WinLTA
USER'S GUIDE

VERSION 3.0

Linda M. Collins
Stephanie T. Lanza
Joseph L. Schafer
Brian P. Flaherty

The Methodology Center
The Pennsylvania State University

May 2002

Development of this program was supported by National Institute on Drug Abuse grant P50 DA10075.

Thanks to Katherine Hames, Mildred Maldonado-Molina, Zhiqun Tang, Chi-Ming Kam, and David Wagstaff for help with the preparation of this manual.

TABLE OF CONTENTS

INTRODUCTION

The WinLTA software described in this manual can be used to fit latent class and latent transition models to data. Latent class analysis (LCA; Clogg & Goodman, 1984; Dayton & Macready, 1976; Goodman, 1974; Lazarsfeld & Henry, 1968) is a measurement theory based on the idea of a static, i.e. unchanging, discrete latent variable that divides a population into mutually exclusive and exhaustive latent classes. Categorical manifest items, often dichotomous, serve as indicators of the latent variable. Latent transition analysis (LTA; Collins & Wugalter, 1992; Graham, Collins, Wugalter, Chung, & Hansen, 1991; Langeheine & van de Pol, 1991; Collins, Graham, Rousculp, & Hansen, 1997; Collins, Schafer, Lanza, & Flaherty, in preparation) is a special case of LCA where the latent variable is dynamic, i.e. changing in systematic ways over time (Collins & Cliff, 1990). LTA is a type of latent Markov model (van de Pol & Langeheine, 1989). Models tested using this program may include both static and dynamic latent variables.

Latent class and latent transition models are unfamiliar to many people. We have found that the biggest problem many new users encounter when they use WinLTA is confusion about how to translate their research ideas into a latent class or latent transition framework. We have also found that reading a few articles or technical reports describing applications of latent class and latent transition models to empirical data is very helpful to users. The reference list at the end of this manual contains some recommended readings.

LATENT CLASS MODELS

Both LCA and LTA models answer research questions involving a discrete latent variable. In latent class models, the latent variable is static or unchanging (at least for purposes of the study) and therefore typically measured at a single occasion. This latent variable divides a population into a set of mutually exclusive and exhaustive latent classes. Example 1 in the Appendix illustrates a latent class analysis. The data were obtained from a random sample of 1,500 individuals from the 1980 High School and Beyond sophomore cohort (Rock & Pollack-Ohls, 1987). A math skills test was administered to these students in their sophomore year. Rock and Pollack-Ohls (1987) derived four "testlets" from the test. Each five-item "testlet" was a mastery test corresponding to one of four domains of math skill: (1) single operations on whole numbers; (2) powers and roots, decimals, and fractions; (3) low level algebra without word problems; (4) low level geometry, algebra with word problems. If four out of the five items in a testlet were answered correctly the student was considered to have passed the testlet and mastered the subject matter (see Rock & Pollack-Ohls, 1987; Collins & Cliff, 1990).

Suppose we believe that math skill acquisition is cumulative. For example, someone who can do low level algebra can also handle powers and roots and single operations on whole numbers. This suggests a model with five latent classes: (1) those with no measurable skill; (2) those who can only do single operations on whole numbers; (3) those who can do powers and roots, decimals, and fractions, and also single operations on whole numbers; (4) those who can do low level algebra without word problems, as well as powers and roots and single operations on whole numbers; and (5) those who have mastered low level geometry and algebra with word

problems, as well as all the other skills measured. The parameter estimates resulting from testing this model appear in Sections 12 - 14 of the output for Example 1 in the Appendix. [Throughout the examples, Sections of the output are denoted by boldface numbers along the left-hand side of the column.]

LCA involves two types of parameters. The first type of parameter, the ρ parameters, appear in Sections 12 and 13 of the output for Example 1. These parameters represent the probability of a particular response to a manifest variable, conditioned on latent class membership. In the matrix appearing in Section 12, the five rows correspond to latent classes and the four columns correspond to the manifest variables. In this example, the manifest variables are the testlets. The entire matrix corresponds to Response Category 1, which in these data is a "fail." The value in the first row, first column of the output in Section 12 is ρ=.986. This means that given membership in the first latent class, here labeled No Skill, the probability is .986 of failing the first testlet, single operations on whole numbers.

The ρ's play two roles in LCA. First, they map the manifest items onto the latent classes in much the same way that factor loadings map variables onto factors. Thus, the ρ parameters are used to interpret a latent class in the same manner that factor loadings are used to interpret a latent factor. For example, in Section 12 of the output for Example 1, the first line of ρ's indicates that for the first latent class, the probability of failing any of the testlets is high. Thus, this latent status can be interpreted as a "No Skill" latent status. In contrast, for members of the third latent class, the probability of passing Testlets 1 and 2 is high, but the probability of passing Testlets 3 and 4 is low. Latent Class 3 can be interpreted as "Powers and roots, decimals, and fractions plus single operations on whole numbers." In this example, we used the WinLTA option to specify labels for the latent classes. Whenever this option is selected, it is important to examine the estimated values of the ρ parameters to verify that the user-specified labels make sense. The second role that the ρ's play in LCA is as indications of how close the correspondence is between each manifest item and each latent class. If the ρ's are close to zero or one, this indicates that the manifest responses are largely determined by latent class membership.

The second type of parameter, denoted γ, represents the proportion of the population of interest that are expected to be members of a particular latent class. As shown in Section 14 of Example 1, our model suggests that 24.8 percent of the students belong to the No Skill latent class, 29.9 percent belong to the Single Operations latent class, etc.

Mathematical Model

In this section, the LCA model is presented formally. For ease of exposition, the LCA model is expressed for problems involving three manifest indicators (items or variables); the extension to fewer than or more than three indicators is direct. Suppose Item 1 has $i = 1,...I$ response categories; Item 2 has $j = 1,...J$ response categories; Item 3 has $k = 1,...K$ response categories; and there are $c = 1,...C$ latent classes. Let $y = \{i,j,k\}$ represent a "response pattern," a vector of possible responses to the three items. Then, the proportion of individuals contributing a particular response pattern, $P(Y=y)$, can be expressed as follows:

$$P(Y = y) = \sum_{c=1}^{C} \gamma_c \rho_{i|c} \rho_{j|c} \rho_{k|c}$$

where

γ_c represents the proportion of the population in latent class c;

$\rho_{i|c}$ represents the probability of response i to Item 1, conditional on membership in latent status c; $\rho_{j|c}$ represents the probability of response j to Item 2, conditional on membership in latent status c, and $\rho_{k|c}$ represents the probability of response k to Item 3, conditional on membership in latent status c.

LATENT TRANSITION ANALYSIS

As described above, LCA models are for static, i.e. unchanging, latent variables measured at a single time. In contrast, LTA models are for stage-sequential dynamic latent variables that have been measured in a longitudinal panel design. Example 2 in the Appendix illustrates an LTA model that involves the data used in Example 1, including data that were collected when the 1,500 students were sophomores and then seniors in high school. Suppose we are interested in testing a model of math skill development over time. The model involves the following stages: individuals first learn single operations on whole numbers; then progress to powers and roots, decimals, and fractions; then learn low level algebra without word problems; then go on to low level geometry and algebra with word problems. In LTA each stage is called a latent status. Further suppose that in our model, it is possible to remain in the current stage or to advance, but not to decline. When an individual progresses to the next stage, all skills learned in earlier stages are retained.

The output in Example 2, Sections 13-17, shows the parameter estimates for this problem. In this example, three sets of parameters are estimated: ρ, δ, and τ parameters. First, the ρ estimates appear in Sections 13-15. They have the same meaning in LTA models as they do in LCA; that is, they represent the probability of a particular item response, conditional on latent status membership. They are used to interpret the characteristics of each latent status. The estimates in Section 14 indicate that for individuals in the first latent status, the probability of passing any of the testlets is low; thus, it makes sense to label the first latent status as a "no skills" latent status. Students in the second latent status have a high probability of passing the first testlet and a low probability of passing the other testlets; thus, it makes sense to label this latent status "single operations on whole numbers."

The second set of parameters, appearing in Section 16, are the δ parameters. These are estimates of the proportion of the population in each latent status at each measurement occasion.

For example, 44.2 percent are in the No Skill latent status at the outset, 18.0 percent are in the Single Operations latent status at the outset, etc. (Only the Time 1 δ parameters are actually estimated. Those for all subsequent times do not have to be estimated - instead, the program computes them from other parameter estimates.)

The third set of parameter estimates, the τ's, are transition probabilities. These estimates appear in Section 17. For a first-order model, these parameters represent the probability of being in a particular latent status at Time 2 (the senior year), conditional on latent status membership at Time 1 (the sophomore year). For example, Section 17 shows that the probability of being in the No Skill latent status in the senior year given that the individual was in the No Skill latent status as a sophomore is .855. (Second-order models stipulate that the probability of latent status membership at Time t is conditional not only on latent status membership at Time t-1, but on latent status membership at Time t-2 as well. Example 4 in the Appendix illustrates a second-order model.)

Mathematical Model

This section presents the LTA model formally. For ease of exposition, the latent transition model will be presented for problems involving three occasions of measurement, three manifest indicators of the dynamic latent variable at each occasion, and one exogenous static latent variable measured by one manifest indicator. The extension to other problems is direct. Suppose the first occasion of measurement is Time t, the second Time t+1, and the third Time t+2. Also suppose the three manifest indicators are Item 1, with i,i',i''=1,...I response categories; Item 2, with j,j',j''=1,...J response categories, and Item 3, with k,k',k''=1,...K response categories, where i, j, and k refer to responses obtained at Time t, i',j', and k' refer to responses obtained at Time t+1, and i'',j'', and k'' refer to responses obtained at Time t+2. The exogenous static latent variable divides the population into latent classes c = 1,...C, and is measured by a manifest indicator with m = 1,...M response categories. There are p, q, r = 1,...S latent statuses, with p denoting a latent status at Time t, q denoting a latent status at Time t+1, and r denoting a latent status at Time t+2.

Let $y = \{m,i,j,k,i',j',k',i'',j'',k''\}$ represent a "response pattern", a vector of possible categorical responses made up of a single response to the manifest indicator of the variable and exogenous responses to the three items at Times t, t+1, and t+2. Then, for a first-order model, the proportion of individuals making a particular response pattern, $P(Y=y)$, is expressed as follows:

$$P(Y = y) = \sum_{c=1}^{C} \sum_{p=1}^{S} \sum_{q=1}^{S} \sum_{r=1}^{S}$$

$$\gamma_c \rho_{m|c} \delta_{p|c} \rho_{i|p,c} \rho_{j|p,c} \rho_{k|p,c} \tau_{q|p,c} \rho_{i'|q,c} \rho_{j'|q,c} \rho_{k'|q,c} \tau_{r|q,c} \rho_{i''|r,c} \rho_{j''|r,c} \rho_{k''|r,c}$$

where

γ_c represents the proportion in latent class c;

$\delta_{p|c}$ represents the proportion in latent status p at Time t conditional on membership in latent class c; that is, the proportion of latent class c members whose latent status is p at Time t;

$\tau_{q|p,c}$ is an element of the latent transition probability matrix, representing the probability of membership in latent status q at Time t+1 conditional on membership in latent status p at Time t and membership in latent class c; that is, the proportion of those in latent class c and latent status p at Time t who are in latent status q at Time t+1;

$\rho_{i|p,c}$ represents the probability of response i to Item 1 at Time t, conditional on membership in latent status p at Time t and on membership in latent class c; $\rho_{i'|q,c}$ represents the probability of response i' to Item 1 at Time t+1, conditional on membership in latent status q at Time t+1 and on membership in latent class c, etc.; and $\rho_{m|c}$ represents the probability of having a value of m on the indicator of latent class membership, conditional on membership in latent class c.

With a second-order model, transitions between latent statuses are conditional on latent status memberships at Time *t-1* and *t-2*. A second-order model can be expressed as follows:

$$P(Y = y) = \sum_{c=1}^{C} \sum_{p=1}^{S} \sum_{pq=1}^{S^2} \sum_{r=1}^{S}$$

$$\gamma_c \rho_{m|c} \delta_{p|c} \rho_{i|p,c} \rho_{j|p,c} \rho_{k|p,c} \tau_{q|p,c} \rho_{i'|q,c} \rho_{j'|q,c} \rho_{k'|q,c} \tau_{r|pq,c} \rho_{i''|r,c} \rho_{j''|r,c} \rho_{k''|r,c}$$

where

$\tau_{r|pq,c}$ is an element of the latent transition probability matrix, representing the probability of membership in latent status *r* at Time *t*+2 conditional on membership in latent status *p* at Time *t*, membership in latent status *q* at Time t+1, and membership in latent class *c*.

The first-order model is a special case of the second-order model where the $\tau_{r|pq,c}$'s for a given *q* and *c* are equal across all *p*'s. At least three occasions of measurement are necessary to fit a second-order model.

ESTIMATION

Parameter estimation in WinLTA is performed by means of the EM algorithm (Dempster, Laird, & Rubin, 1977; Goodman, 1974). At each iteration, the program computes the convergence index, the Mean Absolute Deviation (MAD). MAD is the mean of the absolute value of the difference between the current value of each parameter being estimated and its value at the previous iteration. The MAD is an index of how much progress is being made at each iteration. In the first several iterations, the MAD is usually relatively large, because each iteration is making large changes in the parameter estimates. Usually the MAD becomes steadily smaller with each successive iteration.

The user can specify the maximum number of iterations to perform and a convergence criterion. The convergence criterion is the value of the MAD that the user considers so small that any further changes in the parameter estimates are meaningless. We recommend using a convergence criterion of 10^{-6} (the program default value) or smaller. The program stops iterating when either the MAD is smaller than the specified convergence criterion, or the program has performed the maximum number of iterations specified by the user.

Sometimes WinLTA does not converge. This happens when the maximum number of iterations is reached before MAD drops below the convergence criterion. During specification of the control file, the user may choose to have the program write the parameter estimates to a text file for use in another run. If this file has been created and WinLTA did not converge, the estimates written to this file can be used as starting values and additional iterations with WinLTA can be attempted. For example, suppose a user runs a job where the program does not converge after 100 iterations. If the user has saved the parameter estimates in a file, the program can be instructed to read in these parameter estimates and begin the estimation procedure with iteration 101. This can save a lot of time. (See the section "How to Continue a Run that Did Not Converge".)

Data Augmentation

Because standard errors are not a byproduct of the EM algorithm, this method does not allow the user to conduct hypothesis tests. Recently, data augmentation (DA), a Gibbs sampling-based procedure, has been added to WinLTA. Once a final latent class or latent transition model has been selected and the EM algorithm has been applied, DA can be used to obtain final parameter estimates and standard errors for that LTA model. This DA procedure is discussed in detail in the supplemental document *WinLTA User's Guide for Data Augmentation*.

PARAMETERS ESTIMATED

Up to five different sets of parameters are estimated by WinLTA. The models may involve a static latent variable, a dynamic latent variable, or both. Depending on the model being tested, some or all of the five sets of parameters may be estimated. Latent classes always refer to static latent variables and latent statuses always refer to dynamic latent variables.

The γ parameters represent the proportion of the population in each latent class. These parameters are estimated in all LCA, and any LTA analyses that involve an exogenous static variable. The γ's always sum to one because the latent class are mutually exclusive and exhaustive. Given this restriction, if C represents the number of latent classes, at most C-1 of these parameters are independently estimated; once C-1 have been estimated, the remaining category can be obtained by subtraction.

The δ parameters represent the proportion of the population in each latent status at each occasion, conditional on latent class membership. These parameters are estimated in any LTA analysis. The δ parameters corresponding to times other than Time 1 are not independently estimated, but rather are a function of other parameters. The δ parameters always sum to one across latent statuses within a latent class. If S represents the number of latent statuses, at most $C(S\text{-}1)$ of these parameters can be independently estimated; within each latent class, once S-1 of these parameters have been estimated, the δ for the remaining category can be obtained by subtraction.

The τ parameters are the transition probabilities, conditional on latent class membership. These parameters are estimated in any LTA analysis. Each row of the transition probability matrix sums to one. If there are T times, at most $CS(S\text{-}1)(T\text{-}1)$ transition probability parameters can be independently estimated for a first-order model. For a second-order model, there are at most $CS(S\text{-}1) + CS^2(S\text{-}1)(T\text{-}2)$ independent parameters.

There are two separate sets of ρ parameters. One set, referred to as the "little ρ's," is associated with the static latent variable; the other set, referred to as the "big ρ's," is associated with the dynamic latent variable. These parameters always sum to one across response categories within an item, time, latent status, latent class combination. If there are I manifest variables measuring the static latent variable, and each of these manifest variables has R response categories, then there are at most $CI(R\text{-}1)$ independent ρ parameters associated with the static latent variable. If there are J manifest variables measuring the dynamic latent variable at T times, and each of these manifest variables has M response categories, then there are at most $CSTJ(M\text{-}1)$ independent ρ parameters.

In general, the maximum number of parameters estimated for a first-order model is $(C\text{-}1) + C(S\text{-}1) + CS(S\text{-}1)(T\text{-}1) + CI(R\text{-}1) + CSTJ(M\text{-}1)$, and for a second-order model $(C\text{-}1) + C(S\text{-}1) + CS(S\text{-}1) + CS^2(S\text{-}1)(T\text{-}2) + CI(R\text{-}1) + CSTJ(M\text{-}1)$. In most applications, there will be fewer

parameters estimated because some parameters will be constrained or fixed (see Identification and Constraints).

SAMPLE SIZE

Users often ask about sample size limitations in LTA. Like other contingency table procedures, LTA is not suitable for very small samples. It is impossible to give simple recommendations about sample size requirements, because these requirements depend on the complexity of the model being fit, the size of the contingency table, and the strength of the rho parameters. All else being equal, complex models involving many parameters, and large contingency tables, require larger sample sizes. When the rho parameters are close to zero and one, on average a smaller N is needed. In general, we recommend that LTA be used only with great caution, or not at all, with sample sizes less than about 300.

GOODNESS OF FIT

Goodness of fit is assessed by comparing the observed and predicted response pattern frequencies. If the assumed model provides a good representation of the data, then the predicted frequencies will be close to the observed frequencies. A poor model will not reproduce the observed response pattern frequencies well.

The WinLTA program output provides the likelihood-ratio chi-square, usually denoted G^2. This quantity is computed as follows:

$$G^2 = 2\sum f_{ijk} \ln(f_{ijk} / \hat{f}_{ijk})$$

where f_{ijk} represents the observed frequency of response pattern *ijk*, and $\hat{f}_{ijk}$ represents the frequency predicted by the model.

Asymptotically, G^2 is distributed as a chi-square with degrees of freedom equal to K-P-1, where K is the number of possible response patterns (i.e., the size of the contingency table; maximum possible number of response patterns) and P is the number of parameters estimated. When the data are sparse (i.e., there are few or no observations in many cells), the chi-square distribution is usually not a good approximation for the distribution of G^2. Sparse tables are likely to be a problem whenever complex LCA or LTA models are estimated. Unfortunately, a better approximation for the distribution of G^2 is not known at this time.

As a partial remedy for the problems associated with the distribution of G^2, we recommend that researchers use double crossvalidation (Cudeck & Browne, 1983; Collins, Graham, Long, & Hansen (1994)). Double crossvalidation involves splitting a sample into two (or more) subsamples, for example, Sample A and Sample B, and fitting a series of plausible

models to each sample. Each model is fitted to Sample A (the calibration sample), the predicted response frequencies for each model are compared to the observed response frequencies in Sample B (the crossvalidation sample), and G^2 is computed. Then the reverse is done; each model is fitted to Sample B (now the calibration sample), the predicted response frequencies for this model are compared to the observed response frequencies in Sample A (now the crossvalidation sample), and another G^2 is computed. A model crossvalidates well if the G^2 is relatively small when the estimated model is applied to a crossvalidation sample. When a series of models is tested, the model or models that crossvalidate best are considered best-fitting.

LTA allows the user to carry out crossvalidation easily. The user has the option of saving the final parameter estimates in a file for use in another LTA run. These parameter estimates can then be read in and a G^2 can be computed on any data. (See the section "How to Crossvalidate," p. 25.)

IDENTIFICATION AND PARAMETER RESTRICTIONS

In order for parameter estimation to proceed properly, the model must be identified. Specifically, there must be enough independent information with which to estimate the parameters of the model. A necessary but not sufficient condition for identification is that the number of parameters being estimated does not exceed one less than the number of possible response patterns. Identification problems are more likely in sparse matrices, although they can occur even with large sample sizes. The user is advised to check the identification of his or her models. Some symptoms of underidentification in LTA and LCA models are: parameter estimates that remain at the starting value (unless they are fixed at that value by the user); parameter estimates that deviate widely from what seems reasonable; parameter estimates that vary wildly when small changes are made in the model or the model is tested on another sample. One can check for identification problems by using two or more sets of start values for a single LTA analysis. The estimates obtained using the various sets of start values should be identical. If they are not, there may be identification problems.

Another indication of potential identification problems is that estimates become worse (meaning that they are farther from the maximum likelihood solution) during the EM run. When a problem is identified, a property of the EM algorithm is that estimates at iteration t are always at least as good as those of iteration t-1 (Dempster, Laird & Rubin, 1977). Typical behavior of an EM run is that as the maximum likelihood solution is approached, the size of the MAD becomes smaller. If the MAD increases, then those estimates may be farther from the maximum likelihood estimate which may indicate identification problems. The iteration history appearing on the output shows the MAD for each pair of iterations and may be checked for an increasing MAD value.

How to Deal with Identification Problems

The user can often achieve identification by placing restrictions on a model. The term "restriction" is used here to refer to any restriction placed on a parameter so that the parameter is not estimated freely. When restrictions are imposed, fewer parameters are estimated and this

means that less information is required from the data. There are two types of restrictions. The first type of restriction is when a parameter is fixed at some particular value. Because the value is fixed, no information from the data is used to estimate it. The second type of restriction is an equality constraint. This is when a parameter is constrained to be equal to one or more other parameters. Parameters that are constrained to be equal to each other are said to form an equivalence set. Estimation of an entire equivalence set requires the same amount of information as estimating one parameter.

In WinLTA the user has the option of specifying restrictions on any parameter. There is a tab corresponding to each type of parameter: gamma, rho, DELTA, TAU, and RHO (where the lower case refers to the static part of the model and the upper case refers to the dynamic part of the model). Within each tab, the user is presented with a spreadsheet-like table. Each cell corresponds to a parameter. By entering a number into each cell, the user can control whether the parameter is estimated freely or constrained. (Examples of these tables appear in the Appendix and below.) The user specifies that a parameter is to be estimated freely by putting a "1," in the appropriate cell. This is the default. A "0" denotes that the user will fix the parameter estimate to a specific value. (The user specifies what the fixed value is in the Starting Values tab, which is discussed below.) Integers greater than one are used to denote equivalence sets. For example, if five cells of a parameter restriction matrix contain the number 4, these five parameters form an equivalence set and will be estimated at the same value.

Some points to remember about parameter restrictions:

1. The number used to specify an equivalence set is arbitrary. It does not matter whether you use a 2, 5, or 23: Parameters designated with the same number form an equivalence set. Other than this the number used to specify the equivalence set has no effect on the parameter estimate, e.g. a larger number will not produce larger estimates.

2. It is not possible to constrain parameters to be equal across types. For example, a ρ cannot be constrained equal to a τ.

3. Because of 2 above, it is OK to use the same numbers in different parameter restriction matrices.

4. In WinLTA, for a particular item, it is not permissible to mix freely estimated and constrained ρ's. It is permissible to mix freely estimated and fixed ρ's for an item, or constrained and fixed ρ's for an item.

Some Rules Governing Constraints on Conditional Parameters

All of the parameters except the γ's are conditional probabilities. Think of a group of parameters that is conditioned on the same quantity (e.g., a latent status) as a *row*. This is perhaps easiest to understand in the case of the τ parameters, which are often arranged in a matrix where each row represents the probabilities of latent status membership at Time t+1 conditional on Time t latent status membership. It is worth describing the 'rows' in the ρ matrices because they are less obvious. The little ρ parameters are conditional on item and latent class membership. The big ρ parameters are conditional on item, latent status membership, time, and in some cases latent class membership. A row of ρ's consists of ρ parameters corresponding to all the response categories of a single item for a single latent class/latent status combination.

The following rules governing the use of constraints on all conditional parameters must be followed in order to guarantee that the estimates will be maximum likelihood estimates. These rules pertain to forming equivalence sets that involve elements from more than one row.

1. If a parameter in Row A is in the same equivalence set as a parameter in Row B, then the sums of any fixed parameters in Row A and Row B must be identical.

2. If Row A contains n elements in a single equivalence set, then if Row B contains any element of that equivalence set it must contain exactly n elements of that equivalence set. It is OK if Row B contains NO members of the equivalence set, but if it contains any members, it must contain exactly n members.

3. Suppose a parameter in Row A is in the same equivalence set, say Set 2, as a parameter in Row B. If a parameter in Row A belongs to another equivalence set, say Set 3, then there must be a parameter in Row B belonging to Set 3. Likewise, if a parameter in Row B belongs to Set 3, there must be a parameter in Row A belonging to Set 3.

We illustrate the rules using a 5 x 5 τ matrix (i.e., five latent statuses at two times).

The restrictions in Table 1 (numbers not in parentheses) follow the system used in WinLTA. The example in Table 1 follows Rule 1 because the sum of the fixed parameters is the same in rows 2 and 3 (i.e., these parameters sum to .1). There are no fixed parameters in row 1, so Rule 1 does not apply to this row. Note that the transition from LS 2 to LS 1 could not be fixed to any value other than 0.1, unless the fixed values in row 3 were changed accordingly. Rule 2 is followed because there is only one element from equivalence set 2 in every row where a 2 appears. The same applies to the equivalence set 3. Rule 3 is followed because in every row where there is a 2 there is also a 3.

Table 1
An example of a τ matrix with correctly specified conditional parameter restrictions. The numbers not in parentheses represent the restriction being made. The numbers in parentheses represent the values of fixed parameters.

	LS 1	LS 2	LS 3	LS 4	LS 5
LS 1	2	3	1	1	1
LS 2	0 (0.1)	2	3	1	1
LS 3	0 (0.05)	0 (0.05)	2	3	1
LS 4	0 (0.0)	0 (0.0)	0 (0.0)	1	1
LS 5	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (1.0)

Table 2 below provides an example that does not meet the rules.

Table 2
An example of a τ matrix with INCORRECTLY specified conditional parameter restrictions.

	LS 1	LS 2	LS 3	LS 4	LS 5
LS 1	2	3	3	1	0 (0.0)
LS 2	0 (0.1)	2	3	1	1
LS 3	0 (0.05)	2	2	3	1
LS 4	0 (0.033)	1	2	2	3
LS 5	0 (0.025)	1	1	2	1

Rule 1 is not met because the sum of the non-zero fixed elements is not the same in each row where there are members of equivalence set 2. Rule 2 is not met because some rows have only one element from equivalence set 2 and others have two elements from equivalence set 2. The same applies to equivalence set 3. Rule 3 is not met because members of equivalence set 2 and equivalence set 3 appear together in rows 1, 2, 3, and 4, but a member of equivalence set 2 appears in row 5 without a member of equivalence set 3.

Hints about Parameter Restrictions

Parameter restrictions can be applied to any of the parameters in WinLTA, but they are most often applied to τ and big ρ parameters. Restrictions placed on the τ parameters provide a way for users to test ideas about change over time. Restrictions placed on the ρ parameters allow the user to ensure that the meaning of the latent statuses remains the same over time and across latent classes. In addition, the big ρ's are usually the largest set of parameters, so reducing the number of big ρ's can make a big difference in model identification. In this section we discuss some hints for setting up restrictions on these parameters.

τ parameters. In Example 2 (discussed in more detail in Collins & Wugalter, 1992), the math skills of high school students were assessed in the 10^{th} and 12^{th} grades. In this model there are five latent statuses: No skill; Single operations on whole numbers; Powers and roots; Algebra; and Geometry. The τ matrix, often referred to as the transition probability matrix, expresses the probability of being in a particular latent status in 12^{th} grade, given 10^{th} grade latent status membership. In Example 2, the τ matrix is freely estimated. To denote this, 1's have been entered in the entire parameter restriction matrix. In this model, there are no restrictions on longitudinal change in math skill.

Suppose a user wishes to test a model which states that high school students do not forget math skills or regress in math ability. This model can be tested by fixing some of the τ's to zero. This requires the user to do two things: (1) place a zero in the corresponding elements of the τ parameter restriction matrix (which is found on the TAU Parameter Restrictions tab in WinLTA); and (2) place the desired value for the τ in the corresponding element of the starting value matrix (which is found on the Tau - Starting Values tab). Figure 1 shows the τ parameter restriction matrix that would be needed. The 0's in this matrix mean that the corresponding parameter is fixed to a prespecified value, not that the parameters are fixed to zero. The prespecified values for the parameters are entered in the starting value matrix.

	NO SKILL	SINGLE	POWERS	ALGEBRA	GEOMETRY
▶ NO SKILL	1	1	1	1	1
SINGLE	0	1	1	1	1
POWERS	0	0	1	1	1
ALGEBRA	0	0	0	1	1
GEOMETRY	0	0	0	0	1

Figure 1. Restrictions on τ parameters

Now suppose a user wishes to test a model which states that there is no change in math skills over time. For this model, the user would place 1's along the diagonal and 0's off the diagonal. These restrictions result in a τ matrix that consists of only fixed parameters.

ρ parameters. We recommend constraining the big ρ's to be equal across times in LTA models. This ensures that the latent statuses have the same meaning across times, and makes the results easier to interpret. This also can effect a considerable reduction in parameter estimation.

With some models, constraint patterns applied to the ρ parameters can greatly reduce the number of parameters that have to be estimated. We will illustrate these ideas using a latent class model and little ρ's, but they can be applied to latent transition models as well. Suppose four pass/fail test items are given to a group of people to test whether they have a particular skill. It is hypothesized that there are three latent classes: Non-Masters of the skill, Novices, and Masters. Table 3 reflects which ρ parameters are expected to be high and which are expected to be low in this model.

Table 3
Expected Pattern of ρ Parameters for a Model With Non-Master, Novice, and Master Latent Classes

Probability of Failing				
	Item 1	**Item 2**	**Item 3**	**Item 4**
Non-Masters	High	High	High	High
Novices	Low	Low	High	High
Masters	Low	Low	Low	Low

Probability of Passing				
	Item 1	**Item 2**	**Item 3**	**Item 4**
Non-Masters	Low	Low	Low	Low
Novices	High	High	Low	Low
Masters	High	High	High	High

There are a total of 24 ρ's. However, as the probability of passing and the probability of failing must sum to one within each latent class, at most 12 ρ's can be freely estimated.

Dayton and Macready (1976) argued that there are two types of response error that could occur. One type of response error is failing when according to the latent class the individual should have passed. We will call this Type A. The other type of response error is passing when according to the latent class the individual should have failed. We will call this Type B. We will refer to the other two latent class-response combinations, i.e., passing when passing is expected and failing when failing is expected, as Correct-P and Correct-F, respectively. We can then consider which ρ parameters correspond to correct responses, which to Type A errors, and which to Type B errors, as depicted in Table 4.

Table 4
Pattern of Errors and Correct Responses, Conditional on Latent Class

Probability of Failing				
	Item 1	**Item 2**	**Item 3**	**Item 4**
Non-Masters	Correct-F	Correct-F	Correct-F	Correct-F
Novices	A	A	Correct-F	Correct-F
Masters	A	A	A	A

Probability of Passing				
	Item 1	**Item 2**	**Item 3**	**Item 4**
Non-Masters	B	B	B	B
Novices	Correct-P	Correct-P	B	B
Masters	Correct-P	Correct-P	Correct-P	Correct-P

With the constraints shown above, the A errors form one equivalence set and the B errors form a separate equivalence set. The ρ's in the Correct-P cells are the complements of the ρ's in the A cells, and ρ's in the Correct-F cells are the complements of the ρ's in the B cells. The complements must be treated as equivalence sets when the user is specifying constraints, however, they are not estimated because they can be obtained by subtraction. Thus, there are four equivalence sets per item, of which two are actually estimated.

The constraints can be set up as shown in Figure 2. With this pattern of constraints, eight ρ's are estimated. Note that if additional latent classes were added to this model and the pattern of constraints was unchanged, the number of ρ's estimated would remain eight.

	ITEM 1	ITEM 2	ITEM 3	ITEM 4
	FAIL	FAIL	FAIL	FAIL
NON-MAST	2	7	11	15
NOVICES	5	9	11	15
MASTERS	5	9	13	17
	PASS	PASS	PASS	PASS
NON-MAST	4	8	12	16
NOVICES	6	10	12	16
MASTERS	6	10	14	18

Figure 2. Pattern of constraints for estimating a single A error and B error for each item.

The number of ρ parameters can be reduced further by constraining all the A errors to form a single equivalence set, and all the B errors to form an equivalence set, in other words, constraining each type of error to be equal across items. This pattern of constraints, which results in estimation of two ρ's, is depicted in Figure 3.

	ITEM 1	ITEM 2	ITEM 3	ITEM 4
	FAIL	FAIL	FAIL	FAIL
NON-MAST	2	2	2	2
NOVICES	5	5	2	2
MASTERS	5	5	5	5
	PASS	PASS	PASS	PASS
NON-MAST	4	4	4	4
NOVICES	6	6	4	4
MASTERS	6	6	6	6

Figure 3. Pattern of constraints for estimating a single A error and B error overall.

Rather than constraining the ρ's to be equal across items, another approach is to constrain the A and B errors to be equal for each item. This type of constraint effectively defines one type of response error per item, an error associated with giving the wrong response. This pattern of constraints is depicted in Figure 4.

		ITEM 1	ITEM 2	ITEM 3	ITEM 4
		FAIL	FAIL	FAIL	FAIL
▶	NON-MAST	2	4	7	9
	NOVICES	5	6	7	9
	MASTERS	5	6	8	10
		PASS	PASS	PASS	PASS
	NON-MAST	5	6	8	10
	NOVICES	2	4	8	10
	MASTERS	2	4	7	9

Figure 4. Pattern of constraints for estimating a single error parameter for each item.

Finally, it is possible to constrain all of the errors to be equal across all items. This requires estimation of only one ρ parameter.

RESIDUALS

At the user's option, the program prints out raw and standardized Pearson residuals. The raw residual associated with each response pattern is the difference between the observed and expected cell frequency. The standardized residual is obtained by dividing the raw residual by the square root of the expected frequency. Residuals are often helpful when a user is trying to understand why a model does not fit well. A large residual for a particular response pattern indicates that the model does not predict the response pattern well, and usually points to adjustments that can be made in the model to improve fit. The program prints only residuals associated with observed response patterns.

MISSING DATA IN LTA

Introduction to Missing Data

When individuals do not respond to all of the items used to determine latent class/status, WinLTA uses computational techniques that have been developed to handle incomplete data in contingency tables (Schafer, 1997). From the user's point of view, the missing data routine is nearly transparent. The user need only code the missing data with a 0, specify that there is missing data in the dataset, and run WinLTA as usual. By doing this, the parameter estimates arrived at by the EM algorithm take into account the missing data. In a simulation study, Hyatt and Collins (1998) showed that WinLTA's estimation procedure was robust when the data are missing completely at random (MCAR) or missing at random (MAR). If missingness on a variable *Y* does not depend on *Y* itself, it is MAR. MCAR is a special case of MAR where the cause of missingness is completely unrelated to *Y*. (For a more detailed explanation, see Collins, Schafer, and Kam, 2001.) If listwise deletion is applied to a dataset where data are MAR, the final parameter estimates will be biased. The missing data procedure in LTA adjusts parameter estimates for this bias provided that the cause of missingness or close correlates are included in the analysis. It also adjusts the model fit statistic, provides a test for whether the data are

MCAR, and adjusts the residuals (see below). As the missing data procedure is quite simple for the user to implement, we recommend that all analyses conducted on incomplete datasets employ this technique. Example 5 in the Appendix demonstrates the missing data technique in LTA.

(New users may wish to skip the following three sections.)

Goodness of Fit Statistic Adjusted for Missing Data

The typical G^2 statistic is used as a goodness-of-fit measure for LTA models. In the case of incomplete data, if the data are MAR this statistic is not appropriate. Under these conditions the G^2 is inflated, making fit appear worse than it actually is. This may result in rejecting models that actually fit the data sufficiently well. This is because the G^2 reflects the sum of two quantities: a component of the loglikelihood for model fit and a component of loglikelihood for the missing data mechanism (Little & Rubin, 1987). In order to assess fit, these two components must be separated. Fitting the saturated model to the incomplete data according to Little and Rubin (1987) provides a basis for separating these two components. The fit statistic which represents model fit for incomplete data is equal to the typical G^2 statistic minus the G^2 statistic based on the saturated model. This difference reflects the adequacy of the current LTA model and can be interpreted as a typical G^2 statistic with the typical associated degrees of freedom for complete data problems. This tests the null hypothesis that the current LTA model fits the data sufficiently well, and we reject the null when the final G^2 statistic is large compared to the degrees of freedom. See Lanza, Collins, and Schafer (in preparation) for details on goodness of fit in LTA models for incomplete data.

Test of MCAR

In order to obtain a G^2 statistic that reflects model fit for incomplete datasets, WinLTA performs the following. First, G^2_{raw} is computed, which is the unadjusted G^2 for the LTA model. Then, G^2_{sat} is computed, which is the G^2 statistic for the fit of the saturated model to the incomplete data. Finally, the difference between the two is computed:

$$G^2_{adj} = G^2_{raw} - G^2_{sat}$$

The result, G^2_{adj} (referred to in the output file as "G-Squared Test of Model Fit"), provides a statistic which reflects the fit of the LTA model for incomplete datasets. G^2_{sat} (referred to in the output file as "G-Squared Test for MCAR") is also useful, in that it provides a test of the null hypothesis that the data are MCAR. For large G^2_{sat} values relative to the degrees of freedom, we reject the null hypothesis that the data are MCAR. Small G^2_{sat} values provide evidence that the incomplete data are missing completely at random. For more information see Lanza, Collins, and Schafer (in preparation).

Fit Indicators

For complete data, each cell of the contingency table has an associated expected cell count under the LTA model and an observed count. The difference between the observed and expected cell counts is the residual for each cell. Just as the traditional G^2 statistic reflects a combination of lack of fit and departures from MCAR, so do the traditional residuals. In order to utilize the residuals for diagnosing fit problems, it would be helpful to tease apart these two components. For each cell, the difference in the expected cell count under the saturated model and the expected cell count under the LTA model reflects the lack of fit. When the user specifies that the data are incomplete and requests the residuals output, the following quantities appear: the response pattern, the expected cell count under the saturated model, the expected cell count under the LTA model, the difference between these two cell counts (called the fit indicator, and analogous to a residual), and a scaled fit indicator (analogous to a Pearson residual). Large scaled fit indicators indicate response patterns that may not be sufficiently accounted for by the LTA model and suggest parts of the model which may be misspecified. For datasets with a severe amount of missing data, the scaled fit indicators lack statistical power. Possible solutions to this issue are currently being explored. However, the relative sizes of the scaled fit indicators are meaningful regardless of their magnitude, and possible sources of misspecification can be explored by identifying the largest scaled fit indicators. See Lanza, Collins, and Schafer (in preparation) for more details on model fit and residuals for LTA models with incomplete data.

GENERAL INSTRUCTIONS FOR RUNNING THE PROGRAM

Overview

WinLTA takes input from several files and can create several data sets. A diagram of this appears in Figure 5. The data sets read by WinLTA are: (1) the data to be analyzed; (2) the program control file (which contains information about the model that is being tested and the options for estimation and output that have been chosen by the user); and (3) optionally, a data set containing parameter estimates from another run. This third data set is needed when the user is carrying out crossvalidation, or when the user wishes to continue estimation started by a previous run that did not converge. The data sets written out are: (1) always, a data set containing the program output, including the iteration history and parameter estimates; and (2) optionally, a data set containing parameter estimates. This optional data set is useful when the user wants to use the parameter estimates in a future crossvalidation, or when the WinLTA run does not converge and the user can continue the estimation procedure in a subsequent run.

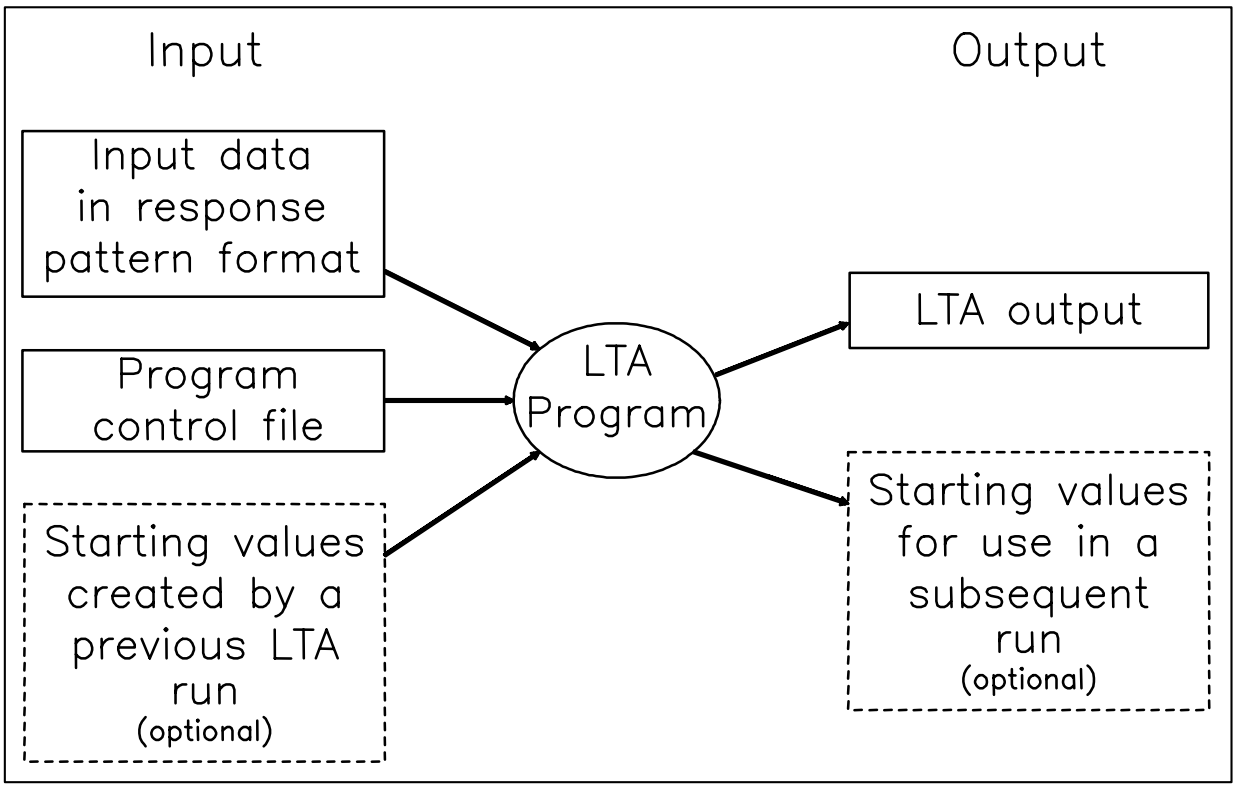

Figure 5. An overview of input to and output from LTA.

Preparing the Data

The data must be entered into the WinLTA program in response pattern format (discussed below). Our web site http://methodology.psu.edu contains a utility program, DataAgg.exe, that is available for downloading. It will take an individual level, ASCII data file and create an ASCII, free field, response pattern format data file. You can also use a statistical package to format your data. This section of the manual covers a few things you need to know before you format your data using either the DataAgg.exe program or some other means.

A response pattern is a set of possible item responses. For example, suppose an item has three response alternatives: No, Maybe, and Yes. For WinLTA, each response alternative must be coded as a non-zero integer, beginning with 1. No, Maybe, and Yes could be coded 1, 2, and 3, respectively. Missing data must be coded 0. Now, suppose four such items were administered. Then one possible response pattern would be: 1111 (representing responses of No to all four items); another would be 1112 (representing No to the first three items and Maybe to the fourth); another would be 1012 (representing No to the first item, missing on the second item, No to the third item, Maybe to the fourth), etc.

The WinLTA program requires as input the response patterns and the number of individuals in the sample associated with each response pattern. Below are some guidelines to follow when setting up the data:

(1) Each data line contains one response pattern and its corresponding frequency.

(2) Each response pattern contains responses to the items measuring the static latent variable first, followed by the items measuring the dynamic latent variable at Time 1, then the items measuring the dynamic latent variable at Time 2, etc.

(3) The response alternatives are coded sequentially, i.e. 1, 2, 3 rather than 1, 5, 8. Remember that 0 always means missing.

(4) The easiest way to set up the data is to leave a space between each item and between the items and the frequency (free field). Such a data file might appear as follows:

```
1 1 1  55
1 1 2  31
1 2 1  44
```

and so on, where the first row means that the response pattern 111 has a frequency of 55.

If you use our utility DataAgg.exe, it will set the data up this way for you. If the data are set up this way, WinLTA can read the data set without needing an input format. If your data are set up without a space between each variable, WinLTA will require a FORTRAN input format. The WinLTA help file contains information about how to specify an input format.

(5) Only response patterns with nonzero frequencies need be included. If no one in your sample responds 111, then this response pattern does not need to be included in the data set.

For examples of how to set up the data for input to WinLTA, see the .dat files that are part of the examples included in the WinLTA package.

Starting values

The user must supply starting values in order to start the estimation procedure. The starting values follow essentially the same format as the parameter restrictions. One important difference is that whereas the parameter restrictions are always indicated with integers, the starting values are probabilities.

There are four points to remember about starting values:

1. Starting values should be plausible values for the parameters. The closer the starting values are to the final parameter estimates, the less time the estimation procedure will take.

2. Starting values should be consistent with the constraints. If two or more parameters are set equal to each other, their starting values should be equal.

3. If you have entered a 0 as a parameter restriction for a particular parameter, the parameter is *fixed* at the starting value you enter. In other words, the estimation procedure will not change it.

4. If you use 0 or 1 for a starting value for any parameter, the parameter will effectively be fixed at that value.

If the starting values do not sum to one where appropriate, you will get an error message in the output and the program will not run. Thus, it is important to understand where starting values must sum to one:

1. γ parameters: starting values must sum to one.

2. little ρ parameters: starting values must sum to one for each response category within each latent class. See the Examples for more details.

3. δ parameters: starting values must sum to one for each latent class.

4. τ parameters: each row of the starting value matrix must sum to one.

5. big ρ parameters: starting values must sum to one across response categories within each latent class/latent status combination. Note that starting values for the big ρ parameters are specified only once. The program will use these start values for all times, whether the estimates are constrained to be equal across times or not. See the Examples for details.

HINTS

How to Get Started

When you download the WinLTA program, you also receive program control files and data for each of the examples in the Appendix. You can get started by running these sample problems. You can even modify these control files to suit your own data.

How to Crossvalidate

Suppose you wish to fit a model to Sample A and crossvalidate in Sample B. First, set up the model for Sample A. On the General tab, where you are asked "Send parameter estimates to a file to be used later?" choose Yes and specify the file name. Run the model for Sample A.

You can use the same program control file to do the crossvalidation, with some minor changes.

1. General tab: You will probably want to change the title.

2. Model and Data tab: Number of Subjects has to be changed if Sample B has a different number of subjects than Sample A. Number of Response Patterns has to be changed if Sample B has a different number of response patterns than Sample A.

3. Estimation tab: Where it says Type of Estimation, choose "No estimation, goodness-of-fit only." Where it says "No Estimation/Continue Previous," enter the name of the file where you are storing the parameter estimates from the run on Sample A.

The following must stay the same: Number of Latent Classes; Number of Latent Statuses; Number of Items Measuring the Static Latent Variable; Number of Items Measuring the Dynamic Latent Variable; Number of Times; Order of the Process.

How to Continue a Run That Did Not Converge

On the General tab, the user is asked "Send parameter estimates to a file to be used later?" If you choose "Yes" and specify the file name, you can continue the job if it does not converge. A job fails to converge when the maximum number of iterations is reached before the convergence criterion is met. You will know if the job did not converge because a warning message will appear on the output. If you did not choose to send the parameter estimates to a file when you ran the job, you cannot continue the run if it did not converge.

If a job does not converge and you have saved the parameter estimates in a file, you can continue the job using the same program control file, with the following modifications:

1. You may want to change the title.

2. On the Estimation tab, choose "Continue Previous Run."

3. Change the Maximum Number of Iterations to the maximum number of additional iterations you want. For example, if your previous job went for 40 iterations without converging, and you would like no more than 100 iterations overall, enter 60. On the job output, the first iteration will be 41.

The following must stay the same: Number of Latent Classes; Number of Latent Statuses; Number of Items Measuring the Static Latent Variable; Number of Items Measuring the Dynamic Latent Variable; Number of Subjects; Number of Times; Order of the Process; Number of Response Patterns.

TROUBLESHOOTING

Here are a few commonly occurring problems and some suggestions for how to deal with them.

- STARTING VALUES DO NOT SUM TO ONE message appears on the program output. The program will not run if the starting values do not sum exactly to one where they are supposed to. Make sure you understand which quantities are supposed to sum to one (see the section "Parameters Estimated"). It is easy to make this mistake, especially in the transition probability matrix.

- PROGRAM DOES NOT CONVERGE AFTER MANY ITERATIONS. It is not unusual for LTA problems to take 500 or more iterations to converge. If a problem seems not to be converging, try the following:

  (1) Try a new set of starting values. Make sure you have thought about the start values and have chosen values that seem reasonable.

  (2) Examine the iteration history. If the MAD is steadily decreasing, the estimation procedure will converge eventually. However, if the convergence index is alternating between two values, or cycling repeatedly among several values, the problem may never converge at the convergence criterion you have chosen. Consider choosing a larger convergence criterion, or apply parameter restrictions to reduce the number of parameters being estimated.

- DIVISION BY ZERO message. This message appears if one of the latent statuses has an estimated probability of zero or very near zero. The program has to divide by the proportion in each latent status to obtain some of the parameters. Sometimes a latent status that has become very small is effectively empty and can be removed from the model.

REFERENCES

Clogg, C.C., & Goodman, L.A. (1984). Latent structure analysis of a set of multidimensional contingency tables. *Journal of the American Statistical Association*, *79*, 762-771.

Collins, L.M., & Cliff, N. (1990). Using the longitudinal Guttman simplex as a basis for measuring growth. *Psychological Bulletin*, *108*, 128-134.

Collins, L. M., Graham, J. W., Long, J., & Hansen, W. B. (1994). Cross validation of latent class models of early substance use onset. *Multivariate Behavioral Research, 29,* 165-183.

Collins, L. M., Schafer, J. L., & Kam, C.-M. (2001). A comparison of inclusive and restrictive strategies in modern missing-data procedures. *Psychological Methods*, *6*, 330-351.

Collins, L.M., & Wugalter, S.E. (1992). Latent class models for stage-sequential dynamic latent variables. *Multivariate Behavioral Research*, *27*, 131-157.

Cudeck, R.A., & Browne, M.W. (1983). Cross-validation of covariance structures. *Multivariate Behavioral Research*, *18*, 147-167.

Dayton, C.M., & Macready, G.B. (1976). A probablistic model for validation of behavioral hierarchies. *Psychometrika*, *41*, 189-204.

Dempster, A.P., Laird, N.M., & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society (B)*, *39*, 1-38.

Hyatt, S.L., & Collins, L.M. (1998). Estimation in latent transition models with missing data. University Park, PA: Technical report 98-22, The Methodology Center, Pennsylvania State University.

Lanza, S.T., Collins, L.M., & Schafer, J.L. (in preparation). A likelihood ratio statistic and residuals for LTA models with missing data.

Little, R.J.A., & Rubin, D.B. (1987). *Statistical analysis with missing data.* New York: Wiley.

Rock, D.A., & Pollack-Ohls, J. (1987). Measuring gains: A new look at an old problem. Princeton: Educational Testing Service.

Schafer, J.L. (1997). *Analysis of incomplete multivariate data*. London: Chapman & Hall.

van de Pol, F., & Langeheine, R. (1989). Mover-stayer models, mixed Markov models and the EM algorithm; with an application to labour market data from the Netherlands Socio Economic Panel. In R. Coppi & S. Bolasco (eds.), *Multiway data analysis*. North Holland: Amsterdam, pp. 485-495.

RECOMMENDED READINGS

This section contains a set of introductory and advanced references on latent class and latent transition models. Some of the references were cited in the manual, others are additional recommended reading.

Introductory Readings

Collins, L.M., Graham, J.W., Rousculp, S.S., & Hansen, W.B. (1997). Heavy caffeine use and the beginning of the substance use onset process: An illustration of latent transition analysis. In K. Bryant, M. Windle, & S. West (Eds.), *The Science of Prevention: Methodological Advances from Alcohol and Substance Abuse Research*. Washington, D.C.: American Psychological Association. pp. 79-99.

Collins, L.M., Hyatt, S.L., & Graham, J.W. (2000). LTA as a way of testing models of stage-sequential change in longitudinal data. In Little, T. D., Schnabel, K. U., & Baumert, J. (Eds.), *Modeling Longitudinal and Multiple-Group Data: Practical Issues, Applied Approaches, and Specific Examples*. Hillsdale, NJ: Erlbaum.

Collins, L. M., Schafer, J. L., & Kam, C.-M. (2001). A comparison of inclusive and restrictive strategies in modern missing-data procedures. *Psychological Methods*, *6*, 330-351.

Hyatt, S.L., & Collins, L.M. (2000). Using latent transition analysis to examine the relationship between parental permissiveness and the onset of substance use. In Rose, J., Chassin, L., Presson, C., & Sherman, S. (Eds.), *Multivariate Applications in Substance Use Research*. Hillsdale, NJ: Erlbaum.

Velicer, W.F., Martin, R.A., & Collins, L.M. (1996). Latent transition analysis for longitudinal data. *Addiction*, *91* (supplement), S197-S209.

Advanced Readings

Clogg, C.C., & Goodman, L.A. (1984). Latent structure analysis of a set of multidimensional contingency tables. *Journal of the American Statistical Association*, *79*, 762-771.

Collins, L.M., & Wugalter, S.E. (1992). Latent class models for stage-sequential dynamic latent variables. *Multivariate Behavioral Research*, *27*, 131-157.

Cudeck, R.A., & Browne, M.W. (1983). Cross-validation of covariance structures. *Multivariate Behavioral Research*, *18*, 147-167.

Dayton, C.M., & Macready, G.B. (1976). A probablistic model for validation of behavioral hierarchies. *Psychometrika*, *41*, 189-204.

Goodman, L.A. (1974). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*, *61*, 215-231.

Rindskopf, D. (1983). A general framework for using latent class analysis to test hierarchical and nonhierarchical models. *Psychometrika*, *48*, 85-97.

Schafer, J.L. (1997). *Analysis of incomplete multivariate data*. London: Chapman & Hall.

van de Pol, F., & Langeheine, R. (1989). Mover-stayer models, mixed Markov models and the EM algorithm; with an application to labour market data from the Netherlands Socio Economic Panel. In R. Coppi & S. Bolasco (eds.), *Multiway data analysis*. North Holland: Amsterdam, pp. 485-495.

Introduction to the Appendix

The appendix of the WinLTA User's Manual contains five examples of analyses that can be performed using WinLTA. The examples are intended to help the user identify and carry out the type of analysis that is appropriate for his or her research question and data. The user will learn how to enter information into the program control file through detailed explanations (including screen shots) of each of the tabs in WinLTA. The examples will also provide an explanation of the different sections of the output file. The reader will find these examples to be more beneficial if he or she has previously read the main body of the WinLTA User's Manual.

Example 1 is a latent class example that uses empirical data. This example includes a detailed description of the substantive and logical thought involved in setting up an analysis. Attention is also given to the constraints for the little rho parameters.

Example 2 uses the same substantive idea and data used in Example 1 to illustrate a latent transition problem. This example does not contain a static variable.

Example 3 is a full latent transition analysis with three occasions of measurement, three latent classes, and two latent statuses. Artificial data is used in this example.

Example 4 uses artificial data to illustrate a second-order latent transition analysis. This model has five occasions of measurement and two latent statuses, but does not contain a static variable.

Example 5 uses empirical data to illustrate a latent transition problem with missing data. This model has two occasions of measurement and five latent statuses, but does not contain a static variable.

The five examples are available from the WinLTA download page at our website (see below).

We are very interested in any questions or comments you have about the WinLTA program. Please contact:

Linda M. Collins
The Methodology Center
159 Henderson South
Pennsylvania State University
University Park, PA 16802

winlta@psu.edu

We recommend checking our web site regularly for updates of WinLTA, revised versions of this manual, technical reports, and other information, including software for other applications:

http://methodology.psu.edu