parameters other than θ_L, subject to the constraint $F(\theta_L, \boldsymbol{\theta}_q) = F(\hat{\boldsymbol{\theta}}) + \chi^2_{1,1-\alpha}/2n$, where $F(\hat{\boldsymbol{\theta}})$ is the value of the discrepancy function evaluated at ML estimate $\hat{\boldsymbol{\theta}}$, $\chi^2_{1,1-\alpha}$ is the χ^2 critical value at *df* = 1 and level of significance = α (about 3.84 with α = .05), and n is the sample size (total sample size in multisample models). To find the upper bound, θ_U, $-\theta_U$ is minimized (equivalently, θ_U is maximized). We used 2n is because it is how the model χ^2 is computed from the value of the discrepancy function in *lavaan* (the output of *lavaan::fitMeasures()* with *what = "fmin"*).[3] This algorithm has two advantages. First, it is easy to program (Pek & Wu, 2015). As long as an SEM program returns the discrepancy function value for user-supplied estimates, any optimization function that allows for equality constraints can be used to find the confidence bound. As an example, our implementation is a generalization of the examples in *OpenMx* by Pek and Wu to *lavaan*. Second, it can be extended to a function of parameters $h(\boldsymbol{\theta})$, such as a standardized path coefficient or a correlation, by minimizing $h(\boldsymbol{\theta})$ with the aforementioned constraint (note that the derived parameter may depend only on some elements in $\boldsymbol{\theta}$). For a model with equality constraints, they are simply included as additional equality constraints in the optimization.

Robust LBCI

Robust LBCI is formed similarly but the Satorra (2000) LR test is used as proposed by Falk (2018), which is natively supported in *lavaan*. The principle is the same: search for the 95% LBCI bounds for a parameter by finding values to which the Satorra (2000) LR test p-values are .05 when fixing the parameter to these bounds, without the need to know how to adjust the χ^2. The procedure proposed by Falk can be readily generalized to multisample models in *lavaan* because *lavaan* supports doing the Satorra LR test for multisample models, and *semlbci* uses this test to form robust LBCIs and check their validity. For a multisample model with m samples, when constraining a function of all parameters, $h_0(\boldsymbol{\theta})$, to a value in the more restricted model, the scaling factor can be estimated by (Satorra, 2000, Equation

[3] This is not the case if the argument *likelihood* is set to *"wishart"*. In the current version, *semlbci()* will not run on a model fitted with *likelihood* set to *"wishart"*.

23)[4]

$$\hat{c} = \mathrm{tr}(\mathbf{M}),$$

$$\mathbf{M} = (\mathbf{W}\boldsymbol{\Gamma}^{*}\mathbf{W})\left\{\boldsymbol{\Delta}\boldsymbol{P}^{-1}\dot{h_0}(\widehat{\boldsymbol{\theta}})'\left(\dot{h_0}(\widehat{\boldsymbol{\theta}})\boldsymbol{P}^{-1}\dot{h_0}(\widehat{\boldsymbol{\theta}})'\right)^{-1}\dot{h_0}(\widehat{\boldsymbol{\theta}})P^{-1}\boldsymbol{\Delta}'\right\},$$

$$\boldsymbol{\Gamma}^{*} = \begin{pmatrix} f_1\boldsymbol{\Gamma}_1^{*} & & & & \\ & \ddots & & & \\ & & f_j\boldsymbol{\Gamma}_j^{*} & & \\ & & & \ddots & \\ & & & & f_m\boldsymbol{\Gamma}_m^{*} \end{pmatrix},$$

$$\mathbf{W} = \begin{pmatrix} f_1\mathbf{W}_1 & & & & \\ & \ddots & & & \\ & & f_j\mathbf{W}_j & & \\ & & & \ddots & \\ & & & & f_m\mathbf{W}_m \end{pmatrix},$$

$$\boldsymbol{\Delta} = \begin{pmatrix} \boldsymbol{\Delta}_1 \\ \vdots \\ \boldsymbol{\Delta}_j \\ \vdots \\ \boldsymbol{\Delta}_m \end{pmatrix}$$

where $f_j = n_j/n$, n_j is the sample size of the *j*th sample, *n* is the total sample size, $\mathbf{W}_j$ is the weight matrix (the inverse of the estimated asymptotic covariance matrix of sample statistics assuming multivariate normality) in the *j*th sample, $\boldsymbol{\Gamma}_j^{*}$ is the asymptotic distribution-free estimate of the covariance matrix of the sample statistics in the *j*th sample (Browne, 1984), $\boldsymbol{\Delta}_j = \partial \boldsymbol{s}_j/\partial\boldsymbol{\theta}'$ evaluated at $\widehat{\boldsymbol{\theta}}$, $\boldsymbol{s}_j$ is the sample statistics in the *j*th sample, $\dot{h_0}(\widehat{\boldsymbol{\theta}}) = \partial h_0(\boldsymbol{\theta})/\partial\boldsymbol{\theta}'$ evaluated at $\widehat{\boldsymbol{\theta}}$, and $\boldsymbol{P} = \boldsymbol{\Delta}'\mathbf{W}\boldsymbol{\Delta}$ (Satorra, 2000, p. 239). For one-sample models, $\hat{c}$ is equivalent to Equation 10 in Falk (2018) with one equality constraint. When finding the LBCI for a free parameter θ_j, $h_0(\boldsymbol{\theta}) = \theta_j$. When finding the LBCI for a derived parameter $h(\boldsymbol{\theta})$, $h_0(\boldsymbol{\theta}) = h(\boldsymbol{\theta})$. The *lavTestLRT()* in *lavaan* uses the method proposed by Asparouhov and Muthén (2010) when Satorra (2000) LR test is used. Therefore, the critical value is

[4] This is how Satorra's (2000) method is implemented in *lavTestLRT()* of *lavaan* (version 0.6-13).

adjusted to ${\chi^*}^2_{1,1-\alpha} = a^{-1}\left(\chi^2_{1,1-\alpha} - b\right)$, where $a = 1/\mathrm{tr}(\mathbf{M}^2)$ and $b = 1 - \mathrm{tr}(\mathbf{M})/\mathrm{tr}(\mathbf{M}^2)$, $\mathbf{M}$ depends only on the fitted model and $h_0(\boldsymbol{\theta})$.[5] After the adjusted critical value is computed, the confidence bound can be found as described above with the critical χ^2 in the constraint replaced by ${\chi^*}^2_{1,1-\alpha}$. The validity of a bound of robust LBCI can be verified by doing an LR test using *lavTestLRT()* with *method = "satorra.2000"* and *A.method = "exact"* (because the two models are necessarily nested in the parameter sense).

Validity Checks

Because a closed-form solution for LBCI is not available, the bound found needs to be checked for reasonability (Pritikin et al., 2017). In *semlbci*, the following three checks are conducted for each bound found before returning the results:

1. The optimization function *nloptr()* returns a code of "0" (success).
2. The values of the free parameters at the solution ($\boldsymbol{\theta}$) do not result in an inadmissible solution as defined by the SEM function, *lavaan* in *semlbci*. This is done by *lavaan::lavInspect()* with *what = "post.check"*).
3. The *p*-value of the LR test between the original model and a model with the target parameter constrained to the bound is close enough to 1 – level of confidence of the requested interval (.05 for a 95% LBCI) within a tolerance (default is 0.0005)[6].

A bound certainly should pass the third check to be a valid bound of LBCI because this is the defining characteristic of a bound of LBCI. However, to be conservative, only a bound that passes *all three checks* will be returned. If a bound found in the first attempt fails any of these checks, *semlbci()* will try "harder" several times using different settings (e.g., different tolerance values for convergence, different starting values). If it still fails at least one of the checks, after a certain number of attempts, *NA* (not available) will be returned. We

[5] In future versions of *semlbci*, we can use *lavaan* to compute *a* and *b* directly because they are exported by *lavTestLRT()* in the latest version of *lavaan* at the time of writing (0.6-13).

[6] This value can be changed by users for higher precision in the search.

cannot "prove" that a bound passing all checks is valid, which is impossible due to the nature of LBCI (Meeker & Escobar, 1995). Nevertheless, by imposing these checks, we believe it can at least reject bounds that are suspicious.

In addition to these checks, based on the suggestion by Pritikin et al., 2017, *semlbci* also computes the ratio of the distance of a bound from the point estimate to the distance of the corresponding Wald CI or delta-method CI from the point estimate, to detect whether a bound is too far or too close to the point estimate compared to the Wald CI bound. If the ratio is at 1.5 or above or 1 / 1.5 or below (the default[7]), the bound will still be returned but a note will be included in the printout, alerting the users to check the bound for plausibility.

Comparing Results with *OpenMx*

Although the algorithms used are not new and the definitional validity of an LBCI can be checked by the LR test, we also conducted a small-scale simulation study to compare the LBCIs by *semlbci* with those by *OpenMx*, to check whether *semlbci* works as expected. A simple mediation model identical to that in the previous example was used to generate the data, with *a* path and *b* path equal and *c'* path fixed to zero. Four levels of values for the *a* path and *b* path were examined: .00, .10, .30, and .50. The sample sizes examined were 100 and 300. Each condition had 2000 replications. The LBCIs of *semlbci* and *OpenMx* for the *a* path (unstandardized and standardized), the indirect effect (*ab*, unstandardized and standardized), and the error variance of *y* were compared. They yield LBCIs with negligible differences across all the replications and have similar coverage probabilities across conditions. The full details of the simulation study can be found at the folder *simulation* at the OSF page for this manuscript (https://osf.io/b9a2p/files/osfstorage).

When To Use LBCIs?

Despite the advantages of LBCIs, we are not advocating the use of LBCIs

[7] This ratio can be changed by users.

unconditionally because (a) the WCIs can approximate the likelihood function very well in some cases, and (b) forming LBCIs can be difficult in some models and for some parameters due to the lack of a closed-form solution. Although the LBCIs could be formed quickly in the previous examples, we encountered cases in which the search took more than one minute or even failed. Therefore, we suggest running *semlbci()* with the default settings first and parallel processing enabled. If LBCIs can be formed for all free parameters (error variances and variances excluded by default), then LBCIs should be reported instead of WCIs because WCIs are approximations of the LBCIs. If it takes too long to run or the search for confidence bounds fails for some parameters even after tweaking the options of the optimizer, then researchers can use *semlbci()* to form LBCIs only for major parameters for which asymmetry in the likelihood function is suspected, such as standardized path coefficients, indirect effects, and correlations, because this is when the quadratic approximation by WCI is likely to be poor. If necessary, researchers can tweak some technical options in the search (described in the help documents and the technical appendices). However, this is usually not necessary because *semlbci()* internally will try adjusting some options, including randomizing the start values, if the initial search fails. The easiest option is to set *try_k_more_times* to an integer higher than 2 (the default value), telling the function to do more attempts.

If researchers deemed that bootstrapping CIs are more appropriate for a parameter or a derived parameter, the LBCIs or robust LBCIs can still be used as proxies if the computation cost of bootstrapping CIs is high. During the model selection and comparison stage, LBCIs can be used for hypothesis testing and interval estimates. Once the final model has been selected, NPCIs can be formed for selected parameters for the final results.

A Cautionary Note on Using LBCIs

Despite the advantages of LBCIs over WCIs, like all confidence interval methods, their performance such as coverage probabilities cannot be taken for granted. As found by

Falk (2018), even percentile bootstrap CI, which outperformed many methods investigated, LBCI included, may still perform poorly in some conditions. Therefore, our goal is to provide one more option for researchers to form CIs for parameter estimates. Researchers still need to justify their choice of a method, ideally based on previous empirical findings. Note that this also holds for bootstrapping. Bootstrapping CIs can be formed for virtually any parameter estimates easily (e.g., using *bootstrapLavaan()* in the *lavaan* package) but its appropriateness cannot be taken for granted (Yung & Bentler, 1996). With *semlbci*, researchers interested in comparing the performance of LBCI with other methods can have one more option in the choice of SEM package.

Limitations and Future Development

The LBCI, formed by inverting the LR test, is only "as good as" the LR test. First, we took a defensive approach in developing *semlbci* and the current version will only accept a model fitted by ML, generalized least squares (GLS), or asymptotic distribution-free (ADF) method (called WLS in *lavaan*) (including their variants in *lavaan*, such as MLR, MLM, etc.) with missing data. Second, multilevel models are also currently not supported. Development is ongoing to support multilevel models. Last, *semlbci* does not yet support the adjustment for bounded parameters proposed by Wu and Neale (2012) when a point estimate is close to a boundary. This should not be a problem in most typical cases but may be relevant when correlations close to one or variances close to zero are possible. In the current version, the algorithm should fail because a valid bound cannot be found without failing the checks and so *NA* should be returned. Support for point estimates close to the boundary will be included in the future.

References

Arbuckle, J. L. (2021). Amos (Version 26.0) [Computer Program]. IBM SPSS.

Asparouhov, T. (2020, June 3). *Multi-level confidence intervals* [Discussion post]. Retrieved from http://www.statmodel.com/discussion/messages/12/26884.html?1591227383

Asparouhov, T., & Muthén, B. O. (2010). Simple second order chi-square correction. Retrieved from https://www.statmodel.com/download/WLSMV_new_chi21.pdf

Bollen, K. A. (1989). *Structural equations with latent variables*. Wiley-Interscience.

Browne, M. W. (1984). Asymptotically distribution-free methods for the analysis of covariance structures. *British Journal of Mathematical and Statistical Psychology*, *37*(1084), 62–83. https://doi.org/10.1111/j.2044-8317.1984.tb00789.x

Casella, G., & Berger, R. L. (2001). *Statistical Inference* (2nd Ed.). Cengage Learning.

Cheung, M. W.-L. (2009a). Comparison of methods for constructing confidence intervals of standardized indirect effects. *Behavior Research Methods, 41*(2), 425–438. https://doi.org/10.3758/BRM.41.2.425

Cheung, M. W.-L. (2009b). Constructing approximate confidence intervals for parameters with structural equation models. *Structural Equation Modeling: A Multidisciplinary Journal, 16*(2), 267–294. https://doi.org/10.1080/10705510902751291

Cox, D. R., & Hinkley, D. V. (1974). *Theoretical statistics*. Chapman and Hall/CRC.

Efron, B., & Hastie, T. (2016). *Computer age statistical inference: Algorithms, evidence, and data science*. Cambridge University Press.

Falk, C. F. (2018). Are robust standard errors the best approach for interval estimation with nonnormal data in structural equation modeling? *Structural Equation Modeling: A Multidisciplinary Journal, 25*(2), 244–266. https://doi.org/10.1080/10705511.2017.1367254

Falk, C. F., & Biesanz, J. C. (2015). Inference and interval estimation methods for indirect effects with latent variable models. *Structural Equation Modeling: A Multidisciplinary Journal, 22*(1), 24–38. https://doi.org/10.1080/10705511.2014.935266

Gonzalez, R., & Griffin, D. (2001). Testing parameters in structural equation modeling: Every "one" matters. *Psychological Methods, 6*(3), 258–269. https://doi.org/10.1037//1082-989X.6.3.258

Gregory, A. W., & Veall, M. R. (1985). Formulating Wald tests of nonlinear restrictions. *Econometrica, 53*(6), 1465–1468. https://doi.org/10.2307/1913221

Hayes, A. F. (2022). *Introduction to mediation, moderation, and conditional process analysis: A regression-based approach* (3rd Ed.). The Guilford Press.

Johnson, S. G. (n.d.) The NLopt nonlinear-optimization package. Retrieved on Jan 10, 2023: http://github.com/stevengj/nlopt

Kraft, D. (1994). Algorithm 733: TOMP–Fortran modules for optimal control calculations. *ACM Transactions on Mathematical Software, 20*(3), 262–281. https://doi.org/10.1145/192115.192124

Meeker, W. Q., & Escobar, L. A. (1995). Teaching about approximate confidence regions based on maximum likelihood estimation. *The American Statistician, 49*(1), 48–53. https://doi.org/10.2307/2684811

Muthén, L. K., & Muthén, B. O. (2017). *Mplus: Statistical analysis with latent variables: User's guide* (Version 8).

Neale, M. C., Hunter, M. D., Pritikin, J. N., Zahery, M., Brick, T. R., Kirkpatrick, R. M., Estabrook, R., Bates, T. C., Maes, H. H., & Boker, S. M. (2016). OpenMx 2.0: Extended structural equation and statistical modeling. *Psychometrika, 81*(2), 535–549. https://doi.org/10.1007/s11336-014-9435-8

Pawitan, Y. (2001). *In all likelihood: Statistical modelling and inference using likelihood*.

Oxford University Press.

Pek, J., & Wu, H. (2015). Profile likelihood-based confidence intervals and regions for structural equation models. *Psychometrika, 80*(4), 1123–1145. https://doi.org/10.1007/s11336-015-9461-1

Pesigan, I. J. A., & Cheung, S. F. (2020). SEM-based methods to form confidence intervals for indirect effect: Still applicable given nonnormality, under certain conditions. *Frontiers in Psychology, 11*. https://doi.org/10.3389/fpsyg.2020.571928

Pritikin, J. N., Rappaport, L. M., & Neale, M. C. (2017). Likelihood-based confidence intervals for a parameter with an upper or lower bound. *Structural Equation Modeling: A Multidisciplinary Journal, 24*(3), 395–401. https://doi.org/10.1080/10705511.2016.1275969

Rao, C. R. (1973). *Linear statistical inference and its applications* (2nd Ed.). Wiley-Interscience.

R Core Team (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from https://www.R-project.org/

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*(2), 1–36. https://www.jstatsoft.org/article/view/v048i02

Satorra, A. (2000). Scaled and adjusted restricted tests in multi sample analysis of moment structures. In R. D. H. Heijmans, D. S. G. Pollock, & A. Satorra (Eds.), *Innovations in Multivariate Statistical Analysis. Advanced Studies in Theoretical and Applied Econometrics* (Vol. 36). Springer.

Savalei, V. (2010). Expected versus observed information in SEM with incomplete normal and nonnormal data. *Psychological Methods, 15*(4), 352–367. https://doi.org/10.1037/a0020143

Savalei, V. (2014). Understanding robust corrections in structural equation modeling.

Structural Equation Modeling: A Multidisciplinary Journal, 21(1), 149–160. https://doi.org/10.1080/10705511.2013.824793

Tal-Or, N., Cohen, J., Tsfati, Y., & Gunther, A. C. (2010). Testing causal direction in the influence of presumed media influence. *Communication Research, 37*(6), 801–824. https://doi.org/10.1177/0093650210362684

Wald, A. (1943). Tests of Statistical Hypotheses Concerning Several Parameters When the Number of Observations is Large. T*ransactions of the American Mathematical Society, 54*(3), 426–482. https://doi.org/10.2307/1990256

Wu, H., & Neale, M. C. (2012). Adjusted confidence intervals for a bounded parameter. Behavior Genetics, 42(6), 886–898. https://doi.org/10.1007/s10519-012-9560-z

Ypma, J., Johnson, S. G. Borchers, H. W., Eddelbuettel, D, Ripley, B, Hornik, K., Chiquet, J., Adler, A., Dai, X., Stamm, A., & Ooms, J. (2022). *nloptr*: R Interface to NLopt. R package version 2.0.3, URL http://CRAN.R-project.org/package=nloptr

Yung, Y.-F., & Bentler, P. M. (1996). Bootstrapping techniques in analysis of mean and covariance structures. In G. A. Marcoulides & R. E. Schumacker (Eds.), *Advanced structural equation modeling: Issues and techniques* (pp. 195–226). Psychology Press.

Table 1.

Wald Confidence Bounds and the Likelihood-Ratio p-Values at the Bounds

Parameters	Estimate	Lower Bound	Upper Bound	*p*-value (Lower Bound)	*p*-value (Upper Bound)
Factor Loadings					
*x*2 on visual	0.554	0.358	0.749	.053	.102
*x*3 on visual	0.729	0.516	0.943	.045	.105
*x*5 on textual	1.113	0.985	1.241	.038	.064
*x*6 on textual	0.926	0.817	1.035	.042	.069
*x*8 on speed	1.180	0.857	1.503	.011	.070
*x*9 on speed	1.082	0.785	1.378	.053	.229
Factor Covariances					
visual with textual	0.408	0.264	0.552	.054	.091
visual with speed	0.262	0.152	0.373	.030	.070
textual with speed	0.173	0.077	0.270	.035	.075

Note. The *p*-values are the χ^2 difference test *p*-values comparing the fitted model to a model with a parameter fixed to its bound (upper or lower).

Figure 1

A Sample Confirmatory Factor Analysis Model Fitted to the Holzinger and Swineford Dataset

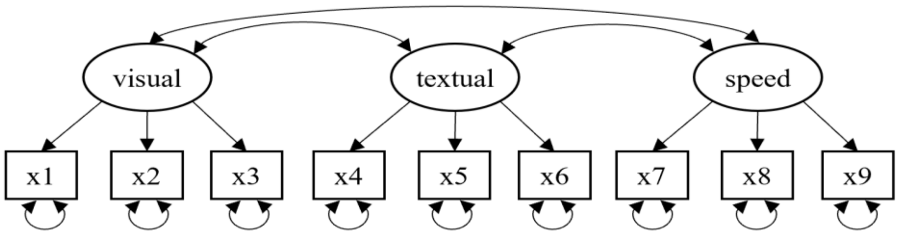
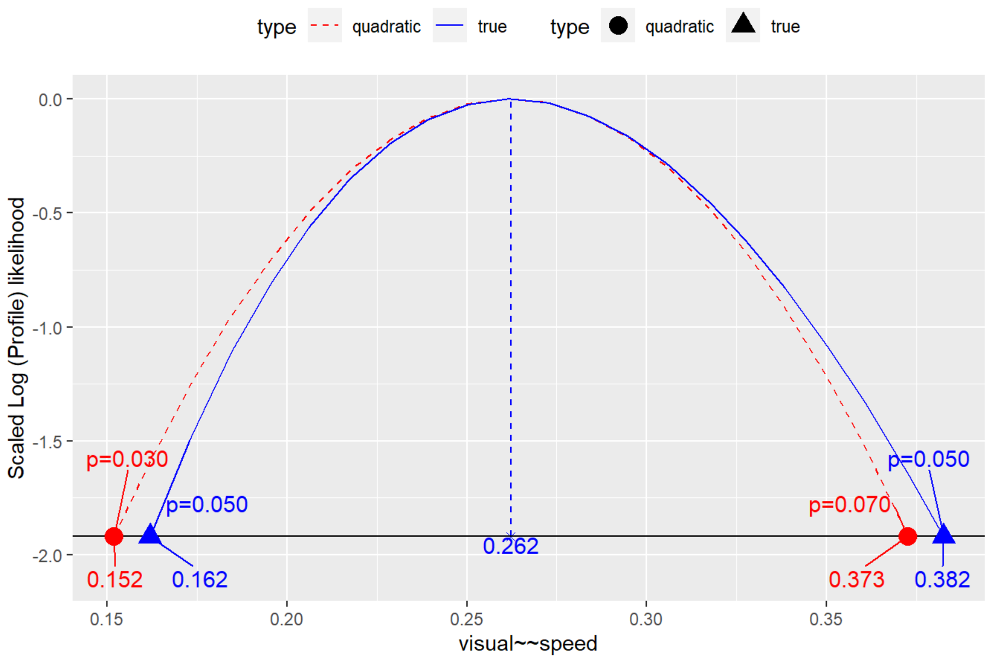


Figure 2

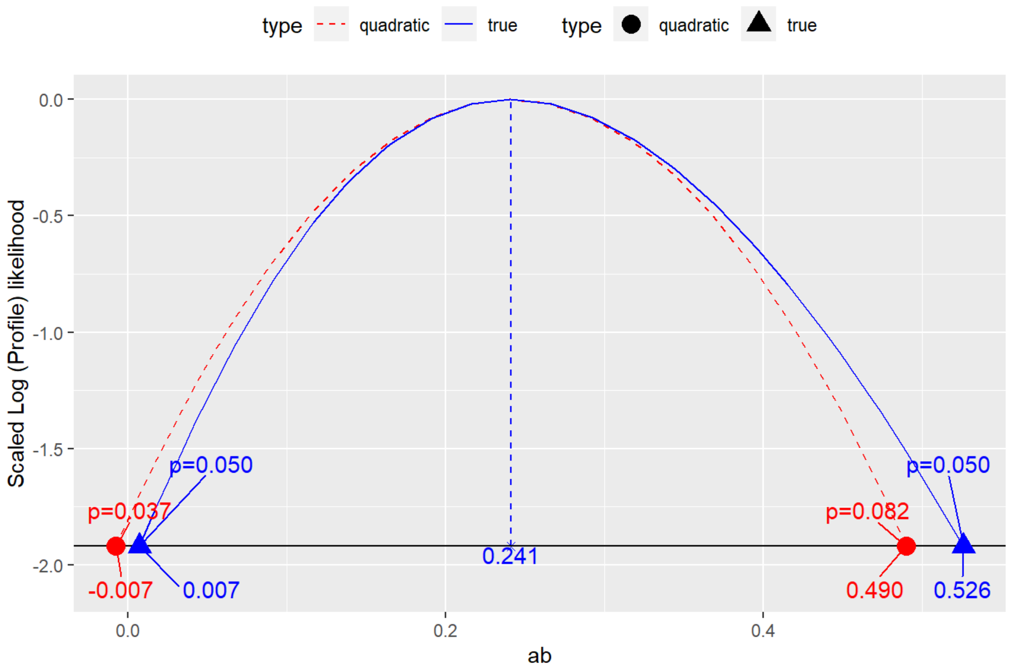
Log Profile Likelihood and Its Quadratic Approximation: Covariance between Visual Ability and Speed Ability in the Confirmatory Factor Analysis Model Fitted to Holzinger-Swineford Dataset



Note. The blue solid line is the plot of log profile likelihood of the covariance between visual and speed in the confirmatory factor analysis on the Holzinger-Swineford dataset. The red dotted line is the plot of log profile likelihood using quadratic approximation. The value when the log profile loglikelihood is zero is the maximum likelihood estimate. The log profile loglikelihood is -1.92 at the horizontal solid line. The two blue triangles form the 95% likelihood-based confidence interval, and the two red circles form the 95% Wald-type confidence interval. The *p*-values are the χ^2 difference test *p*-values when the parameter is fixed to the corresponding values.

Figure 3

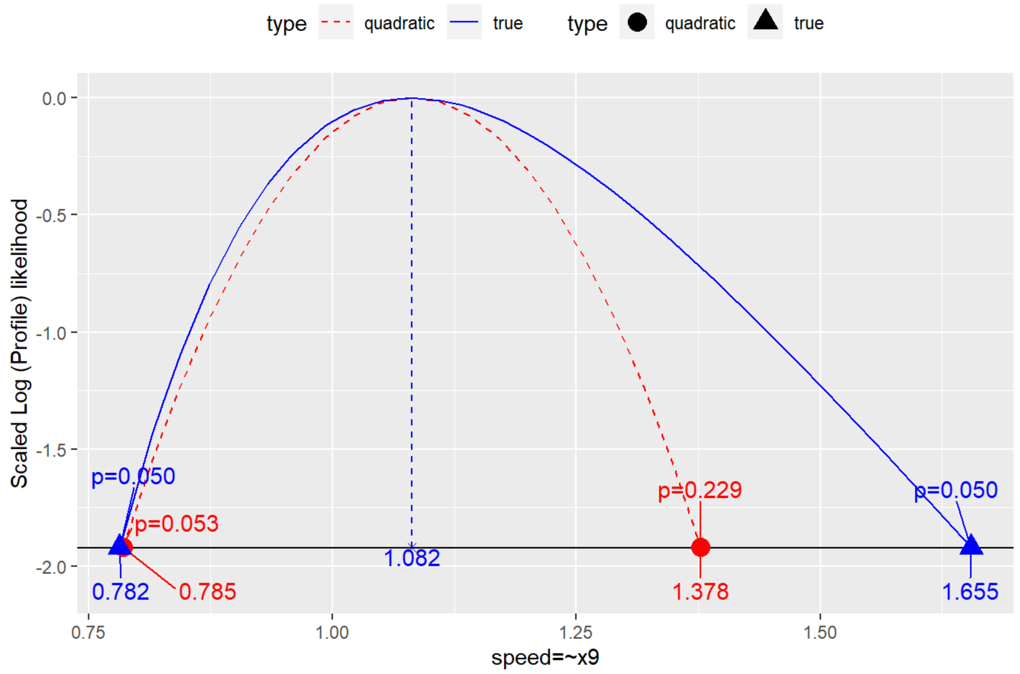
Log Profile Likelihood and Its Quadratic Approximation: Indirect Effect (ab) in the Simple Mediation Model Fitted to the Tar-Or et al. (2010) Dataset



Note. The blue solid line is the plot of log profile likelihood of the indirect effect (*ab*) in the simple mediation model fitted to the Tar-Or et al. (2010) dataset. The red dotted line is the plot of log profile likelihood using quadratic approximation. The value when the log profile loglikelihood is zero is the maximum likelihood estimate. The log profile loglikelihood is -1.92 at the horizontal solid line. The two blue triangles form the 95% likelihood-based confidence interval, and the two red circles form the 95% Wald-type confidence interval. The *p*-values are the χ^2 difference test *p*-values when the parameter is fixed to the corresponding values.

Figure 4

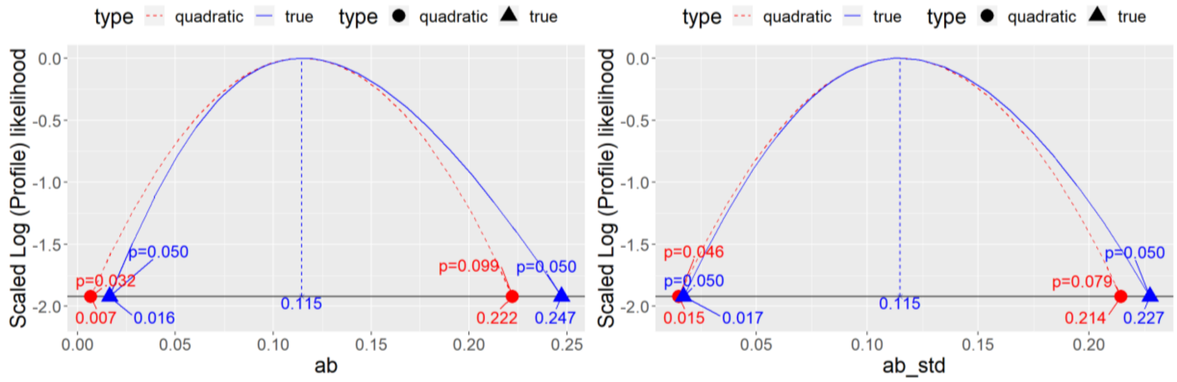
Log Profile Likelihood and Its Quadratic Approximation: Factor Loading of x9 on Speed Ability in the Confirmatory Factor Analysis Model Fitted to Holzinger-Swineford Dataset



Note. The blue solid line is the plot of log profile likelihood of the factor loading of *x*9 on speed ability in the confirmatory factor analysis on the Holzinger-Swineford dataset. The red dotted line is the plot of log profile likelihood using quadratic approximation. The value when the log profile loglikelihood is zero is the maximum likelihood estimate. The log profile loglikelihood is -1.92 at the horizontal solid line. The two blue triangles form the 95% likelihood-based confidence interval, and the two red circles form the 95% Wald-type confidence interval. The *p*-values are the χ^2 difference test *p*-values when the parameter is fixed to the corresponding values.

Figure 5

Log Profile Likelihood and Its Quadratic Approximation: Unstandardized (ab) and Standardized (ab_std) Latent Indirect Effect in the Latent Mediation Model Fitted to the mediation_latent Dataset



Note. The blue solid lines are the plot of log profile likelihoods of the unstandardized (*ab*) and standardized (*ab_std*) latent indirect effect in the latent mediation model fitted to the *mediation_latent* dataset. The red dotted lines are the plots of log profile likelihoods using quadratic approximation. The values when the log profile loglikelihoods are zeros are the maximum likelihood estimates. The log profile loglikelihood is -1.92 at the horizontal solid lines. The two blue triangles in each panel form the 95% likelihood-based confidence interval, and the two red circles in each panel form the 95% Wald-type confidence interval. The *p*-values are the χ^2 difference test *p*-values when a parameter is fixed to the corresponding values.