The Stata Journal (2019)
19, Number 2, pp. 294–334 DOI: 10.1177/1536867X19854015

mixmcm: A community-contributed command for fitting mixtures of Markov chain models using maximum likelihood and the EM algorithm

Legrand D. F. Saint-Cyr
SMART–LERECO, Agrocamus-Ouest, INRA
Rennes, France
legrand.saint-cyr@inra.fr

Laurent Piet
SMART–LERECO, Agrocamus-Ouest, INRA
Rennes, France
laurent.piet@inra.fr

Abstract. Markov chain models and finite mixture models have been widely applied in various strands of the academic literature. Several studies analyzing dynamic processes have combined both modeling approaches to account for unobserved heterogeneity within a population. In this article, we describe `mixmcm`, a community-contributed command that fits the general class of mixed Markov chain models, accounting for the possibility of both entries into and exits from the population. To account for the possibility of incomplete information within the data (that is, unobserved heterogeneity), the model is fit with maximum likelihood using the expectation-maximization algorithm. `mixmcm` enables users to fit the mixed Markov chain models parametrically or semiparametrically, depending on the specifications chosen for the transition probabilities and the mixing distribution. `mixmcm` also allows for endogenous identification of the optimal number of homogeneous chains, that is, unobserved types or "components". We illustrate `mixmcm`'s usefulness through three examples analyzing farm dynamics using an unbalanced panel of commercial French farms.

Keywords: st0556, mixmcm, Markov chain model, finite mixture model, EM algorithm, mlogit, fmlogit

1 Introduction

The Markov chain model (MCM) is a modeling approach widely used in several strands of the literature to analyze dynamic stochastic processes within a given population where future states depend on the past according to some probability. Numerous applications of MCMs can be found, for example, in economics, medicine, sociology, etc. Whatever the context, one problem practitioners often face is that the population under study may comprise heterogeneous agents who behave differently. Such heterogeneity is generally unobserved and cannot be captured by the observable characteristics of agents. As a result, some models have been developed to deal with unobserved heterogeneity (Greene 2018). Among these, finite mixture models offer advantages that have contributed to their prevalence in the literature (Compiani and Kitamura 2016). Briefly, these models can be considered a special case of individual parameter models, where the parameters of the model are supposed to differ according to specific types of individuals or agents. Finite mixture models, also called latent class models, are generally used to partition a population into homogeneous types to account for heterogeneous behaviors. In Stata, official commands such as `fmm` and community-contributed commands (`gllamm` by Rabe-Hesketh [1999], `lclogit` by Pacifico and Yoo [2013], etc.) provide methods for fitting finite mixture models. However, none of these commands can be used to directly fit a mixture of MCMs.

Mixed Markov chain models (MMCM) have a long history in the literature and have proved to be useful for analyzing dynamic Markovian processes in heterogeneous populations. MMCM have been applied to analyze several economic issues such as labor mobility (see Blumen, Kogan, and McCarthy [1955] and Fougère and Kamionka [2003] for examples); income mobility (see Shorrocks [1976] and Dutta, Sefton, and Weale [2001] for examples); financial rating (see Frydman and Kadam [2004] and Frydman and Schuermann [2008] for examples); or firm-size dynamics (see Cipollini, Ferretti, and Ganugi [2012] and Saint-Cyr and Piet [2017] for examples). Other applications of MMCM can be found in sociology (see Singer and Spilerman [1974] and Dias and Vermunt [2007] for examples); in medicine (see Albert [1991], Chen, Duffy, and Tabar [1997], and Dias and Willekens [2005] for examples); or in other strands of the literature such as natural resources management (see Jackson [1975] for example). These studies of dynamics use different forms of the MMCM, ranging from a simple stationary mover–stayer specification (see Blumen, Kogan, and McCarthy [1955], Fougère and Kamionka [2003], and Saint-Cyr and Piet [2017] for examples) to more general specifications (see Dias and Vermunt [2007], Frydman and Schuermann [2008], Saint-Cyr [2017], and Frydman and Matuszyk [2018] for examples).

In this article, we present `mixmcm`, a community-contributed command that fits the general form of the MMCM. The rest of this article is structured as follows. Section 2 presents the formulation of the MMCM and the estimation strategy. Section 3 presents the `mixmcm` command. Section 4 provides three examples of how the command can be applied. Section 5 presents some procedure for hypothesis testing, transition probability, and elasticity derivation using the estimates from the `mixmcm` command. Section 6 concludes with some possible improvements to the command.

2 Fitting finite mixtures of MCMs

Let N be the total number of agents in the population and $K+1$ be the total number of states or choice alternatives. Assuming a discrete-time process, y_{it} represents the state of agent i ($i = 1, 2, \ldots, N$) at time t ($t = 0, 1, \ldots, T_i$). The indicator variable y_{it} is equal to j ($j = 0, 1, \ldots, K$) if agent i is in state j at time t. State $j = 0$ is arbitrarily chosen to indicate entry into or exit from the population. The length of time (T_i) for which an agent is observed may vary across agents (that is, $T_i \leq T$). Over time period T_i, the row-vector $\mathbf{y}_i = (y_{i0}, y_{i1}, \ldots, y_{iT_i})$ represents the set of transitions of agent i over the K state-space categories. Assuming that the movements of agents follow a first-order Markov process, the probability density function describing the transition process across states can be derived as (Dias and Willekens 2005)

$$f(\mathbf{y}_i) = P(y_{i0} = j) \left\{ \prod_{t=1}^{T_i} P\left(y_{it} = k | y_{it-1} = j\right) \right\}$$

where $P(y_{i0} = j)$ is the probability that agent i starts in state j at time $t = 0$ and $P(y_{it} = k|y_{it-1} = j)$ is the probability that agent i moves to state k at time t given that it was in state j at time $t - 1$.

Suppose now that the observed sample of agents is divided into G homogeneous types instead of a single type, where agents of the same type are characterized by a similar Markovian process. The density function of $\mathbf{y}_i$ is thus a discrete mixing distribution with G support points (McLachlan and Peel 2000). Assuming heterogeneity, the transition process of agents can be represented by the MMCM formulated as (Vermunt 2010)

$$f(\mathbf{y}_i) = \sum_{g=1}^{G} P(g_i = g) P(y_{i0} = j|g_i = g) \left\{ \prod_{t=1}^{T_i} P(y_{it} = k|y_{it-1} = j, g_i = g) \right\} \tag{1}$$

Thus, the MMCM has three components. The first term on the right-hand side of (1) represents the probability that agent i belongs to a specific type g. The second and the third terms, respectively, represent the probability that agent i begins in a specific state j and the probability that agent i moves across states during the T_i time period, given that it is of specific type g.

2.1 Specification method

The three components of (1) can be specified as functions of exogenous variables. In this case, the model is fully parametric. Assuming that the states are finite, exhaustive, and mutually exclusive, we use a discrete-choice approach to specify initial states (or entry) and transition probabilities. Furthermore, if independence from irrelevant alternatives can be assumed for the odds ratios, we can use a multinomial specification (Greene 2018). This leads to a multinomial logit expression for the probability of starting in a specific state j and the conditional probability of making a transition from state j to state k. The probability of starting in state j thus writes

$$P(y_{i0} = j|g_i = g, \mathbf{x}_{i0}) = \frac{\exp\left(\boldsymbol{\beta}'_{j|g}\mathbf{x}_{i0}\right)}{\sum_{k=1}^{K} \exp\left(\boldsymbol{\beta}'_{k|g}\mathbf{x}_{i0}\right)} \quad (\forall j, k = 1, 2, \ldots, K) \tag{2}$$

where $\boldsymbol{\beta}_{j|g}$ is a vector of parameters specific to each type g and each initial state j and $\mathbf{x}_{i0}$ is the vector of explanatory variables for agent i at time $t = 0$. One of these vectors is set to zero for identification purposes.

The transition probabilities across states are specified as

$$P(y_{it} = k | y_{it-1} = j, g_i = g, \mathbf{x}_{it-1}) = \frac{\exp\left(\boldsymbol{\theta}'_{jk|g}\mathbf{x}_{it-1}\right)}{\sum_{l=1}^{K}\exp\left(\boldsymbol{\theta}'_{jl|g}\mathbf{x}_{it-1}\right)} \tag{3}$$

where $\boldsymbol{\theta}_{jk|g}$ is a vector of parameters specific to each type g and each transition jk. Because entries into the population are specified separately from transitions across states, the initial state j in (3) takes the values $1, 2, \ldots, K$ while the final state k takes the values $0, 1, 2, \ldots, K$. Choosing to stay in the same state for two consecutive time periods as the reference scenario sets $\boldsymbol{\theta}_{jj|g} = \boldsymbol{0} \quad \forall g = 1, 2, \ldots, G$ and $\forall j = 1, 2, \ldots, K$ for identification purposes.

Because agent types are not known beforehand, the probability that an agent belongs to a specific type is also estimated. For this, we use a fractional multinomial logit specification because these probabilities are constrained to be between zero and one (Papke and Wooldridge 1996). Thus, the type-membership probability that agent i belongs to type g is

$$P(g_i = g|\mathbf{z}_i) = \frac{\exp\left(\boldsymbol{\lambda}'_g\mathbf{z}_i\right)}{\sum_{h=1}^{G}\exp\left(\boldsymbol{\lambda}'_h\mathbf{z}_i\right)} \quad (\forall g = 1, 2, \ldots, G) \tag{4}$$

where $\boldsymbol{\lambda}_g$ is a vector of parameters and $\mathbf{z}_i$ is a matrix of agent characteristics supposed to be constant over time. The vector of parameters $\boldsymbol{\lambda}_G$ must be normalized to zero for identification.

If a nonparametric form is used for type membership probabilities, the mixing distribution has a nonparametric or discrete-factor interpretation (Pacifico and Yoo 2013). In this case, the mixture model can be viewed as a semiparametric model because we use a nonparametric specification for type membership probabilities and a parametric specification for both the probability of starting in a specific state and the transition probabilities across states. In such a case, the type-membership probability is the same for all agents.

2.2 Expectation-maximization algorithm for the MMCM

We estimate the parameters of the MMCM using the maximum-likelihood estimation technique. Considering that the observed sample of agents is divided into G homogeneous types, the log-likelihood (LL) function for the parameters of the model is

$$\begin{aligned}\text{LL}(\mathbf{y};\boldsymbol{\Phi}) = \sum_{i=1}^{N}\ln\Bigg[\sum_{g=1}^{G}P(\mathbf{z}_i;\boldsymbol{\lambda}_g)\prod_{j}^{K}\left\{P(\mathbf{x}_{i0};\boldsymbol{\beta}_{j|g})\right\}^{d_{ij0}} \\ \prod_{t=1}^{T_i}\prod_{j,k}^{K}\left\{P(\mathbf{x}_{it-1};\boldsymbol{\theta}_{jk|g})\right\}^{d_{ijkt}}\Bigg]\end{aligned} \tag{5}$$

where $\mathbf{\Phi} = (\mathbf{\Phi}_1, \mathbf{\Phi}_2, \ldots, \mathbf{\Phi}_G)$ is the vector of parameters to be estimated, with $\mathbf{\Phi}_g = (\boldsymbol{\lambda}_g, \boldsymbol{\beta}_{j|g}, \boldsymbol{\theta}_{jk|g})$ $\forall g = 1, 2, \ldots, G$, $\forall j = 1, 2, \ldots, K$ and $\forall k = 0, 1, 2, \ldots, K$. The indicators d_{ij0} and d_{ijkt}, respectively, take the value 1 if agent i starts in state j and moves from state j to state k at time t and zero otherwise.

The expectation-maximization (EM) algorithm developed by Dempster, Laird, and Rubin (1977) simplifies the complex LL function in (5) into a set of easily solvable LL functions by introducing a so-called missing variable.[1] Let v_{ig} be a discrete unobserved variable indicating agent type. The random vector $\mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{iG})$ is thus g-dimensional with $v_{ig} = 1$ if agent i belongs to type g and zero otherwise. Assuming that v_{ig} is unconditionally multinomially distributed with probability π_g, the complete likelihood for $(\boldsymbol{\beta}, \boldsymbol{\pi})$, conditional on observing $\mathbf{y}_c = (\mathbf{y}, \mathbf{v})$, is[2]

$$\text{LL}(\mathbf{y}; \mathbf{\Phi}) = \sum_{i=1}^{N} \sum_{g=1}^{G} v_{ig} \ln \left[P(\mathbf{z}_i; \boldsymbol{\lambda}_g) \prod_{j}^{K} \left\{ P(\mathbf{x}_{i0}; \boldsymbol{\beta}_{j|g}) \right\}^{d_{ij0}} \prod_{t=1}^{T_i} \prod_{j,k}^{K} \left\{ P(\mathbf{x}_{it-1}; \boldsymbol{\theta}_{jk|g}) \right\}^{d_{ijkt}} \right] \tag{6}$$

Because agent type is not observed, the "posterior" probability that agent i belongs to type g (that is, v_{ig}) must be derived from the observations. The EM algorithm therefore consists of the following four steps:

i) **Initialization:** Arbitrarily choose initial values $\mathbf{\Phi}^0 = (\mathbf{\Phi}_1^0, \mathbf{\Phi}_2^0, \ldots, \mathbf{\Phi}_G^0)$. To obtain appropriate initial values, we proceed as in Pacifico and Yoo (2013) by randomly assigning each agent to one of the G possible types. Parameters are then estimated for each type.

ii) **Expectation:** At iteration $p+1$ of the algorithm, compute the expected probability that agent i belongs to a specific type g given $\mathbf{y}_i$ and parameters $\mathbf{\Phi}^p$. This conditional expected (that is, "posterior") probability $v_{ig}^{(p+1)} = v_{ig}(\mathbf{y}_i; \mathbf{\Phi}^p)$ is obtained using the following Bayes formula (Dempster, Laird, and Rubin 1977):

$$v_{ig}^{(p+1)} = \frac{P(\mathbf{z}_i; \boldsymbol{\lambda}_g) \prod_j^K \left\{ P(\mathbf{x}_{i0}; \boldsymbol{\beta}_{j|g}) \right\}^{d_{ij0}} \prod_{t=1}^{T_i} \prod_{j,k}^K \left\{ P(\mathbf{x}_{it-1}; \boldsymbol{\theta}_{jk|g}) \right\}^{d_{ijkt}}}{\sum_{h=1}^G P(\mathbf{z}_i; \boldsymbol{\lambda}_h) \prod_j^K \left\{ P(\mathbf{x}_{i0}; \boldsymbol{\beta}_{j|h}) \right\}^{d_{ij0}} \prod_{t=1}^{T_i} \prod_{j,k}^K \left\{ P(\mathbf{x}_{it-1}; \boldsymbol{\theta}_{jk|h}) \right\}^{d_{ijkt}}}$$

Replacing v_{ig} in (6) by its expected value $v_{ig}^{(p+1)}$ gives the conditional expectation of the LL function for the complete data.

1. Indeed, the likelihood function does not yield an explicit solution for the model parameters. The EM algorithm maximizes a log of sums that is transformed into a recursive maximization of the sum of logs (McLachlan and Krishnan 2008).
2. By this assumption, the distribution of the complete-data vector implies the appropriate distribution for the incomplete-data vector (McLachlan and Peel 2000).

iii) **Maximization:** Update $\mathbf{\Phi}^p$ by maximizing the complete LL function conditional on the observations. The model parameters are thus updated as

$$\boldsymbol{\lambda}^{(p+1)} = \text{argmax}_{\boldsymbol{\lambda}_g} \sum_{i=1}^{N}\sum_{g=1}^{G} v_{ig}^{(p+1)} \ln\left\{P(\mathbf{z}_i;\boldsymbol{\lambda}_g)\right\} \quad \forall g = 1,2,\ldots,G-1$$

$$\boldsymbol{\beta}^{(p+1)} = \text{argmax}_{\boldsymbol{\beta}_{j|g}} \sum_{i=1}^{N}\sum_{g=1}^{G} v_{ig}^{(p+1)} \sum_{j}^{K} d_{ij0} \ln\left\{P(\mathbf{x}_{i0};\boldsymbol{\beta}_{j|g})\right\} \quad \forall j = 1,2,\ldots,K-1$$

$$\boldsymbol{\theta}^{(p+1)} = \text{argmax}_{\boldsymbol{\theta}_{jk|g}} \sum_{i=1}^{N}\sum_{g=1}^{G} v_{ig}^{(p+1)} \sum_{t=1}^{T_i}\sum_{j,k}^{K} d_{ijkt} \ln\left\{P(\mathbf{x}_{it-1};\boldsymbol{\theta}_{jk|g})\right\} \quad \forall j \neq k$$

The maximization process of the above equations is straightforward. The parameters ($\mathbf{\Phi}^p$) are updated using $v_{ig}(\mathbf{y}_i;\mathbf{\Phi}^p)$ as a weighting factor for each observation (Pacifico and Yoo 2013). The built-in `mlogit` Stata command is used for the estimation of the starting state and the transition probabilities, while the community-contributed command `fmlogit` (Buis 2008) is used for the type membership probabilities. If a nonparametric form is used for the mixing distribution, the "prior" type membership probability [that is, $P(g_i = g)$] is the same for all agents and can be derived as

$$\pi_g^{(p+1)} = \frac{\sum_{i=1}^{N} v_{ig}^{(p+1)}}{\sum_{i=1}^{N}\sum_{h=1}^{G} v_{ih}^{(p+1)}} \qquad \forall g \in G-1$$

iv) **Iteration:** Return to step ii using $\boldsymbol{\pi}^{(p+1)}$ and $\boldsymbol{\beta}^{(p+1)}$, and iterate until the observed LL given by (5) converges, that is, until the relative difference in the LL between two consecutive iterations is sufficiently small. At convergence, the resulting parameters are considered to be the optimal estimators ($\widehat{\mathbf{\Phi}}$) given the set of initial values ($\mathbf{\Phi}^0$) randomly chosen.

2.3 Heuristic strategy

A problem that often occurs in a mixture analysis with several components is that some solutions may be suboptimal. Indeed, the nonconcavity of the LL function in (5) does not make it possible to identify a global maximum in the mixture model, even for mixtures of multinomial logit models (Hess, Bierlaire, and Polak 2007). Given the potential presence of a high number of local maximums, the EM solutions may depend on the initial values chosen for $\mathbf{\Phi}^0$ (Baudry and Celeux 2015). To increase the chances of obtaining a global maximum in the estimation procedure reported above, we proceed as follows.

First, short-run EMs are used to obtain initial values for long-run EMs. For each short-run EM, the iterative procedure presented in section 2.2 is performed several times with various randomly chosen initial values for the parameters of the model. To do this, the sample is randomly divided into the total number of agent types, and the parameters of the model are estimated separately over each subsample. The resulting parameters are then used in the expectation step of the algorithm to compute the "posterior" probability of agent-type membership. The iterative procedure is stopped after a few iterations, and the resulting parameters that produce the largest LL according to (5) are chosen as the best initial values for long-run EMs.

Second, several long-run EMs are used to obtain the best parameters for the model. For each long-run EM, the iterative procedure is performed until the LL function converges. The parameters that provide the largest LL at the convergence of the EM algorithm are chosen as the global maximum.

3 The mixmcm command

3.1 Syntax

The general syntax for the `mixmcm` command is

`mixmcm` *depvar* [*indepvars*] [*if*] [*in*] [*weight*], `id(`*varname*`)`
`timevar(`*varname*`)` [`noconstant entry(`*varlist*`) exitcode(`*name*`)`
`ncomponents(`*ncomponents_suboptions*`) membership(`*varlist*`)`
`emiterate(`*emiterate_suboptions*`) constraints(`*clist*`)`]

depvar is the dependent variable that indicates agents' states at each time period. *indepvars* are optional explanatory variables that enter the specification of transition probabilities. `fweight`s and `pweight`s are allowed; see [U] **11.1.6 weight**.

3.2 Options

`id(`*varname*`)` specifies the variable that identifies agents. `mixmcm` computes the probability of belonging to a specific homogeneous type for each `id()`. `id()` is required.

`timevar(`*varname*`)` specifies the numeric variable that identifies dates on (or time periods during) which transitions occur. This variable is used to identify the transitions of agents across states. `timevar()` is required.

`noconstant` suppresses the constant term (or intercept) in the specification of the transition probabilities. Specifying the `noconstant` option requires that at least one *indepvar* be specified.

`entry(`*varlist*`)` specifies the dependent and independent variables that enter the specification of entry probabilities. Specifying `entry()` requires that at least the dependent variable indicating the entry state be specified.

`exitcode(`*name*`)` indicates the modality of *depvar* that identifies the exit state.

`ncomponents(`[*#1 #2*`, selcrit(`*name*`) graph(`*namelist*`,` *twoway_options*`)`
`save(`*filename*`, replace detail) force`]`)` specifies the number of (unobserved) homogeneous types and related options.

#1 and *#2* indicate the range for the number of components. If both *#1* and *#2* are specified, `mixmcm` will fit the *#1* − *#2* MMCM, starting from the model with the *#1* component or components and proceeding to the model with the *#2* components to identify the optimal number of components within this range based on the selection criterion specified in the suboption `selcrit()` (see below). Unless the suboption `force` is specified (see below), the estimation will stop automatically when the optimal number of components is found, and the results will solely be displayed for this optimal number of components. If only *#1* is specified, `mixmcm` will estimate only the parameters for this number of component or components, and the corresponding results will be displayed. A standard (homogeneous) MCM will be fit if the specified number of components is *#1* = 1. By default, a two-components MMCM is fit.

`selcrit(`*name*`)` specifies the information criterion used to select the optimal number of components within the *#1* to *#2* range. The available information criteria are `aic`, `bic`, `aic3`, or `caic` (the default). See Andrews and Currim (2003) for a discussion on information criteria for the retention of the optimal number of components.

`graph(`*namelist*`,` *twoway_options*`)` specifies that a graph for the information criteria specified in *namelist* be drawn for the number of components in the *#1* to *#2* range. The `graph()` suboption thus requires that at least one information criterion among `aic`, `bic`, `aic3`, or `caic` be specified in *namelist*. Users can manage the graph using standard *twoway_options*.

`save(`*filename*`, replace detail)` saves the information criteria for the numbers of components estimated within the *#1* to *#2* range in *filename*. Specifying `replace` as a suboption will overwrite an existing *filename*. If `detail` is specified as a `save()` suboption, the resulting parameters for all the estimated numbers of components will be jointly saved.

`force` indicates that models should be fit for all the numbers of components within the *#1* to *#2* range, even when the optimal number of components is found to be smaller than *#2* based on `selcrit()`. Therefore, estimations will continue until *#2* in either case.

`membership(`*varlist*`)` specifies the independent variables to be included in the specification of the component-membership probabilities. These variables must be constant over time for each agent. The parametric form for the mixing distribution is `fmlogit`, which allows the dependent variable to lie between 0 and 1. Specifying `membership()` requires that at least one explanatory variable be specified. By default, type-membership probabilities will be estimated nonparametrically (see Train [2008]).

`emiterate(`[`lr(`*#1 #2*`,` *eps*`) sr(`*#1 #2*`) seed(`*#*`) emlog)`]`)` specifies suboptions for the EM algorithm.

`lr(`*#1 #2*`,` *eps*`)` specifies the number of long-run EMs (*#1*) to be performed, the maximum number of iterations (*#2*) to be used for each long-run EM, and the convergence criterion to stop the iterations (*eps*), respectively. The default is `lr(5 100, 0.0000001)`. *eps* is the tolerance used in the log-likelihood maximization: `mixmcm` declares convergence when the proportional increase in the log likelihood over two consecutive iterations is less than the specified *eps*.

`sr(`*#1 #2*`)` specifies the number of short-run EMs (*#1*) and the maximum number of iterations (*#2*) for each short-run EM. The default is `sr(5 5)`.

`seed(`*#*`)` sets the pseudouniform random-numbers seed. Initial parameters for the EM estimation are randomly chosen using the same seed. The default is `seed(123456)`. The seed is a local macro that replaces seeds that have been chosen by users outside of the command.

`emlog` displays the logs for long-run EM iterations.

`constraints(`*clist*`)` lists the constraints to impose on transition probabilities. Each constraint listed in *clist* must be specified as `constraint` *#* `p_`*initialstate_finalstate* `= 0`, where *#* is the number that identifies the constraint. For now, transition probabilities can be constrained only to be 0, and this constraint applies across all components.

4 Examples

To illustrate how `mixmcm` works, the command is used to estimate size-transition probabilities in the French farming sector under the assumption of a heterogeneous farm population. Data are provided by the Réseau d'Information Comptable Agricole (RICA), the French implementation of the Farm Accountancy Data Network.[3] RICA collects technical and economic information on a sample of commercial French farms on an annual basis. The data are freely available online from the RICA France website (see http://agreste.agriculture.gouv.fr/enquetes/reseau-d-information-comptable/).

For this illustration, we use data from 2000 to 2010. The 11 (annual) databases were appended together, leading to an unbalanced panel where `idnum` is the unique farm identifier and `year` is a generated time variable that will be used as `timevar()` in `mixmcm`. Some modifications of the full original database were necessary to be able to use it with `mixmcm`. First, the population was broken down into three size categories ($K = 3$) according to the economic size variable `pbuce`, which measures the production potential of farms in terms of Euros of standard output (SO). The resulting `category` variable, which will be used as *depvar* in `mixmcm`, therefore consists of three categories, namely, medium farms (denoted `medium` with `pbuce` < 100,000 Euros of SO), large farms

3. For more information about the Farm Accountancy Data Network, see http://ec.europa.eu/agriculture/rica/index_en.cfm.

(denoted `large` with 100,000 ≤ `pbuce` < 250,000 Euros of SO), and very large farms (denoted `vlarge` with `pbuce` ≥ 250,000 Euros of SO). We then restricted the sample to farms that have existed in the database for at least two consecutive years to observe at least one transition. We also retained only a subset of variables and renamed them for use in our examples. The new database, `mixmcm.dta`, is organized as follows:

```
. use mixmcm.dta
. list idnum year surplus istock icap debtr category in 1/10, noobs
> abbreviate(12) compress separator(10)
```

idnum	year	surplus	istock	icap	debtr	category
963	2000	36804	22896	76332	7.00	medium
963	2001	28861	17895	76331	6.40	medium
963	2002	30000	30194	76331	3.90	medium
963	2003	5159	0	76331	4.10	medium
1525	2006	58895	202919	283939	14.60	large
1525	2007	51726	101807	283939	22.10	vlarge
1525	2008	54940	176367	283939	27.20	vlarge
1525	2009	51883	198033	283939	20.00	vlarge
1525	2010	88685	183816	283939	18.10	vlarge
1534	2006	90051	124877	110557	51.10	vlarge

```
. list idnum year crop corp educ young category in 1/10, noobs
> abbreviate(12) compress separator(10)
```

idnum	year	crop	corp	educ	young	category
963	2000	0	0	1	0	medium
963	2001	0	0	1	0	medium
963	2002	0	0	1	0	medium
963	2003	0	0	1	0	medium
1525	2006	1	1	1	0	large
1525	2007	1	1	1	0	vlarge
1525	2008	1	1	1	1	vlarge
1525	2009	1	1	1	1	vlarge
1525	2010	1	1	1	1	vlarge
1534	2006	1	1	1	0	vlarge

The variables `surplus`, `istock`, `icap`, and `debtr` are the gross operating surplus, initial stock in Euros, initial capital in Euros, and the debt ratio of the farm in percent, respectively. The other (indicator) variables (`crop` = 1 if the farm specializes in field crop production, `corp` = 1 if the farm has corporate legal status, `educ` = 1 if the farmer has a higher-level education, and `young` = 1 if the farmer is under 41) were derived from original variables for our specific examples.[4]

4. Note that the examples presented in this article are for illustration purposes only and should not be considered to produce sound economic conclusions. They are meant only to illustrate the use of the `mixmcm` command using real data.

4.1 Fitting a simple two-component MMCM

To illustrate the basic use of `mixmcm`, we first fit a two-component MMCM without considering entries or exits. In this example, we specify transition probabilities as a function of four explanatory variables, namely, `surplus`, `istock`, `crop`, and `corp`. The parameters of the mixing distribution are estimated nonparametrically. As mentioned in section 2.1, the resulting model is therefore considered to be semiparametric.

```
. mixmcm category surplus istock crop corp, id(idnum) timevar(year)
(Warning: unbalanced panel data)

Estimating a 2-components mixture of discrete-state Markov chain model

Searching for initial values ...

log-likelihood = -15350.904                        Number of obs   = 78434
                                                   Number of id    = 13123
```

	coef.	Robust Std. Err.	z	p>|z|	[95% Conf.	Interval]
Component 1						
Transition probabilities						
large	initial state					
medium						
surplus	-.0000147	2.10e-06	-7.00	0.000	-.0000189	-.0000106
istock	-.0000334	2.22e-06	-15.03	0.000	-.0000377	-.000029
crop	-.4061981	.118146	-3.44	0.001	-.6377643	-.1746319
corp	-.159711	.1088367	-1.47	0.142	-.373031	.053609
_cons	-2.117094	.1562513	-13.55	0.000	-2.423347	-1.810842
vlarge						
surplus	3.11e-06	1.05e-06	2.95	0.003	1.05e-06	5.18e-06
istock	3.06e-07	4.52e-07	0.68	0.499	-5.81e-07	1.19e-06
crop	.5234549	.0964274	5.43	0.000	.3344571	.7124527
corp	-.1790304	.0978987	-1.83	0.067	-.3709119	.0128511
_cons	-4.812017	.1131314	-42.53	0.000	-5.033755	-4.590279
medium	initial state					
large						
surplus	8.46e-06	2.80e-06	3.03	0.002	2.98e-06	.0000139
istock	3.05e-07	1.82e-06	0.17	0.867	-3.26e-06	3.87e-06
crop	-.8878777	.1604371	-5.53	0.000	-1.202334	-.573421
corp	7.265593	.1092183	66.52	0.000	7.051525	7.479661
_cons	-10.82393	.0941205	-115.00	0.000	-11.00841	-10.63946
vlarge						
surplus	.0000194	4.80e-06	4.04	0.000	9.98e-06	.0000288
istock	-.0000172	.0000109	-1.59	0.113	-.0000385	4.07e-06
crop	14.74219	.3955852	37.27	0.000	13.96684	15.51754
corp	.5576473	.5253997	1.06	0.289	-.4721362	1.587431
_cons	-21.38917	.3700815	-57.80	0.000	-22.11453	-20.66381

vlarge	initial state					
large						
surplus	-6.64e-06	9.60e-07	-6.92	0.000	-8.53e-06	-4.76e-06
istock	-.0000105	8.83e-07	-11.93	0.000	-.0000123	-8.81e-06
crop	-.0625092	.1234159	-0.51	0.613	-.3044043	.179386
corp	-1.07522	.1237099	-8.69	0.000	-1.317692	-.8327486
_cons	-1.966381	.1396501	-14.08	0.000	-2.240095	-1.692666
medium						
surplus	-7.09e-06	1.63e-06	-4.36	0.000	-.0000103	-3.90e-06
istock	-.0000249	6.31e-06	-3.94	0.000	-.0000373	-.0000125
crop	.6342508	.4561181	1.39	0.164	-.2597407	1.528242
corp	-.5315547	.4040807	-1.32	0.188	-1.323553	.2604434
_cons	-3.926738	.4783556	-8.21	0.000	-4.864315	-2.989161

Component 2

Transition probabilities

large	initial state					
medium						
surplus	-.0000177	1.07e-06	-16.49	0.000	-.0000198	-.0000156
istock	-5.43e-06	9.83e-07	-5.52	0.000	-7.35e-06	-3.50e-06
crop	-.3810236	.0677824	-5.62	0.000	-.513877	-.2481701
corp	-.3540718	.0726781	-4.87	0.000	-.4965208	-.2116228
_cons	-.3752858	.0827246	-4.54	0.000	-.537426	-.2131457
vlarge						
surplus	.0000111	8.29e-07	13.36	0.000	9.46e-06	.0000127
istock	1.58e-07	4.27e-07	0.37	0.711	-6.79e-07	9.96e-07
crop	-.4427715	.074652	-5.93	0.000	-.5890895	-.2964536
corp	.650648	.0832685	7.81	0.000	.4874418	.8138542
_cons	-3.383384	.0912461	-37.08	0.000	-3.562227	-3.204542

medium	initial state					
large						
surplus	.0000136	1.47e-06	9.24	0.000	.0000107	.0000165
istock	3.09e-06	1.78e-06	1.74	0.082	-3.94e-07	6.57e-06
crop	-.00887	.0658711	-0.13	0.893	-.1379774	.1202375
corp	.3264058	.0740129	4.41	0.000	.1813406	.471471
_cons	-2.242076	.0892614	-25.12	0.000	-2.417028	-2.067124
vlarge						
surplus	-1.87e-06	.000011	-0.17	0.865	-.0000234	.0000197
istock	-.0000354	.0000115	-3.08	0.002	-.000058	-.0000129
crop	-1.661568	.4817014	-3.45	0.001	-2.605703	-.7174328
corp	1.387473	.4821417	2.88	0.004	.4424748	2.33247
_cons	-4.322117	.4144229	-10.43	0.000	-5.134386	-3.509848

vlarge	initial state					
large						
surplus	-6.73e-06	6.85e-07	-9.82	0.000	-8.07e-06	-5.39e-06
istock	-2.62e-06	5.31e-07	-4.93	0.000	-3.66e-06	-1.58e-06
crop	-.3536559	.0825228	-4.29	0.000	-.5154005	-.1919112
corp	-.0087921	.091816	-0.10	0.924	-.1887514	.1711672
_cons	-.2200508	.1072874	-2.05	0.040	-.4303342	-.0097674
medium						
surplus	-.0000181	5.03e-06	-3.61	0.000	-.000028	-8.28e-06
istock	4.18e-07	1.97e-06	0.21	0.832	-3.45e-06	4.29e-06
crop	-1.427922	.4640655	-3.08	0.002	-2.337491	-.518354
corp	-1.832979	.4402466	-4.16	0.000	-2.695862	-.9700953
_cons	-1.929111	.5617308	-3.43	0.001	-3.030103	-.8281189

Type shares

	Mean	Std. Dev.
pi1	.7233329	.2715939
pi2	.2766671	.2715939

`Model estimated via expectation-maximization (EM) algorithm.`

The table above reports the resulting coefficients of the explanatory variables for each component (that is, the degree to which they contribute to explaining the odds ratios, where remaining in the initial state is the reference scenario). As such, the values of the coefficients are not directly interpretable (see section 5 for the derivation of elasticities). If one focuses on the signs, however, it is evident that the contributions of the coefficients differ across components. For example, the variable `corp` has a negative (-0.179) effect on the odds ratio $\{P(\texttt{large} \rightarrow \texttt{vlarge})\}/\{P(\texttt{large} \rightarrow \texttt{large})\}$ for the first component, while this impact is positive (0.651) for the second component. The parameters `pi1` and `pi2`, respectively, are the resulting shares of type 1 and type 2 in the studied sample.

The standard errors reported in the table are obtained from the official Stata command `mlogit`, using the option `vce(robust)` in the model specification. The variance–covariance matrix is thus obtained by the Huber/White/sandwich estimator. Indeed, `mixmcm` performs multiple weighted multinomial logit estimations according to the number of types × number of initial states specified. Each weighted multinomial logit is estimated separately. For each estimation, we use the robust or sandwich estimator of the variance–covariance matrix, assuming that explanatory variables ($\mathbf{X}_{it}$) and the error terms (ϵ_{it}) are uncorrelated. With these assumptions and a few technical regularity conditions, each weighted `mlogit` yields consistent parameter estimates and standard errors.[5] We can thus use the resulting `mixmcm` parameter estimates and standard errors for valid statistical inference about the coefficients (see section 5).

5. See [U] **20.21 Obtaining robust variance estimates** in Stata 15 for more details on the properties of the robust variance estimate.

4.2 Specifying entry, exit, type membership, and constraints on transition probabilities

In this second example, we fit a two-component MMCM that includes entry and exit. We also specify a set of explanatory variables for entry and type membership probabilities and impose constraints on some transition probabilities.

We now consider farms that leave the sample before the final year of the panel (2010) as exits. We thus add a new category, `exits`, to the variable `category`.

```
. by idnum: generate _last = _n == _N
. drop if _last != 1
. keep idnum year
. by idnum: replace year= year + 1 if _n == _N
. append using "mixmcm.dta"
. sort idnum year
. drop if year > 2010
. replace category = "exits" if category == ""
```

Similarly, we now consider farms that enter the sample after the first year of the panel (2000) as entries. We thus generate a new variable, `entry_class`, that indicates the category in which farms are observed for the first time in the sample.

```
. by idnum: generate _first = _n == 1
. by idnum: generate str entry_class = "1" if _first == 1 & year != 2000
. levelsof category, local(catlevels)
. foreach cat of local catlevels {
  2. replace entry_class = "`cat´" if category == "`cat´" & entry_class == "1"
  3. }
. replace entry_class = "." if entry_class == ""
. drop _first
```

We also generate new variables to be used in the specification of the type-membership probabilities. To ensure that these probabilities do not vary over time, we take the mean of the continuous variable `debtr` and the mode of the dummy variables `educ` and `young`.

```
. by idnum: egen double meandebtr = mean(debtr)
. foreach v in educ young {
  2.         by idnum: egen double mode`v´ = mode(`v´)
  3.         by idnum: replace mode`v´ = `v´[_N] if mode`v´ == .
  4. }
```

Finally, we specify two constraints on the transition probabilities: the probability of moving from the `medium` category to the `vlarge` category is set to zero and vice versa.

```
. constraint 1 p_medium_vlarge = 0
. constraint 2 p_vlarge_medium = 0
```

The MMCM under the above specification is therefore fit as

```
. mixmcm category surplus istock crop corp, id(idnum) timevar(year)
> exitcode(exits) entry(entry_class icap corp)
> membership(meandebt modeeduc modeyoung) constraints(1 2)
(Warning: unbalanced panel data)

Estimating a 2-components mixture of discrete-state Markov chain model

Searching for initial values ...
(Warning: EM not converged)

(Warning: EM not converged)

(Warning: EM not converged)

log-likelihood = -41388.273                          Number of obs  = 84886
                                                     Number of id   = 13123
```

	coef.	Robust Std. Err.	z	p>|z|	[95% Conf.	Interval]
Component 1						
Entry probabilities						
large	(baseoutcome)					
medium						
icap	0 (omitted)					
corp	0 (omitted)					
_cons	0 (omitted)					
vlarge						
icap	6.53e-06	3.62e-07	18.02	0.000	5.82e-06	7.24e-06
corp	4.196947	.1298916	32.31	0.000	3.942359	4.451535
_cons	-5.467211	.1362706	-40.12	0.000	-5.734301	-5.20012
Transition probabilities						
large	initial state					
medium						
surplus	-.0000173	1.69e-06	-10.24	0.000	-.0000207	-.000014
istock	-9.96e-07	1.38e-06	-0.72	0.469	-3.69e-06	1.70e-06
crop	-.1183684	.1134126	-1.04	0.297	-.3406571	.1039204
corp	-.3344715	.1103856	-3.03	0.002	-.5508274	-.1181157
_cons	-3.23115	.1600785	-20.18	0.000	-3.544904	-2.917396
vlarge						
surplus	1.41e-06	1.74e-06	0.81	0.418	-2.00e-06	4.82e-06
istock	1.08e-06	5.44e-07	1.99	0.046	1.88e-08	2.15e-06
crop	5.971234	.1515587	39.40	0.000	5.674179	6.268289
corp	.0391631	.1421559	0.28	0.783	-.2394624	.3177886
_cons	-11.13387	.2006241	-55.50	0.000	-11.52709	-10.74065
exits						
surplus	2.25e-06	5.84e-07	3.86	0.000	1.11e-06	3.40e-06
istock	3.69e-07	2.90e-07	1.27	0.203	-1.99e-07	9.38e-07
crop	-.256642	.0468104	-5.48	0.000	-.3483904	-.1648935
corp	-.5012324	.0479034	-10.46	0.000	-.5951231	-.4073418
_cons	-2.213591	.0512114	-43.22	0.000	-2.313965	-2.113217

medium	initial state					
large						
surplus	7.08e-06	1.98e-06	3.58	0.000	3.20e-06	.000011
istock	1.40e-06	9.52e-07	1.47	0.142	-4.69e-07	3.26e-06
crop	-.7205185	.1367084	-5.27	0.000	-.988467	-.45257
corp	3.242979	.1117609	29.02	0.000	3.023928	3.462031
_cons	-6.69457	.0963498	-69.48	0.000	-6.883416	-6.505725
vlarge						
surplus	0 (omitted)					
istock	0 (omitted)					
crop	0 (omitted)					
corp	0 (omitted)					
_cons	0 (omitted)					
exits						
surplus	-1.02e-06	1.24e-06	-0.82	0.410	-3.45e-06	1.41e-06
istock	-1.91e-06	9.53e-07	-2.01	0.045	-3.78e-06	-4.45e-08
crop	-.0092237	.0443347	-0.21	0.835	-.0961196	.0776723
corp	-.4505092	.0684921	-6.58	0.000	-.5847537	-.3162648
_cons	-2.031039	.0521962	-38.91	0.000	-2.133343	-1.928734
vlarge	initial state					
large						
surplus	-5.51e-06	8.61e-07	-6.40	0.000	-7.20e-06	-3.82e-06
istock	2.74e-07	3.50e-07	0.78	0.434	-4.13e-07	9.60e-07
crop	.4757005	.1788323	2.66	0.008	.1251892	.8262118
corp	-1.390182	.1674019	-8.30	0.000	-1.71829	-1.062074
_cons	-3.466531	.2157733	-16.07	0.000	-3.889447	-3.043615
medium						
surplus	0 (omitted)					
istock	0 (omitted)					
crop	0 (omitted)					
corp	0 (omitted)					
_cons	0 (omitted)					
exits						
surplus	1.34e-06	3.48e-07	3.85	0.000	6.58e-07	2.02e-06
istock	3.87e-07	1.02e-07	3.79	0.000	1.87e-07	5.87e-07
crop	-.2891374	.0809938	-3.57	0.000	-.4478852	-.1303895
corp	-.9858392	.0846571	-11.65	0.000	-1.151767	-.8199113
_cons	-1.923489	.0906852	-21.21	0.000	-2.101232	-1.745746
Component 2						
Entry probabilities						
large	(baseoutcome)					
medium						
icap	-1.74e-06	3.40e-07	-5.12	0.000	-2.41e-06	-1.07e-06
corp	-1.201836	.089677	-13.40	0.000	-1.377603	-1.026069
_cons	-.1009562	.0637887	-1.58	0.113	-.225982	.0240696
vlarge						
icap	1.89e-06	2.34e-07	8.08	0.000	1.43e-06	2.35e-06
corp	.5776288	.0875169	6.60	0.000	.4060958	.7491619
_cons	-1.080876	.0834172	-12.96	0.000	-1.244374	-.9173786

Transition probabilities

large	initial state					
medium						
surplus	-.0000157	1.02e-06	-15.35	0.000	-.0000177	-.0000137
istock	-9.58e-06	9.01e-07	-10.63	0.000	-.0000113	-7.81e-06
crop	-.4394746	.0679593	-6.47	0.000	-.5726749	-.3062743
corp	-.3699244	.0723282	-5.11	0.000	-.5116876	-.2281612
_cons	-.5153514	.0773583	-6.66	0.000	-.6669736	-.3637292
vlarge						
surplus	9.48e-06	7.42e-07	12.78	0.000	8.03e-06	.0000109
istock	1.91e-06	3.98e-07	4.79	0.000	1.13e-06	2.69e-06
crop	-.3887311	.0704429	-5.52	0.000	-.5267992	-.2506631
corp	.4924593	.0769143	6.40	0.000	.3417073	.6432113
_cons	-3.290045	.0845741	-38.90	0.000	-3.45581	-3.12428
exits						
surplus	9.79e-07	8.31e-07	1.18	0.238	-6.49e-07	2.61e-06
istock	-2.51e-06	4.84e-07	-5.18	0.000	-3.46e-06	-1.56e-06
crop	-.2224568	.053576	-4.15	0.000	-.3274657	-.1174478
corp	-.5241336	.0578154	-9.07	0.000	-.6374517	-.4108154
_cons	-1.785737	.0674397	-26.48	0.000	-1.917919	-1.653555
medium	**initial state**					
large						
surplus	.0000115	1.36e-06	8.44	0.000	8.85e-06	.0000142
istock	9.45e-06	9.85e-07	9.59	0.000	7.52e-06	.0000114
crop	-.1324843	.0674814	-1.96	0.050	-.2647478	-.0002208
corp	.2581651	.0750526	3.44	0.001	.1110621	.4052682
_cons	-2.24129	.0766323	-29.25	0.000	-2.391489	-2.09109
vlarge						
surplus	0 (omitted)					
istock	0 (omitted)					
crop	0 (omitted)					
corp	0 (omitted)					
_cons	0 (omitted)					
exits						
surplus	-.0000116	1.76e-06	-6.59	0.000	-.000015	-8.12e-06
istock	-8.68e-06	1.68e-06	-5.15	0.000	-.000012	-5.38e-06
crop	-.1185502	.0607494	-1.95	0.051	-.2376191	.0005186
corp	-.111747	.083309	-1.34	0.180	-.2750327	.0515387
_cons	-1.088767	.0755732	-14.41	0.000	-1.236891	-.9406437

vlarge	initial state					
large						
surplus	-4.62e-06	6.07e-07	-7.61	0.000	-5.80e-06	-3.43e-06
istock	-5.23e-06	4.81e-07	-10.88	0.000	-6.17e-06	-4.29e-06
crop	-.4181612	.0799961	-5.23	0.000	-.5749535	-.2613688
corp	.1599688	.0853567	1.87	0.061	-.0073304	.327268
_cons	-.9184241	.0978913	-9.38	0.000	-1.110291	-.7265572
medium						
surplus	0 (omitted)					
istock	0 (omitted)					
crop	0 (omitted)					
corp	0 (omitted)					
_cons	0 (omitted)					
exits						
surplus	4.72e-07	4.08e-07	1.16	0.247	-3.28e-07	1.27e-06
istock	1.35e-07	1.17e-07	1.15	0.248	-9.44e-08	3.65e-07
crop	-.2717015	.0777804	-3.49	0.000	-.4241512	-.1192518
corp	-.3866597	.0814266	-4.75	0.000	-.5462558	-.2270636
_cons	-2.025348	.0875228	-23.14	0.000	-2.196893	-1.853803

Type shares

	Mean	Std. Dev.
pi1	.6512266	.3125167
pi2	.3487734	.3125167

Membership probabilities

	coef.	Robust Std. Err.	z	p>|z|	[95% Conf.	Interval]
proba1	(baseoutcome)					
proba2						
meandebtr	.0221298	.0001892	116.99	0.000	.021759	.0225006
modeeduc	.2498734	.0203545	12.28	0.000	.2099786	.2897681
modeyoung	.0804814	.0115482	6.97	0.000	.0578469	.1031158
_cons	-1.833442	.021202	-86.48	0.000	-1.874998	-1.791886

Model estimated via expectation-maximization (EM) algorithm.

Compared with the first example, this one has three additional sets of parameters. First, the parameters for the exit state now appear in the table of results. Because entries are estimated separately, the total number of initial states remains the same as in example 1.

Second, the parameters for the probability of entering in a given size category are reported in the section `Entry probabilities` of the table of results. The coefficients represent the contribution of the explanatory variables in explaining the odds ratios, with the `large` category as the reference.[6] The parameters for the odds ratio $\{P(\texttt{medium})\}/\{P(\texttt{large})\}$ in component 1 are set to zero, meaning that there are not enough observations for the identification of these parameters in this component. As in example 1, the effect of the explanatory variables also differs across components. For example, the results show that the variable `corp` has a larger positive effect on the odds ratio $\{P(\texttt{vlarge})\}/\{P(\texttt{large})\}$ for farms that belong to the first component. The estimated coefficient is 4.197 for farms belonging to component 1 and only 0.578 for farms belonging to component 2.

Third, the parameters for the membership probabilities are given in the `Membership probabilities` section of the results table, after the resulting type shares `pi1` and `pi2`. The coefficients represent the contribution of the explanatory variables in explaining the odds ratios using the probability of belonging to component 1 as the reference. All three chosen variables (`meandebtr`, `modeeduc`, and `modeyoung`) have a positive effect on the probability of belonging to component 2 versus component 1.

Finally, one can verify that the parameters for the transition probabilities that were constrained to zero are not estimated by the model, because they appear as set to zero and indicated as `(omitted)` in the table of results.

4.3 Choosing the optimal number of components

In this last example, we illustrate how the number of components can be chosen by relying on statistical information rather than a priori knowledge on the total number of components as in the previous examples. In this case, the criteria used to select the most relevant model are generally based on the value of $-2\text{LL}_G(\mathbf{y};\widehat{\mathbf{\Phi}})$ of the model, where $\text{LL}_G(\mathbf{y};\widehat{\mathbf{\Phi}})$ represents the maximum LL estimate adjusted for the number of free parameters in the model. The basic principle here is parsimony: all other things being equal, the model with fewer parameters should be preferred. In the case of latent class models such as the MMCM, the selection of the total number of types G is generally based on the following criterion (Andrews and Currim 2003; Dias and Willekens 2005)

$$C_G = -2\text{LL}_G\left(\mathbf{y};\widehat{\mathbf{\Phi}}\right) + \kappa \times N_G$$

where N_G is the total number of free parameters of a model with G types. Different values for the penalizing factor κ lead to the various well known information criteria: the Akaike information criterion (AIC) with $\kappa = 2$ and the Bayesian information criterion (BIC) with $\kappa = \log(n)$ (Andrews and Currim 2003). Other information criteria can also be derived such as the consistent Akaike information criterion (CAIC) with $\kappa = \log(n)+1$

6. Note that `mixmcm` estimates the effect of explanatory variables on the probability of entering a specific state conditional on the agent having entered the sample. Other assumptions and calculations are necessary to derive the probabilities of new entries when analyzing population dynamics.

and a modified AIC (AIC3) with $\kappa = 3$ (Andrews and Currim 2003). For each of these heuristic criteria, smaller values indicate more parsimonious models.

As an illustration, we search for the optimal model for our RICA data by letting the number of unobserved types vary within the range of one to five components. We also set alternative numbers of short-run and long-run EM iterations to increase the chances of obtaining a global maximum in each case. We choose the optimal number of components based on the CAIC criterion (the default), but a graph is drawn for all available information criteria. Finally, we also save the estimated parameters for all models in a file that can be used for further analysis. To do so, we use `mixmcm` as follows:

```
. mixmcm category surplus istock crop corp, id(idnum) timevar(year)
> ncomponents(1 5, graph(aic bic caic aic3, title("Fig. 1")
> ytitle("Information criteria") xtitle("Number of components") xlabel(1(1)5)
> scheme(sj) saving(figure.eps, replace)) save(icbtable, replace detail) force)
> emiterate(lr(3 200, 0.000001) sr(3 5)) exitcode(exits)
> entry(entry_class icap corp) membership(meandebtr modeeduc modeyoung)
> constraints(1 2)
```

(*output omitted*)

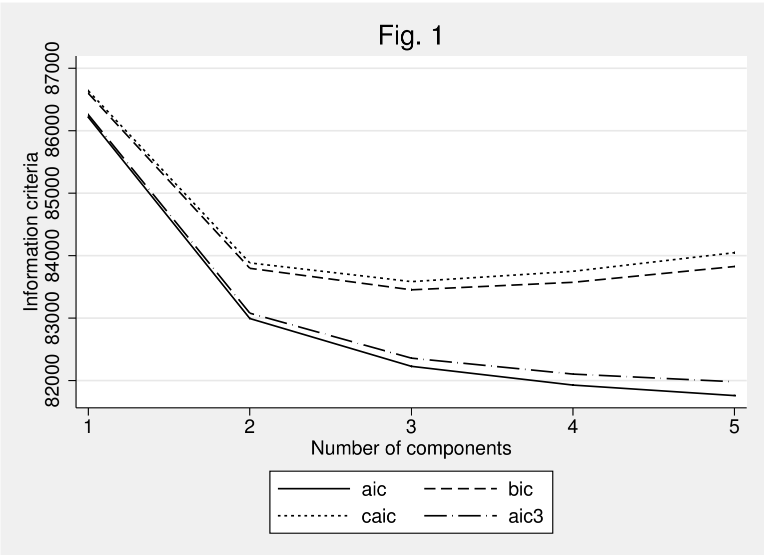

Figure 1. Comparison of statistical information criteria for different numbers of unobserved farm types

Figure 1 shows that, based on the model specification and according to the CAIC, a mixture of three types of farms should be chosen as the optimal number of components. It also shows that the BIC is consistent with the CAIC in identifying three components while, according to the AIC and AIC3, a higher number of farm types should be preferred.

In these two latter cases, however, the improvement in the values of the criteria is relatively small when specifying more than three types so that three components may overall be a relevant compromise.

The command output displays the results regarding the optimal number of components. The corresponding table is not reported here because it is essentially similar to that of example 2, the difference being that it reports the results for three components instead of two. Additionally, because we use the suboption `force` for the `ncomponents()` option, the information criteria and the full set of estimated parameters for all fitted models (from one to five components) are saved to disk in the specified filename `icbtable`; an excerpt of this file is provided in the appendix. Finally, `mixmcm` stores the following in `e()`:

Scalars

`e(N_components)`	optimal number of homogeneous types
`e(min_components)`	minimum number of homogeneous types estimate
`e(max_components)`	maximum number of homogeneous types estimate
`e(ll)`	log likelihood
`e(N)`	number of observations
`e(N_id)`	number of agents identified by `id()`
`e(k)`	number of free parameters estimated
`e(aic)`	Akaike information criterion
`e(aic3)`	restricted Akaike information criterion
`e(bic)`	Bayesian information criterion
`e(caic)`	consistent Akaike information criterion
`e(converged)`	`1` if the EM algorithm converged, `0` otherwise

Macros

`e(cmd)`	`mixmcm` command name
`e(cmdline)`	command line as it was written
`e(depvar)`	name of the dependent variable
`e(title)`	title in estimation output
`e(id)`	name of the `id()` variable
`e(states)`	modalities of the dependent variable
`e(exitcode)`	name of the exit state
`e(indepvars)`	independent variables for transition probabilities
`e(entry_var)`	name of the `entry()` variable
`e(entry_indepvars)`	independent variables for entry probabilities
`e(compvars)`	independent variables for component membership probabilities
`e(mpf)`	functional form of the mixing distribution
`e(selcrit)`	information criterion for the selection of the optimal number of components
`e(seed)`	pseudouniform random-number seed

Matrices

`e(b_tpm)`	vector of coefficients for entry and transition probabilities
`e(V_tpm)`	covariance matrix of the coefficients for entry and transition probabilities
`e(b_proba)`	vector of coefficients for component membership probabilities
`e(V_proba)`	covariance matrix of the coefficients for component membership coefficients
`e(pi)`	vector of component shares
`e(Cns_tpm)`	matrix of constraints

The estimated coefficients and the variance–covariance matrices for the transition probabilities and for component membership are given in matrices `e(b_tpm)`, `e(V_tpm)`, `e(b_proba)`, and `e(V_proba)`, respectively. The matrices `e(pi)` and `e(Cns_tpm)` contain the components' shares and the constraints on transition probabilities, respectively. Standard deviations are also reported for the shares of each type component in the matrix `e(pi)`. The elements of matrix `e(Cns_tpm)` take the value 1 if the respective transition probability is constrained to be 0. Otherwise, the elements of the matrix are set to 0.

5 Hypothesis testing, transition probability, and elasticity derivation

Because the parameters estimated by the `mixmcm` command usually are not directly interpretable in applied work, this section demonstrates how to derive empirically useful information from these estimated parameters. For example, they can be used to conduct hypothesis testing to compare specific parameters across components. They can also be used to compute transition probabilities and derive probability elasticities and structure elasticities.

To perform postestimation calculations for hypothesis testing, transition probability computation, and elasticity derivation, we must first put the vector of coefficients and the variance–covariance matrix in `b` and `V`, respectively. To do this, we can use the following subroutine:

```
. matrix BETA=e(b_tpm)
. matrix COVB=e(V_tpm)
. matrix SHARE=e(pi)
. capture program drop fill_bV
. program define fill_bV, eclass
  1.                 ereturn clear
  2.                 matrix b = BETA
  3.                 matrix V = COVB
  4.                 ereturn post b V
  5. end
```

Then, the above subroutine must first be called just before using any Stata postestimation command for the calculations. In the following, we present how to perform hypothesis testing, transition probability, and elasticity derivation for example 1 but the method would apply to any MMCM model fit with the `mixmcm` command.

5.1 Hypothesis testing

Wald tests of simple and composite linear hypotheses about the parameters of the model can be performed to conduct hypothesis testing. For example, users can test whether the estimated parameters are different from zero as follows:

i) A test on a specific parameter of a specific transition for a specific component.

```
. test [p_1_large_vlarge]:L.surplus
 ( 1)  [p_1_large_vlarge]L.surplus = 0
           chi2(  1) =    8.72
         Prob > chi2 =    0.0032
```

ii) A joint test on all the parameters of a specific transition and a specific component.

```
. test [p_2_medium_vlarge]:L.surplus L.istock L.crop L.corp
 ( 1)  [p_2_medium_vlarge]L.surplus = 0
 ( 2)  [p_2_medium_vlarge]L.istock = 0
 ( 3)  [p_2_medium_vlarge]L.crop = 0
 ( 4)  [p_2_medium_vlarge]L.corp = 0
           chi2(  4) =   23.84
         Prob > chi2 =    0.0001
```

iii) A joint test on all the parameters of all the transitions for all components.

```
. test L.surplus L.istock L.crop L.corp
 ( 1)  [p_1_large_large]oL.surplus = 0
 ( 2)  [p_1_large_medium]L.surplus = 0
 ( 3)  [p_1_large_vlarge]L.surplus = 0
 ( 4)  [p_1_medium_large]L.surplus = 0
 ( 5)  [p_1_medium_medium]oL.surplus = 0
  (output omitted)
```

For the last example, we report only the first five lines of the output to save space. The Wald test is performed for all the estimated parameters of all the transitions, including the base outcome identified with the prefix `o` for each explanatory variable.

In any of the specific examples above, considered Wald tests show that the tested parameters all are different from 0 at the 1% significance level at least.[7]

Users may also test whether the same parameters for two different components are different from each other, that is, whether the MMCM identify agent types characterized by specific parameter values.

i) A test on specific parameters for specific transition probabilities across two specific components.

```
. test [p_1_large_vlarge=p_2_large_vlarge]:L.surplus
 ( 1)  [p_1_large_vlarge]L.surplus - [p_2_large_vlarge]L.surplus = 0
           chi2(  1) =   35.30
         Prob > chi2 =    0.0000
```

7. See, for instance, [R] **test** for a formal interpretation of the results of the Wald test of simple and composite linear hypotheses about the parameters from the fit model.

ii) A joint test on all the parameters for a specific transition across two specific components.

```
. test [p_1_large_vlarge=p_2_large_vlarge]
 ( 1)  [p_1_large_vlarge]L.surplus - [p_2_large_vlarge]L.surplus = 0
 ( 2)  [p_1_large_vlarge]L.istock - [p_2_large_vlarge]L.istock = 0
 ( 3)  [p_1_large_vlarge]L.crop - [p_2_large_vlarge]L.crop = 0
 ( 4)  [p_1_large_vlarge]L.corp - [p_2_large_vlarge]L.corp = 0
           chi2(  4) =  182.63
         Prob > chi2 =    0.0000
```

In these specific examples, the considered Wald test shows that the two components, that is, the two endogenous types of farms, are characterized by different estimated parameter values at the 0.1% significance level at least for the probability to move from category `large` to the category `vlarge`. Of course, such tests can be performed for any transition and whatever the number of components.

5.2 Transition probabilities

As `mixmcm` performs multiple multinomial logistic regressions to estimate the parameters, the `predict` Stata command cannot be used to directly derive transition probabilities.[8] Considering the results from example 1, we can compute transition probabilities using (3). To save space, we report the code lines for the calculations in the appendix. The transition probabilities are reported below.

```
. forvalues s=1/2 {
  2. display _newline
  3. summarize pr_`s´_medium_medium pr_`s´_medium_large pr_`s´_medium_vlarge
>          pr_`s´_large_medium pr_`s´_large_large pr_`s´_large_vlarge
>          pr_`s´_vlarge_medium pr_`s´_vlarge_large pr_`s´_vlarge_vlarge,
>          separator(3)
  4. }
```

Variable	Obs	Mean	Std. Dev.	Min	Max
pr_1_mediu~m	23,458	.9929139	.0152545	.4269942	.9999997
pr_1~m_large	23,458	.0061466	.014383	2.79e-07	.466761
pr_1_mediu..	23,458	.0009395	.00304	8.21e-16	.2149646
pr_1_large~m	29,022	.0091439	.0137811	1.15e-29	.7265102
pr_1_large..	29,022	.9782463	.0136064	.272103	.9938283
pr_1_large..	29,022	.0126098	.0042574	.0013868	.092576
pr_1_vlarg~m	12,831	.0023512	.0051253	6.03e-54	.2039687
pr_1_vlarg..	12,831	.0145335	.0213587	5.44e-24	.4155616
pr_1_vlarg..	12,831	.9831152	.0258175	.3804698	1

8. A postestimation command for `mixmcm` is under construction and will be made available for the users to more easily perform predictions and derive other parameters of interest.

Variable	Obs	Mean	Std. Dev.	Min	Max
pr_2_mediu~m	23,458	.8208468	.0711365	.00101	.9979493
pr_2~m_large	23,458	.1753561	.0713373	.0020507	.99899
pr_2_mediu..	23,458	.0037971	.0043659	3.95e-35	.0426982
pr_2_large~m	29,022	.0990758	.0733726	4.99e-10	.9912049
pr_2_large..	29,022	.8117434	.0625643	.0087897	.9795649
pr_2_large..	29,022	.0891808	.0685449	5.44e-06	.9907419
pr_2_vlarg~m	12,831	.0057708	.0182524	6.14e-13	.7566274
pr_2_vlarg..	12,831	.176613	.0942067	3.48e-07	.6508636
pr_2_vlarg..	12,831	.8176161	.1012113	.0144417	.9999994

The results in the table above show that transition probabilities differ across the two considered components. The farms in type 1 exhibit a very high probability to remain in the same category of size two consecutive years (above 0.95), whatever their initial category. These probabilities are lower for farms in type 2 (below 0.85), which means that those farms are more likely to move across the size categories over the years.

The above procedure can be applied to compute entry probability from (2) and probability of type membership from (4) if a parametric form is used for the mixing distribution.[9]

5.3 Probability elasticities

As mentioned earlier (see section 4.1), the impacts of explanatory variables on log-odds ratios are difficult to interpret directly (Greene 2018). In this case, the impacts of explanatory variables are rather given with respect to transition probabilities, as measured in terms of elasticities. "Probability elasticities" thus measure the (relative) effect of a 1% change in the ith explanatory variable on the transition probabilities (Zepeda 1995). In our mixture model, average transition-probability elasticities from one period to the next for farms belonging to a specific type g can be derived as

$$\boldsymbol{\delta}_{jk|g} = \left(\boldsymbol{\beta}_{jk|g} - \sum_{l=1}^{K} \boldsymbol{\beta}_{jl|g} p_{jl|g}\right) \overline{\mathbf{x}}, \qquad \forall j,k \in K, \quad \forall g \in G$$

where $\boldsymbol{\delta}_{jk|g}$ is a vector gathering elasticities at the means over the period $T-1$ of the explanatory variables in vector $\mathbf{x}$ on the transition probability from category j to category k conditional on belonging to type g ($p_{jk|g}$); and $\boldsymbol{\beta}_{jk|g}$ is the corresponding vector of estimated parameters.[10]

9. Stata codes are given in the appendix to derive probability membership when a nonparametric form is used for the mixing distribution.

10. Note that yearly transition-probability elasticities can be also derived using the following equation:

$$\boldsymbol{\delta}_{jkt|g} = \left(\boldsymbol{\beta}_{jk|g} - \sum_{l=1}^{K} \boldsymbol{\beta}_{jl|g} p_{jlt|g}\right) \overline{\mathbf{x}}_{t-1}$$

Considering again example 1, we can use the transition probabilities computed in section 5.2 to derive transition-probability elasticities using the `nlcom` Stata command as follows:

```
. /* Save transition probabilities as scalars */
. forvalues s=1/2 {
  2. foreach ist in medium large vlarge {
  3.   foreach fst in medium large vlarge {
  4.     quietly summarize pr_`s´_`ist´_`fst´ if category[_n-1]=="`ist´"
  5.       scalar prmean_`s´_`ist´_`fst´ = r(mean)
  6.     }
  7.   }
  8. }
. /* Compute means of explanatory variables over the period (T-1) */
. foreach x in surplus istock crop corp {
  2. quietly summarize `x´ if year < 2010
  3. scalar mean_`x´ = r(mean)
  4. }
. /* Derive transition-probability elasticities */
. forvalues s=1/2 {
  2. foreach ist in medium large vlarge {
  3.   foreach x in surplus istock crop corp {
  4.     foreach fst in medium large vlarge {
  5.       local sum_`s´_`ist´_`x´ "`sum_`s´_`ist´_`x´´-_b
> [p_`s´_`ist´_`fst´:L.`x´]*prmean_`s´_`ist´_`fst´"
  6.     }
  7.     local sum_`s´_`ist´_`x´ `=substr("`sum_`s´_`ist´_`x´´",2,.)´
  8.   }
  9.   fill_bV // call the subroutine to put the mixmcm estimates in b and V
 10.   nlcom (prelast_`s´_`ist´_medium_surplus:(_b[p_`s´_`ist´_medium:L.surplus]
> -`sum_`s´_`ist´_surplus´) * mean_surplus)
> (prelast_`s´_`ist´_large_surplus:(_b[p_`s´_`ist´_large:L.surplus]-
> `sum_`s´_`ist´_surplus´) * mean_surplus)
> (prelast_`s´_`ist´_vlarge_surplus:(_b[p_`s´_`ist´_vlarge:L.surplus]-
> `sum_`s´_`ist´_surplus´) * mean_surplus), noheader
 11.   }
 12. }
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
prelast_1_m~s	-.0048678	.0012316	-3.95	0.000	-.0072818	-.0024539
prelast_1_m~s	.581688	.1926161	3.02	0.003	.2041674	.9592087
prelast_1_m~s	1.338917	.3325685	4.03	0.000	.6870949	1.99074

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
prelast_1_l~s	-1.014097	.1444923	-7.02	0.000	-1.297297	-.7308972
prelast_1_l~s	.0066127	.0016208	4.08	0.000	.0034359	.0097894
prelast_1_l~s	.222363	.0721659	3.08	0.002	.0809204	.3638056

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
prelast_1_v~s	-.4832283	.1124587	-4.30	0.000	-.7036433	-.2628134
prelast_1_v~s	-.4526748	.0655823	-6.90	0.000	-.5812137	-.3241359
prelast_1_v~s	.0078477	.0010028	7.83	0.000	.0058821	.0098132

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
prelast_2_m~s	-.1644774	.0180879	-9.09	0.000	-.199929	-.1290257
prelast_2_m~s	.7762924	.084016	9.24	0.000	.611624	.9409607
prelast_2_m~s	-.2941227	.7593928	-0.39	0.699	-1.782505	1.19426

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
prelast_2_l~s	-1.170893	.0670389	-17.47	0.000	-1.302287	-1.0395
prelast_2_l~s	.0527253	.0089617	5.88	0.000	.0351607	.0702898
prelast_2_l~s	.820892	.0528679	15.53	0.000	.7172729	.9245111

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
prelast_2_v~s	-1.16719	.3466737	-3.37	0.001	-1.846658	-.4877224
prelast_2_v~s	-.3766976	.0391418	-9.62	0.000	-.4534141	-.299981
prelast_2_v~s	.0896085	.0086226	10.39	0.000	.0727086	.1065085

The resulting coefficients in the table above show that, for example, a 1% increase in the amount of farm total surplus will decrease the probability to remain in the `medium` category of size two consecutive years by about 0.005% for farms in type 1 but by about 0.164% for those in type 2.

The above procedure can be applied to derive entry-probability elasticities and probability elasticities for type membership if a parametric form is used for the mixing distribution.

5.4 Structure elasticities

The estimated parameters can be also used to derive "structure elasticities". In the context of our agricultural examples, structure elasticities measure the impacts of the exogenous variable on the distribution of farms across size categories. In other words, farm structure elasticities measure the variation in percentage of the total number of farms in a specific category k for a 1% change in the ith explanatory variable. In our mixture model, average structure elasticities from one period to the next can be derived as follows.

The total number of farms in a specific category k at any specific time can be obtained as

$$n_k = \sum_{j=1}^{G} \pi_g \sum_{j}^{K} n_j \times p_{jk|g}, \qquad \forall k \in K \tag{7}$$

where π_g is the probability of belonging to type g, n_j is the total number of farms in category j at the preceding time period, and $p_{jk|g}$ is the probability of making a transition from category j to category k from one year to the next. The vector of farm structure elasticities is then defined as

$$\boldsymbol{\eta}_k = \frac{\partial n_k}{\partial \mathbf{x}} \times \frac{\overline{\mathbf{x}}}{n_k}$$

where $\overline{\mathbf{x}}$ is the vector of means of the explanatory variables over the period $T-1$.

In (7), only transition probabilities depend on exogenous variables. Farm structure elasticities can therefore be obtained using the corresponding transition-probability marginal effects. Average structure elasticities from one period to the next are thus given by

$$\boldsymbol{\eta}_k = \left(\sum_{g=1}^{G} \pi_g \sum_{j=1}^{K} n_j \times \frac{\partial p_{jk|g}}{\partial \mathbf{x}} \right) \frac{\overline{\mathbf{x}}}{n_k}$$

where $(\partial p_{jk|g})/(\partial \mathbf{x})$ are the marginal effects at the means of the corresponding explanatory variables.[11]

Considering example 1, we can derive structure elasticities at the means of explanatory variables by following three steps: i) computing farm numbers across size categories from the sample; ii) computing predictive margins at the means of the explanatory variables using the estimates; iii) deriving structure elasticities using the distribution of farms across size categories and predictive margins.

```
. /* Compute farm numbers across size categories */
. foreach ist in medium large vlarge {
  2. quietly summarize idnum if year < 2010
  3. scalar n_t_1_`ist´ = r(mean)
  4. quietly summarize idnum if year > 2000
  5. scalar n_t_`ist´ = r(mean)
  6. }
```

11. Note that yearly structure elasticities can also be derived using yearly marginal effects (see Saint-Cyr [2017] for more details).

```
. /* Formula of predictive margins at the means of the explanatory variables */
. forvalues s=1/2 {
  2. foreach x in surplus istock crop corp {
  3.   foreach fst in medium large vlarge {
  4.     foreach ist in medium large vlarge {
  5.       local m_`s´_`fst´_`x´ "`m_`s´_`fst´_`x´´+n_t_1_`ist´*
> (prmean_`s´_`ist´_`fst´*(_b[p_`s´_`ist´_`fst´:L.`x´]-`sum_`s´_`ist´_`x´´))"
  6.       }
  7.       local m_`s´_`fst´_`x´ `=substr("`m_`s´_`fst´_`x´´",2,.)´
  8.     }
  9.   }
 10. }
. /* Compute structure elasticities using predictive margins */
. fill_bV // call the subroutine to put the mixmcm estimates in b and V
. nlcom (stelast_medium_surplus:(SHARE[1,1] * `m_1_medium_surplus´ + SHARE[1,2]
> * `m_2_medium_surplus´) * (mean_surplus/n_t_medium))
> (stelast_large_surplus:(SHARE[1,1] * `m_1_large_surplus´ + SHARE[1,2]
> * `m_2_large_surplus´) * (mean_surplus/n_t_large))
> (stelast_vlarge_surplus:(SHARE[1,1] * `m_1_vlarge_surplus´ + SHARE[1,2]
> * `m_2_vlarge_surplus´) * (mean_surplus/n_t_vlarge)), noheader
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
stelast_med~s	-.170922	.0080758	-21.16	0.000	-.1867503	-.1550937
stelast_lar~s	.0161174	.0108267	1.49	0.137	-.0051025	.0373374
stelast_vla~s	.1548046	.008474	18.27	0.000	.1381958	.1714133

The coefficients in the table above show that the number of farms in the `medium` size category will decrease by 0.171% if farm total surplus increases by 1% from one year to the next. Because in example 1 we do not consider exits or entries, these 0.171% of medium farms will move to the `large` category for 0.016% and to the `vlarge` category for 0.155%.

Structure elasticities can also be derived, accounting for entries and exits as in example 2. To do this, we can use the same procedure considering the corresponding predictive margins.

6 Discussion

The community-contributed `mixmcm` command proposed here enables the estimation of MMCMs in Stata. The command is especially adapted to fit mixtures of nonstationary MCMs, accounting for entries and exits in the population under study. The examples presented in the preceding sections can easily be reproduced using other data provided that they are organized or rearranged in a similar fashion.

Future developments of the command will take three directions. First, the specification of constraints will be made more general to allow for constraining transition probabilities for only a subset of components. This would, for instance, allow for the implementation of the so-called mover–stayer model, a restricted form of the MMCM that has been widely applied in the literature (Saint-Cyr 2017). In this model, only two types of agents are considered, the "movers", who follow a standard Markovian process, and the "stayers", who always remain in their starting category. In this setting, the stayers' transition matrix degenerates to a diagonal matrix, where all but the diagonal transition probabilities are constrained to zero while the diagonal probabilities are constrained to unity, and the movers' transition matrix may be unconstrained. Generalizing the constraints syntax in `mixmcm` would thus allow for the implementation of such a model, or even a more sophisticated model composed of one stayer type and several mover types. Second, the command will also be improved to be able to account for other parametric forms (logit, Poisson, etc.) of the mixing distribution, for which there is already some precedent in the literature (Lindsay and Lesperance 1995; Greene and Hensher 2003; Train 2009). Finally, specific postestimation commands will be incorporated so that deriving second-order parameters of interest such as those exemplified in section 5 will be directly and easily available with no further coding as stand-alone in-line commands.

7 References

Albert, P. S. 1991. A two-state Markov mixture model for a time series of epileptic seizure counts. *Biometrics* 47: 1371–1381.

Andrews, R. L., and I. S. Currim. 2003. Retention of latent segments in regression-based marketing models. *International Journal of Research in Marketing* 20: 315–321.

Baudry, J.-P., and G. Celeux. 2015. EM for mixtures. *Statistics and Computing* 25: 713–726.

Blumen, I., M. Kogan, and P. McCarthy. 1955. *The Industrial Mobility of Labor as a Probability Process*. Ithaca, NY: Cornell University Press.

Buis, M. L. 2008. fmlogit: Stata module fitting a fractional multinomial logit model by quasi maximum likelihood. Statistical Software Components S456976, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s456976.html.

Chen, H. H., S. W. Duffy, and L. Tabar. 1997. A mover-stayer mixture of Markov chain models for the assessment of dedifferentiation and tumour progression in breast cancer. *Journal of Applied Statistics* 24: 265–278.

Cipollini, F., C. Ferretti, and P. Ganugi. 2012. Firm size dynamics in an industrial district: The mover-stayer model in action. In *Advanced Statistical Methods for the Analysis of Large Data-Sets*, ed. A. Di Ciaccio, M. Coli, and J. M. Angulo Ibañez, 443–452. Berlin: Springer.

Compiani, G., and Y. Kitamura. 2016. Using mixtures in econometric models: A brief review and some new results. *Econometrics Journal* 19: C95–C127.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* 39: 1–38.

Dias, J. G., and J. K. Vermunt. 2007. Latent class modeling of website users' search patterns: Implications for online market segmentation. *Journal of Retailing and Consumer Services* 14: 359–368.

Dias, J. G., and F. Willekens. 2005. Model-based clustering of sequential data with an application to contraceptive use dynamics. *Mathematical Population Studies* 12: 135–157.

Dutta, J., J. A. Sefton, and M. R. Weale. 2001. Income distribution and income dynamics in the United Kingdom. *Journal of Applied Econometrics* 16: 599–617.

Fougère, D., and T. Kamionka. 2003. Bayesian inference for the mover–stayer model in continuous time with an application to labour market transition data. *Journal of Applied Econometrics* 18: 697–723.

Frydman, H., and A. Kadam. 2004. Estimation in the continuous time mover–stayer model with an application to bond ratings migration. *Applied Stochastic Models in Business and Industry* 20: 155–170.

Frydman, H., and A. Matuszyk. 2018. Estimation and status prediction in a discrete mover–stayer model with covariate effects on stayer's probability. *Applied Stochastic Models in Business and Industry* 34: 196–205.

Frydman, H., and T. Schuermann. 2008. Credit rating dynamics and Markov mixture models. *Journal of Banking and Finance* 32: 1062–1075.

Greene, W. H. 2018. *Econometric Analysis*. 8th ed. New York: Pearson.

Greene, W. H., and D. A. Hensher. 2003. A latent class model for discrete choice analysis: Contrasts with mixed logit. *Transportation Research, Part B* 37: 681–698.

Hess, S., M. Bierlaire, and J. W. Polak. 2007. A systematic comparison of continuous and discrete mixture models. *European Transport* 37: 35–61.

Jackson, B. B. 1975. Markov mixture models for drought lengths. *Water Resources Research* 11: 64–74.

Lindsay, B. G., and M. L. Lesperance. 1995. A review of semiparametric mixture models. *Journal of Statistical Planning and Inference* 47: 29–39.

McLachlan, G., and D. Peel. 2000. *Finite Mixture Models*. New York: Wiley.

McLachlan, G. J., and T. Krishnan. 2008. *The EM Algorithm and Extensions*. 2nd ed. Hoboken, NJ: Wiley.

Pacifico, D., and H. I. Yoo. 2013. lclogit: A Stata command for fitting latent-class conditional logit models via the expectation-maximization algorithm. *Stata Journal* 13: 625–639.

Papke, L. E., and J. M. Wooldridge. 1996. Econometric methods for fractional response variables with an application to 401(K) plan participation rates. *Journal of Applied Econometrics* 11: 619–632.

Rabe-Hesketh, S. 1999. gllamm: Stata program to fit generalised linear latent and mixed models. Statistical Software Components S401701, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s401701.html.

Saint-Cyr, L. D. F. 2017. Accounting for unobserved farm heterogeneity in modelling structural change: Evidence from France. PhD thesis, UMR SMART–LERECO, Agrocampus-Ouest, INRA.

Saint-Cyr, L. D. F., and L. Piet. 2017. Movers and stayers in the farming sector: Accounting for unobserved heterogeneity in structural change. *Journal of the Royal Statistical Society, Series C* 66: 777–795.

Shorrocks, A. F. 1976. Income mobility and the Markov assumption. *Economic Journal* 86: 566–578.

Singer, B., and S. Spilerman. 1974. Social mobility models for heterogeneous populations. *Sociological Methodology* 5: 356–401.

Train, K. E. 2008. EM algorithms for nonparametric estimation of mixing distributions. *Journal of Choice Modelling* 1: 40–69.

———. 2009. *Discrete Choice Methods with Simulation*. 2nd ed. Cambridge: Cambridge University Press.

Vermunt, J. K. 2010. Longitudinal research using mixture models. In *Longitudinal Research with Latent Variables*, ed. K. van Montfort, J. Oud, and A. Satorra, 119–152. Berlin: Springer.

Zepeda, L. 1995. Technical change and the structure of production: A non-stationary Markov analysis. *European Review of Agricultural Economics* 22: 41–60.

About the authors

Legrand D. F. Saint-Cyr and Laurent Piet are researchers at INRA. Their main research interest is in modeling structural change in the farming sector. The work presented here was developed when Saint-Cyr was a doctoral candidate at the Structures et Marchés Agricoles, Ressources et Territoires–Laboratoire d'Étude et de Recherche en Économie (SMART–LERECO) research unit in Rennes, France, where he benefited from a research grant from Crédit Agricole en Bretagne for the Enterprises and Agricultural Economics project in partnership with Agrocampus-Ouest. He is now a postdoctoral fellow at the Centre d'Économie et Sociologie appliquées à l'Agriculture et aux Espaces Ruraux (UMR CESAER) research unit in Dijon, France. Laurent Piet is a research analyst at the SMART–LERECO research unit in Rennes, France.

A Appendix

A.1 Results from example 3

```
. mixmcm category surplus istock crop corp, id(idnum) timevar(year)
> ncomponents(1 5, graph(aic bic caic aic3, title("Fig. 1")
> ytitle("Information criteria") xtitle("Number of components") xlabel(1(1)5)
> scheme(sj) saving(figure.eps, replace)) save(icbtable, replace detail) force)
> emiterate(lr(3 200, 0.000001) sr(3 5)) exitcode(exits)
> entry(entry_class icap corp) membership(meandebtr modeeduc modeyoung)
> constraints(1 2)
(file figure.eps saved)
(Warning: unbalanced panel data)
  (output omitted)
log-likelihood = -40983.226                              Number of obs  = 84886
                                                         Number of id   = 13123

------------------------------------------------------------------------------
             |               Robust
             |      coef.   Std. Err.      z     p>|z|     [95% Conf. Interval]
------------------------------------------------------------------------------
Component 1
------------------------------------------------------------------------------
Entry probabilities
------------------------------------------------------------------------------
       large | (baseoutcome)
-------------+----------------------------------------------------------------
medium       |
        icap | -2.73e-06   5.77e-07    -4.74   0.000    -3.87e-06   -1.60e-06
        corp | -1.717319   .1016484   -16.89   0.000     -1.91655   -1.518088
       _cons |  .4525435   .0907567     4.99   0.000     .2746603    .6304268
vlarge       |
        icap |  3.37e-06   3.61e-07     9.33   0.000     2.66e-06    4.07e-06
        corp |  1.831689   .1633747    11.21   0.000     1.511475    2.151904
       _cons | -3.467929   .1687544   -20.55   0.000    -3.798688    -3.13717
------------------------------------------------------------------------------
Transition probabilities
------------------------------------------------------------------------------
       large | initial state
-------------+----------------------------------------------------------------
medium       |
     surplus |  -.000018   1.14e-06   -15.82   0.000    -.0000202   -.0000158
      istock | -4.46e-06   1.07e-06    -4.18   0.000    -6.55e-06   -2.37e-06
        crop | -.1936054    .071734    -2.70   0.007     -.334204   -.0530068
        corp | -.6566904   .0772081    -8.51   0.000    -.8080183   -.5053626
       _cons | -.2696556   .0894501    -3.01   0.003    -.4449777   -.0943335
vlarge       |
     surplus |  .0000112   8.43e-07    13.30   0.000     9.57e-06    .0000129
      istock |  1.95e-06   4.16e-07     4.69   0.000     1.14e-06    2.77e-06
        crop | -.5334796   .0823049    -6.48   0.000    -.6947972   -.3721619
        corp |  1.207092   .0974889    12.38   0.000     1.016014    1.398171
       _cons | -3.887953   .1072533   -36.25   0.000    -4.098169   -3.677736
exits        |
     surplus |  6.47e-06   9.33e-07     6.93   0.000     4.64e-06    8.30e-06
      istock | -5.69e-06   6.79e-07    -8.38   0.000    -7.02e-06   -4.36e-06
        crop | -.2926585   .0639043    -4.58   0.000     -.417911    -.167406
        corp |  -.735256   .0688008   -10.69   0.000    -.8701055   -.6004065
       _cons | -1.848153   .0814931   -22.68   0.000    -2.007879   -1.688427
------------------------------------------------------------------------------
```

medium	initial state					
large						
surplus	8.94e-06	1.53e-06	5.84	0.000	5.94e-06	.0000119
istock	3.89e-06	9.36e-07	4.16	0.000	2.05e-06	5.72e-06
crop	-.1340931	.0719786	-1.86	0.062	-.2751711	.006985
corp	.1633837	.082237	1.99	0.047	.0021991	.3245682
_cons	-1.560572	.0795478	-19.62	0.000	-1.716486	-1.404659
vlarge						
surplus	0 (omitted)					
istock	0 (omitted)					
crop	0 (omitted)					
corp	0 (omitted)					
_cons	0 (omitted)					
exits						
surplus	-.0000125	1.96e-06	-6.39	0.000	-.0000164	-8.67e-06
istock	1.45e-06	1.53e-06	0.95	0.344	-1.55e-06	4.44e-06
crop	-.0111323	.0759815	-0.15	0.884	-.1600561	.1377915
corp	-.1360791	.1024887	-1.33	0.184	-.336957	.0647988
_cons	-1.39741	.0940792	-14.85	0.000	-1.581806	-1.213015

vlarge	initial state					
large						
surplus	-6.19e-06	7.60e-07	-8.15	0.000	-7.68e-06	-4.70e-06
istock	-4.20e-06	5.48e-07	-7.67	0.000	-5.28e-06	-3.13e-06
crop	-.2655335	.093431	-2.84	0.004	-.4486584	-.0824086
corp	-.1009242	.1070697	-0.94	0.346	-.3107809	.1089324
_cons	.1730211	.1274562	1.36	0.175	-.0767931	.4228353
medium						
surplus	0 (omitted)					
istock	0 (omitted)					
crop	0 (omitted)					
corp	0 (omitted)					
_cons	0 (omitted)					
exits						
surplus	1.38e-06	4.66e-07	2.96	0.003	4.68e-07	2.29e-06
istock	3.35e-07	1.40e-07	2.39	0.017	5.97e-08	6.10e-07
crop	-.1276811	.1040235	-1.23	0.220	-.3315671	.076205
corp	-.6197546	.1012326	-6.12	0.000	-.8181704	-.4213387
_cons	-1.801007	.1128898	-15.95	0.000	-2.022271	-1.579743

Component 2

Entry probabilities

large	(baseoutcome)					
medium						
icap	0 (omitted)					
corp	0 (omitted)					
_cons	0 (omitted)					
vlarge						
icap	.0000128	7.49e-07	17.10	0.000	.0000113	.0000143
corp	8.392369	.8229348	10.20	0.000	6.779417	10.00532
_cons	-13.7123	.9212189	-14.88	0.000	-15.51789	-11.90671

```
Transition probabilities
------------------------------------------------------------------------------
       large | initial state
-------------+----------------------------------------------------------------
medium       |
     surplus | -.0000288   1.69e-06   -17.00   0.000    -.0000321   -.0000255
      istock | -.0000188   1.50e-06   -12.56   0.000    -.0000218   -.0000159
        crop | -.9911957   .1134279    -8.74   0.000    -1.213514   -.768877
        corp |  .3123283   .1018867     3.07   0.002     .1126303    .5120263
       _cons | -.7619696   .1343828    -5.67   0.000     -1.02536   -.4985792
vlarge       |
     surplus | -2.13e-07   1.69e-06    -0.13   0.899    -3.52e-06    3.09e-06
      istock |  1.93e-06   5.28e-07     3.66   0.000     8.94e-07    2.96e-06
        crop |  14.30387   .1328083   107.70   0.000     14.04356    14.56417
        corp |   1.16041   .1910299     6.07   0.000      .785991    1.534828
       _cons | -20.19702   .2057501   -98.16   0.000    -20.60029   -19.79375
exits        |
     surplus |  3.53e-06   6.47e-07     5.46   0.000     2.27e-06    4.80e-06
      istock | -8.66e-08   3.32e-07    -0.26   0.794    -7.38e-07    5.65e-07
        crop | -.3289612   .0543866    -6.05   0.000     -.435559   -.2223634
        corp | -.5046155   .0553418    -9.12   0.000    -.6130855   -.3961455
       _cons | -2.229801   .0571992   -38.98   0.000    -2.341911    -2.11769
-------------+----------------------------------------------------------------
      medium | initial state
-------------+----------------------------------------------------------------
large        |
     surplus |  .0000216   2.06e-06    10.53   0.000     .0000176    .0000257
      istock |  2.88e-06   1.03e-06     2.80   0.005     8.64e-07    4.89e-06
        crop | -.8224463   .1157572    -7.10   0.000     -1.04933   -.5955622
        corp |   2.29892     .10971    20.95   0.000     2.083889    2.513952
       _cons | -5.975013   .0996405   -59.97   0.000    -6.170309   -5.779718
vlarge       |
     surplus | 0 (omitted)
      istock | 0 (omitted)
        crop | 0 (omitted)
        corp | 0 (omitted)
       _cons | 0 (omitted)
exits        |
     surplus | -4.29e-06   1.47e-06    -2.92   0.003    -7.17e-06   -1.41e-06
      istock | -7.87e-06   1.03e-06    -7.67   0.000    -9.88e-06   -5.86e-06
        crop | -.0498999   .0479314    -1.04   0.298    -.1438454    .0440455
        corp | -.3045666   .0725609    -4.20   0.000     -.446786   -.1623473
       _cons | -1.806748   .0575469   -31.40   0.000     -1.91954   -1.693956
------------------------------------------------------------------------------
```

vlarge	initial state					
large						
surplus	-8.71e-06	1.35e-06	-6.43	0.000	-.0000114	-6.06e-06
istock	3.06e-07	3.68e-07	0.83	0.406	-4.15e-07	1.03e-06
crop	-.7007915	.2080657	-3.37	0.001	-1.1086	-.2929826
corp	-.3687532	.1979199	-1.86	0.062	-.7566763	.0191698
_cons	-2.27068	.2616954	-8.68	0.000	-2.783603	-1.757757
medium						
surplus	0 (omitted)					
istock	0 (omitted)					
crop	0 (omitted)					
corp	0 (omitted)					
_cons	0 (omitted)					
exits						
surplus	1.88e-06	4.01e-07	4.69	0.000	1.10e-06	2.67e-06
istock	2.85e-07	1.10e-07	2.59	0.010	6.92e-08	5.01e-07
crop	-.4222275	.1074133	-3.93	0.000	-.6327575	-.2116975
corp	-.8750444	.1070356	-8.18	0.000	-1.084834	-.6652546
_cons	-1.936757	.1149548	-16.85	0.000	-2.162069	-1.711446

Component 3

Entry probabilities

large	(baseoutcome)					
medium						
icap	-6.02e-06	4.84e-07	-12.43	0.000	-6.97e-06	-5.07e-06
corp	-1.133343	.0911934	-12.43	0.000	-1.312082	-.954604
_cons	-.082066	.0681124	-1.20	0.228	-.2155663	.0514343
vlarge						
icap	2.89e-06	2.68e-07	10.77	0.000	2.36e-06	3.41e-06
corp	.9864864	.0863775	11.42	0.000	.8171865	1.155786
_cons	-1.357328	.0882696	-15.38	0.000	-1.530337	-1.18432

Transition probabilities

large	initial state					
medium						
surplus	-6.16e-08	2.42e-06	-0.03	0.980	-4.81e-06	4.69e-06
istock	-.0000455	3.50e-06	-13.01	0.000	-.0000524	-.0000387
crop	1.032274	.157029	6.57	0.000	.7244969	1.34005
corp	-.7022026	.1827769	-3.84	0.000	-1.060445	-.34396
_cons	-3.892133	.2018793	-19.28	0.000	-4.287816	-3.496449
vlarge						
surplus	6.58e-06	8.88e-07	7.40	0.000	4.84e-06	8.32e-06
istock	-3.10e-07	5.18e-07	-0.60	0.549	-1.32e-06	7.05e-07
crop	.1236297	.0921907	1.34	0.180	-.0570641	.3043235
corp	-.618462	.0915013	-6.76	0.000	-.7978046	-.4391195
_cons	-3.698632	.097256	-38.03	0.000	-3.889254	-3.508011
exits						
surplus	-1.56e-06	6.70e-07	-2.33	0.020	-2.87e-06	-2.48e-07
istock	1.01e-06	3.30e-07	3.07	0.002	3.68e-07	1.66e-06
crop	-.144217	.0492098	-2.93	0.003	-.2406682	-.0477659
corp	-.4534594	.051372	-8.83	0.000	-.5541486	-.3527703
_cons	-2.023236	.0588712	-34.37	0.000	-2.138624	-1.907849

medium	initial state					
large						
surplus	-2.82e-08	2.50e-06	-0.01	0.991	-4.93e-06	4.87e-06
istock	1.15e-06	1.04e-06	1.11	0.269	-8.86e-07	3.18e-06
crop	.9217199	.1263195	7.30	0.000	.6741337	1.169306
corp	.1662634	.1541611	1.08	0.281	-.1358924	.4684192
_cons	-4.788261	.1087963	-44.01	0.000	-5.001502	-4.575021
vlarge						
surplus	0 (omitted)					
istock	0 (omitted)					
crop	0 (omitted)					
corp	0 (omitted)					
_cons	0 (omitted)					
exits						
surplus	8.57e-07	1.35e-06	0.63	0.526	-1.79e-06	3.51e-06
istock	-1.16e-06	1.06e-06	-1.10	0.272	-3.23e-06	9.09e-07
crop	-.058948	.0545807	-1.08	0.280	-.1659262	.0480302
corp	-.5020593	.0850698	-5.90	0.000	-.6687961	-.3353224
_cons	-1.815007	.0553823	-32.77	0.000	-1.923557	-1.706458

vlarge	initial state					
large						
surplus	-5.33e-06	8.71e-07	-6.12	0.000	-7.04e-06	-3.63e-06
istock	-8.80e-06	8.24e-07	-10.67	0.000	-.0000104	-7.18e-06
crop	-.0435028	.1148826	-0.38	0.705	-.2686727	.181667
corp	-1.19845	.1154647	-10.38	0.000	-1.424761	-.9721393
_cons	-1.507748	.1288795	-11.70	0.000	-1.760352	-1.255144
medium						
surplus	0 (omitted)					
istock	0 (omitted)					
crop	0 (omitted)					
corp	0 (omitted)					
_cons	0 (omitted)					
exits						
surplus	5.48e-08	4.22e-07	0.13	0.897	-7.73e-07	8.82e-07
istock	2.17e-07	1.20e-07	1.81	0.070	-1.78e-08	4.52e-07
crop	-.2726227	.0756379	-3.60	0.000	-.420873	-.1243724
corp	-.6692721	.0794823	-8.42	0.000	-.8250573	-.5134869
_cons	-1.982747	.082816	-23.94	0.000	-2.145066	-1.820428

Type shares

	Mean	Std. Dev.
pi1	.1843579	.2460405
pi2	.4187667	.3302194
pi3	.3968754	.3012153

Membership probabilities

	coef.	Robust Std. Err.	z	p>|z|	[95% Conf.	Interval]
proba1	(baseoutcome)					
proba2						
meandebtr	-.0534433	.0003089	-173.02	0.000	-.0540487	-.0528379
modeeduc	-.306331	.0282084	-10.86	0.000	-.3616194	-.2510426
modeyoung	.0015993	.0148086	0.11	0.914	-.0274255	.0306241
_cons	2.965792	.0301302	98.43	0.000	2.906736	3.024847
proba3						
meandebtr	.0109601	.0002757	39.75	0.000	.0104197	.0115006
modeeduc	-.1730981	.0276553	-6.26	0.000	-.2273024	-.1188937
modeyoung	-.2885172	.0141397	-20.40	0.000	-.316231	-.2608034
_cons	.4402923	.0295252	14.91	0.000	.3824228	.4981618

Model estimated via expectation-maximization (EM) algorithm.

CAIC values are used to identify the optimal number of components.

A.2 Parameters saved in table of results

The two tables below report only the first 20 lines of `icbtable.dta`. `nco~s`, `npa~s`, `com~t`, `ist`, and `fst`, respectively, stand for the number of components, the number of parameters, the component number, the initial state, and the final state. The table reports the estimated parameters only for one explanatory variable, namely, `surplus`. To save space, we truncate the numbers of the variables `var_pi` and `pi_` to three decimal places.

```
. use "icbtable.dta", clear
. list ncomponents LL nparameters aic bic caic aic3 in 1/20, abbreviate(5)
> compress noobs separator(20)
```

nco~s	LL	npa~s	aic	bic	caic	aic3
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
1	-43066.907	41	86215.814	86599.126	86640.126	86256.814
2	-41410.741	86	82993.482	83797.501	83883.501	83079.482
2	-41410.741	86	82993.482	83797.501	83883.501	83079.482
2	-41410.741	86	82993.482	83797.501	83883.501	83079.482
2	-41410.741	86	82993.482	83797.501	83883.501	83079.482
2	-41410.741	86	82993.482	83797.501	83883.501	83079.482

```
. format var_pi pi_ %12.3f
. list ncomponents component ist fst b_surplus se_surplus pi_ var_pi_ in 1/20,
> abbreviate(5) compress noobs separator(20)
```

nco~s	com~t	ist	fst	b_surplus	se_surp~s	pi_	var~_
1	1	entry	large	.	.	1.000	.
1	1	entry	medium	.	.	1.000	.
1	1	entry	vlarge	.	.	1.000	.
1	1	large	exits	1.817e-06	5.627e-07	1.000	.
1	1	large	large	0	0	1.000	.
1	1	large	medium	-.00001792	1.043e-06	1.000	.
1	1	large	vlarge	7.392e-06	6.672e-07	1.000	.
1	1	medium	exits	-3.911e-06	1.168e-06	1.000	.
1	1	medium	large	.00001078	1.357e-06	1.000	.
1	1	medium	medium	0	0	1.000	.
1	1	medium	vlarge	0	0	1.000	.
1	1	vlarge	exits	8.403e-07	3.319e-07	1.000	.
1	1	vlarge	large	-5.599e-06	5.627e-07	1.000	.
1	1	vlarge	medium	0	0	1.000	.
1	1	vlarge	vlarge	0	0	1.000	.
2	1	entry	large	.	.	0.660	0.312
2	1	entry	medium	.	.	0.660	0.312
2	1	entry	vlarge	.	.	0.660	0.312
2	1	large	exits	2.548e-06	5.810e-07	0.660	0.312
2	1	large	large	0	0	0.660	0.312

A.3 Compute transition probabilities

```
. /* Generate the components of the multinomial logit formula */
. sort idnum year
. quietly xtset idnum year // declare panel data
. forvalues s=1/2 {
2.   foreach ist in medium large vlarge {
3.      by idnum: generate double deno_betax_`s´_`ist´= 0 if
> category[_n-1]=="`ist´"
4.      foreach fst in medium large vlarge {
5.         if "`ist´"=="`fst´" {
6.            by idnum: generate double sum_betax_`s´_`ist´_`fst´=0 if
> category[_n-1]=="`ist´"
7.         }
8.         else {
9.            by idnum: generate double sum_betax_`s´_`ist´_`fst´=
> BETA[rownumb(BETA,"r1"), colnumb(BETA,"p_`s´_`ist´_`fst´:_cons")] if
> category[_n-1]=="`ist´"
10.           foreach x in surplus istock crop corp {
11.              local sum_betax_`s´_`ist´_`fst´ "`sum_betax_`s´_`ist´_`fst´´ +
> (p_`s´_`ist´_`fst´:L.`x´)*L.`x´"
12.              by idnum: replace sum_betax_`s´_`ist´_`fst´=
> sum_betax_`s´_`ist´_`fst´ + (BETA[rownumb(BETA,"r1"),
> colnumb(BETA,"p_`s´_`ist´_`fst´:L.`x´")])*L.`x´ if category[_n-1]=="`ist´"
13.           }
14.           by idnum: replace deno_betax_`s´_`ist´= deno_betax_`s´_`ist´ +
> exp(sum_betax_`s´_`ist´_`fst´) if category[_n-1]=="`ist´"
15.        }
16.     }
17.  }
18.}
```

```
. /* Use the multinomial logit formula to compute transition probabilities */
. forvalues s=1/2 {
2.   foreach ist in medium large vlarge {
3.      foreach fst in medium large vlarge {
4.         capture drop pr_`s´_`ist´_`fst´
5.         quietly generate double pr_`s´_`ist´_`fst´ =.
6.         by idnum: replace pr_`s´_`ist´_`fst´ = exp(sum_betax_`s´_`ist´_`fst´)/
> (1+deno_betax_`s´_`ist´) if category[_n-1]=="`ist´"
7.      }
8.   }
9.}
```

A.4 Compute type membership probabilities

```
. /* Generate the components of the Bayes formula */
. capture drop _obs
. by idnum: generate _obs=1 if _n==1
. capture drop _dens_tot
. generate double _dens_tot = 0
. forvalues s=1/2 {
2.   capture drop _pr_`s´
3.   generate double _pr_`s´ = 1
4.   foreach ist in medium large vlarge {
5.      foreach fst in medium large vlarge {
6.         by idnum: replace _pr_`s´ = pr_`s´_`ist´_`fst´ if
> category[_n]=="`fst´" & category[_n-1]=="`ist´"
7.      }
8.   }
9.   capture drop _dens1_`s´
10.  capture drop sumprod1_`s´
11.  by idnum: gen double sumprod1_`s´=exp(sum(ln(_pr_`s´)))
12.  by idnum: gen double _dens1_`s´ = sumprod1_`s´[_N]
13.  capture drop pi_dens_`s´
14.  quietly generate double pi_dens_`s´ = round(SHARE[rownumb(SHARE,"Mean"),
> colnumb(SHARE,"pi_`s´")],0.01)*_dens1_`s´
15.  quietly replace _dens_tot = _dens_tot+pi_dens_`s´
16.}
. /* Use the Bayes formula to compute posterior membership probabilities */
. forvalues s=1/2 {
2.   capture drop proba_`s´
3.   quietly generate double proba_`s´ = pi_dens_`s´/_dens_tot
4.   summarize proba_`s´ if _obs==1
5.}
```

Variable	Obs	Mean	Std. Dev.	Min	Max
proba_1	13,123	.7233329	.2715939	2.39e-19	1

Variable	Obs	Mean	Std. Dev.	Min	Max
proba_2	13,123	.2766671	.2715939	4.04e-16	1