Examples and tips for estimating Markovian models with `seqHMM`

Satu Helske
University of Oxford, UK

May 9, 2018

This vignette is supplementary material to the paper Helske and Helske (2017), giving more detailed tips and suggestions on model estimation and setting starting values. The examples here come from the social sciences but are hopefully general enough to be understandable without any deeper knowledge.

Estimation of Markovian models typically needs starting values for model parameters, i.e., initial, transition, and/or emission probabilities. If the models are small and simple, any random starting values will usually do. With large and complex models, good starting values are needed for finding the optimal solution in a reasonable time. Starting values may also be used for setting restrictions to the structure of the model (e.g., for determining a left-to-right model, see Section 3). In order to reduce the risk of being trapped in a poor local maximum, a large number of initial values should be tested.

The model building functions `build_hmm` (for hidden Markov models), `build_mhmm` (mixture hidden Markov models), `build_mmm` (mixture Markov models), and `build_lcm` (latent class models) do not require starting values given by the user; if those are not provided, models are initialized with random values. In this case the user must provide the number of hidden states or clusters/submodels. For ordinary Markov models, the `build_mm` function automatically estimates the initial probabilities and the transition matrix based on the observations. The helper functions `simulate_initial_probs`, `simulate_transition_probs`, and `simulate_emission_probs` are also available for creating random starting values for corresponding model parameters.

Here we mostly focus on the case of hidden Markov models (HMMs) but everything should be relatively easily applied to other models as well.

1 Model parameters

Initial probabilities

Initial probabilities give the probability of starting in a given hidden state in HMM or observed state in Markov model. In a typical life course problem, most individuals start from a similar hidden state before any interesting transitions

have happened. This hidden state should be given the highest starting value and states that typically occur later in life should have small probabilities. In a time use problem, on the other hand, initial probabilities should reflect how probable different activities are at the start of the follow-up: at 4am, for instance, most individuals are typically still sleeping, some are working, and a minority doing other activities. In an HMM, starting values for initial probabilities of hidden states should thus be higher for hidden states that are related to sleeping, lower for hidden states related to work, and very small for hidden states related to other activities. If the goal is to distinguish different types of time use patterns with mixture models, then in one submodel there might be high initial probability for sleeping and in another submodel for working (and small initial probabilities for other activities).

Transition probabilities

Typically, the highest transition probabilities are found on the diagonal, indicating not leaving the state. With life course data, transitions between different (hidden) states are often relatively rare, meaning that individuals tend to stay in one state for quite a while. In a time use study, on the other hand, the duration of some activities might typically be quite short, indicating that the probability of staying in the state should be set relatively low.

The probabilities of making a transition to another state depend on the data and the time scale, e.g., transitions are naturally more frequent if the same data are coded annually rather than monthly. A transition probability of 0.9 on the diagonal means that nine times out of ten, the hidden state remains the same. In annually coded data that means a transition to a new state once every ten years on average, whereas in monthly coded data the same transition probability leads to more than one transition within a year, on average.

1.1 Emission probabilities

Emission probabilities in HMM, MHMM, and LCM tell how likely each observed state is given being in a certain hidden state and/or subpopulation. The question of setting starting values for emission probabilities comes down to what the hidden states represent or what those are used for. Sometimes hidden states are close to observed states with some "error", sometimes they "generate" multiple observed states with varying probabilities. The next section gives more detailed suggestion regarding some special cases.

2 Tips for special cases

Measurement error

One situation for using HMMs is to account for measurement error (see, e.g., Breen and Moisio, 2004; Pavlopoulos and Vermunt, 2015; Vermunt, Tran, and Magidson, 2008, for some examples). As an example, let us consider sequences

of two observed states, employed and out of employment, where we know that observations are recorded inaccurately. In this case, hidden states represent "true" observations and they are most likely quite close to observed states, i.e., there are as many hidden states as there are observed states and the highest emission probabilities are on the diagonal. If we assume that the chances of having false measurements are 5% for both observations, starting values for the emission matrix might look like this:

Hidden states	Observed states	
("truth")	Employed	Out of empl.
Employed	0.95	0.05
Out of empl.	0.05	0.95

Similarly, also transition probabilities between hidden states are likely to be close to transition probabilities between observed states. Note that in the case of observed zero transition probabilities, for the starting values one should make a distinction between impossible transitions and transitions that are possible but not observed in the sample. The former should be set to zero and the latter given a small probability.

Episodes of stability and change

Another situation might be to look for episodes of stability and change. With the employment/out of employment data the hidden states could represent a relatively stable stage of employment, another relatively stable stage of being out of employment, and a third stage characterized by frequent changes between the two states (e.g., between unemployment and short-term jobs). In this case your emission matrix would have one hidden state with a high emission probability for employment, another hidden state with a high probability for being out of employment, and a third hidden state with emission probabilities closer to 0.5 for both observed states. Good starting values for the emission probabilities might thus be something like this:

	Observed states	
Hidden states	Employed	Out of empl.
Stable employment	0.97	0.03
Stable out of empl.	0.03	0.97
Unstable employment	0.50	0.50

For setting starting values for the transition matrix, one could look at the data and see how frequent transitions in and out of employment are in general and how long the episodes tend to be (statistics about the distribution, such as percentiles, may be of use here).

Compressing multichannel data

Hidden Markov models may also be used for compressing multichannel observed data, e.g., for finding more general life stages based on information from multiple life domains (see Helske, Helske, and Eerola, 2016, for an example). In this case, at least some of the hidden states may be quite close to the most prevalent combinations of observed states. One option is to split the data, e.g., by time period or age, and use observed state frequencies as starting values for emission probabilities. The `seqstatf` function in the `TraMineR` package is useful for this task.

Mixture models

In the case of mixture models (MHMM, MMM, and LCM), starting values are needed for several submodels. Model-free and data-driven sequence analysis (see the `TraMineR package`) is often a good and relatively fast starting point for finding subpopulations. As these subpopulations are relatively homogeneous in terms of observed states and the timing and/or length of different episodes, determining starting values should become easier.

3 Restricted models

In model estimation in `seqHMM`, zeros in starting values are regarded as structural zeros and not changed during estimation. This makes it possible to determine restricted model structures. These may be favourable due to easier interpretation, but simpler models with more structural zeros also reduce estimation time. It is thus beneficial to restrict models when there is prior knowledge on some parts of the model. For example, if all subjects are in the same situation at the start of the follow-up, there is no need for estimating initial probabilities (i.e., the first hidden state is given the probability of 1 and others set to 0). Similarly, if some transitions are impossible by theory, these should be set to zero before estimation.

Left-to-right model is a model where transitions to previous hidden states are impossible (on the hidden level of the HMM or on the observed level of the Markov model). It is determined with an upper triangular transition probability matrix such as the following one for the number of children.

	To			
From	No child	1 child	2 children	3+ children
No child	0.90	0.085	0.0148	0.0002
1 child	0	0.800	0.1850	0.0150
2 children	0	0	0.9000	0.1000
3+ children	0	0	0	1

A Bakis model is a restricted version of the left-to-right model where transitions are only possible to the next two states, i.e., skipping two or more states is not

allowed. In the previous example of the number of children this would mean that it is only possible to have two children at a time which, given the rarity of multiple births, might be the case in a sample.

	To			
From	No child	1 child	2 children	3+ children
No child	0.90	0.085	0.015	0
1 child	0	0.800	0.185	0.015
2 children	0	0	0.900	0.100
3+ children	0	0	0	1

In this way it is, of course, possible to restrict the model in more unique ways. For example, consider having eight types of partnership statuses: never-partnered single, living apart together (LAT; having a committed relationship before or without cohabitation or marriage), cohabiting, married, separated from non-marital partner, separated from marital spouse, divorced, and widowed. In a Western society, a transition probability matrix for these partnership statuses might look something like the following.

	To							
From	Single	LAT	Cohab.	Marr.	Sep. (p)	Sep. (s)	Div.	Wid.
Single	0.70	0.28	0	0	0	0	0	0
LAT	0	0.60	0.12	0.12	0.16	0	0	0
Cohabiting	0	0	0.70	0.15	0.15	0	0	0
Married	0	0	0	0.80	0	0.08	0.08	0.04
Sep. (partner)	0	0.40	0	0	0.60	0	0	0
Sep. (spouse)	0	0	0	0	0	0.40	0.59	0.01
Divorced	0	0.30	0	0	0	0	0.70	0
Widowed	0	0.30	0	0	0	0	0	0.70

The structure of the transition matrix suggests that from the first state, never-partnered single, the entry into any partnership is through LAT relationship. From a LAT relationship it is possible to move in with the partner, get married, or separate. Cohabiting partners may get married or separate and marital spouses may separate or get divorced or widowed. Individuals separated from marriage are assumed to get divorced or widowed before forming new partnerships, and after non-marital separation, divorce, or widowhood, the first step is a LAT relationship.

4 Tips for model estimation

In order to reduce the risk of being trapped in a poor local optimum, a large number of initial values should be tested. For finding the best solution, it

is advisable to try a few different approaches to parameter estimation such as direct numerical maximization (DNM) and expectation-maximization (EM) algorithm and a combination of these. The help file of the `fit_model` function gives examples on several estimation options.

One option is to run the EM algorithm multiple times with more or less randomized starting values for transition or emission probabilities or both. These are called with the `control_em` argument. The user can set the number of restarts and choose if they want randomization to transition or emission probabilities or both. Randomization is done with the `rnorm` function, for which the user can set the standard deviation. Smaller standard deviation keeps the randomized values closer to the starting values. The function returns the log-likelihoods of the best models (at most 25 by default, but this may be changed with the `n_optimum` argument). It may be a good idea to first use a larger standard deviation for randomization and then re-estimate with smaller standard deviation, using the estimated parameters of the best model from the previous step as a starting point.

A global DNM method, the multilevel single-linkage method (MLSL), is another option for estimating a number of models with a single call. There are some theoretical guarantees that the MLSL method finds all local optima in a finite number of local optimizations. Of course, it might not always succeed in a reasonable time. Also, it requires setting boundaries for the parameter space, which is not always straightforward. In DNM steps the transition, emission, and initial probabilities are estimated using unconstrained reparameterization using the softmax function (a generalization of the logistic function), but good boundaries are essential for the efficient use of the MLSL algorithm. If the boundaries are too strict, the global optimum cannot be found; if too wide, the probability of finding the global optimum is decreased. The `fit_model` function uses starting values or results from the preceding estimation step to adjust the boundaries. EM can help in setting good boundaries, but in some cases it can also lead to worse results.

In practice we have had most success with randomized EM, but it is advisable to try a few of different settings, e.g., randomized EM, EM followed by global DNM, and only global DNM with different optimization routines.

Note also that parallel computation may reduce estimation time considerably. The number of parallel threads (typically, the number of cores) is chosen with the `threads` argument. The `system.time` function may be of help in determining the optimal number of threads.

References

Richard Breen and Pasi Moisio. Poverty dynamics corrected for measurement error. *The Journal of Economic Inequality*, 2(3):171–191, 2004. doi: 10.1007/s10888-004-3227-9.

Satu Helske and Jouni Helske. Mixture hidden Markov models for sequence

data: the seqHMM package in R. Preprint ArXiv:1704.00543, 2017. URL `http://arxiv.org/abs/1704.00543`.

Satu Helske, Jouni Helske, and Mervi Eerola. Analysing complex life sequence data with hidden markov modelling. In Gilbert Ritschard and Matthias Studer, editors, *Proceedings of the International Conference on Sequence Analysis and Related Methods, Lausanne, June 8-10, 2016*, pages 209–240. 2016.

Dimitris Pavlopoulos and Jeroen K Vermunt. Measuring temporary employment: Do survey or register data tell the truth? *Statistics Canada, Catalogue No. 12-001-X*, 41(1):197–214, 2015.

Jeroen K Vermunt, Bac Tran, and Jay Magidson. *Latent Class Models in Longitudinal Research*, pages 373–385. Handbook of Longitudinal Research: Design, Measurement, and Analysis. Elsevier, Burlington, MA, 2008.