Dipartimento di Economia Politica

Materiali di discussione

\\ 663 \\

Estimating nonparametric mixed Logit Models via EM algorithm

Daniele Pacifico

September 2011

Italian Department of the Treasury
e-mail: daniele.paci_co@tesoro.it

ISSN: 2039-1439 a stampa
ISSN: 2039-1447 on line

Viale Jacopo Berengario 51 – 41100 MODENA (Italy) tel. 39-059.2056711Centralino) 39-059.2056942/3 fax. 39-059.2056947

The Stata Journal (*yyyy*) *vv*, Number *ii*, pp. 1–17

Estimating nonparametric mixed logit models via EM algorithm

Daniele Pacifico
Italian Department of the Treasury
daniele.pacifico@tesoro.it

Abstract. The aim of this paper is to describe a Stata routine for the nonparametric estimation of mixed logit models with an Expectation-Maximisation algorithm proposed in Train (2008). We also show how to use the Stata command `lclogit`, which performs the estimation automatically.

Keywords: st0001, `lclogit`, latent classes, EM algorithm, mixed logit

1 Introduction

In a recent contribution Train (2008) has showed how EM algorithms can be used for the nonparametric estimation of mixing distributions in discrete choice models. In this paper we consider one of the three nonparametric methods he proposes and show how it can be implemented in Stata. The method presented here allows estimating discrete mixing distributions with mass points and share probabilities as parameters. Therefore, the nonparametric estimation is based on a typical latent class model and is reached by increasing the number of mass points of each coefficient so as to approximate their mixing distributions.

Traditionally, latent class models have been estimated using gradient-based algorithms, such as Newton-Raphson or BHHH. However, the estimation based on standard optimization procedures becomes difficult when the number of mass points increases, as the higher the number of latent classes the more difficult the empirical inversion of the Hessian matrix, with the possibility of singularity at some iterations. In such situation an EM algorithm could help as it implies the repeated evaluation of a function that is far easier to maximize. Moreover, EM recursions have proved to be particularly stable and - under conditions given by Boyles (1983) and Wu (1983) - they always climb uphill until convergence to a local maximum.

The paper is structured as follows: section 2 presents a mixed logit model based on discrete mixed distributions; section 3 shows how this model can be estimated via EM algorithm; section 4 presents a detailed step-by-step description of EM estimation in Stata; section 5 introduces the Stata command `lclogit`; section 6 contains an empirical application based on accessible data; section 7 concludes.

 st0001

2 A mixed logit model with discrete mixing distributions

Assume there are N agents who face J alternatives on T choice occasions. Agents choose the alternative that maximizes their utility in each choice occasion. The random utility of agent i from choosing alternative j in period t is defined as follows:

$$U_{ijt=}\boldsymbol{\beta}_i \boldsymbol{x}_{ijt} + \epsilon_{ijt} \tag{1}$$

Where $\boldsymbol{x}_{ijt}$ is a vector of alternative-specific attributes and ϵ_{ijt} is a stochastic term, which is assumed to be distributed IID extreme value. Importantly, each β_i is assumed to be random with unconditional density $f(\beta \mid \boldsymbol{\vartheta})$, where $\boldsymbol{\vartheta}$ collects the parameters that characterize the distribution.

Conditional on knowing $\boldsymbol{\beta}_i$ the probability of the observed sequence of choices for agent i is given by the traditional McFadden's choice model:[1]

$$Pr_i(\boldsymbol{\beta}) = \prod_{t=1}^{T} \prod_{j=1}^{J} \left(\frac{exp(\boldsymbol{\beta}_i \boldsymbol{x}_{ijt})}{\sum_{k=1}^{J} exp(\boldsymbol{\beta}_i \boldsymbol{x}_{ikt}))} \right)^{d_{ijt}} \tag{2}$$

Where d_{ijt} is a dummy that selects the chosen alternative in each choice occasion. However, since $\boldsymbol{\beta}_i$ is unknown the conditional probability of the sequence of observed choices has to be evaluated for any possible value of $\boldsymbol{\beta}_i$. Hence, assuming that $f(\beta \mid \boldsymbol{\vartheta})$ has a continuous distribution, the unconditional probability becomes:

$$Pr_i(\boldsymbol{\vartheta}) = \int \prod_{t=1}^{T} \prod_{j=1}^{J} \left(\frac{exp(\boldsymbol{\beta} \boldsymbol{x}_{ijt})}{\sum_{k=1}^{J} exp(\boldsymbol{\beta} \boldsymbol{x}_{ikt}))} \right)^{d_{ijt}} f(\boldsymbol{\beta}|\boldsymbol{\vartheta}) \tag{3}$$

Typically, the log likelihood function derived from this model is estimated with simulation methods.[2]

If the distribution of each β_i is discrete, the probability in equation 3 becomes:

$$Pr_i(\boldsymbol{\vartheta}) = \sum_{c=1}^{C} \pi_c \prod_{t=1}^{T} \prod_{j=1}^{J} \left(\frac{exp(\boldsymbol{\beta_c} \boldsymbol{x}_{ijt})}{\sum_{k=1}^{J} exp(\boldsymbol{\beta_c} \boldsymbol{x}_{ikt}))} \right)^{d_{ijt}} \tag{4}$$

Where $\pi_c = f(\boldsymbol{\beta_c}|\boldsymbol{\vartheta})$ represents the share of the population with coefficients $\boldsymbol{\beta_c}$.

Equation 4 is a typical latent class model. Nevertheless, here we follow the classification proposed by McFadden and Train (2000) and define it as a discrete mixed model, in order to emphasize the similarities with the continuous mixed model of equation 3.

The estimation of discrete mixed models is usually based on standard gradient-based methods. However, these methods often fail to achieve convergence when the number of latent classes becomes high. In this case an EM algorithm could help, as it requires the repeated maximization of a function that is far simpler with respect to the log likelihood derived from equation 4.

1. See McFadden (1973).
2. See Train (2003). In Stata, continuous mixed logit models can be estimated with Simulated Maximum Likelihood through the program `mixlogit`, written by Hole (2007).

3 An EM algorithm for the estimation of mixed logit models with discrete mixing distributions

EM algorithms were initially proposed in the literature to deal with missing data problems. Nevertheless, they turned out to be an efficient method to estimate latent class models, where the missing information consists of the class share probabilities. Nowadays, they are widely used in many economic fields where the assumption that people can be grouped in classes with different unobserved preference heterogeneity is reasonable.

The recursion is known as "E-M" because it consists of two steps, namely an "Expectation" and a "Maximization". As explained in Train (2008), the term to be maximized is the expectation of the *missing-data* log likelihood - i.e the *joint* density of the observed choice *and* the missing data - whilst the expectation is over the distribution of the missing information, conditional on the density of the data *and* the previous parameter estimates.

Consider the conditional logit model with discrete mixing distributions outlined in the previous section. Following equation 4, the log likelihood is defined as:

$$LL = \sum_{i=1}^{N} ln \sum_{c=1}^{C} \pi_c \prod_{t=1}^{T} \prod_{j=1}^{J} \left(\frac{exp(\boldsymbol{\beta_c x}_{ijt})}{\sum_{k=1}^{J} exp(\boldsymbol{\beta_c x}_{ikt}))} \right)^{d_{ijt}} \tag{5}$$

Which can be maximized by means of standard, gradient-based optimization methods. However, the same log likelihood can be also maximized by repeatedly updating the following recursion:

$$\begin{aligned} \boldsymbol{\eta}^{s+1} &= argmax_{\eta} \textstyle\sum_i \sum_c C_i(\boldsymbol{\eta}^s) ln\, \pi_c \prod_t \prod_j \left(\frac{exp(\boldsymbol{\beta_c x}_{ijt})}{\sum_{k=1}^{J} exp(\boldsymbol{\beta_c x}_{ikt}))} \right)^{d_{ijt}} \\ &= argmax_{\eta} \textstyle\sum_i \sum_c C_i(\boldsymbol{\eta}^s) ln(L_i \,|\, class_i = c) \end{aligned} \tag{6}$$

Where $\boldsymbol{\eta}$ is as vector that contains the whole set of parameters to be estimated - i.e. those that enter the probability of the observed choice plus those that may define the class shares - L_i is the *missing-data* likelihood function and $C_i(\boldsymbol{\eta}^s)$ is the *conditional* probability that household i belongs to class c, which depends on the density of the data and the previous value of the parameters.

This conditional probability - $C_i(\boldsymbol{\eta}^s)$ - is the key future of the EM recursion and can be computed by means of the Bayes' theorem:

$$C_i(\boldsymbol{\eta}^s) = \frac{L_i | class_i = c}{\sum_{c=1}^{C} L_i | class_i = c} \tag{7}$$

Now, given that:

$$ln(L_i \,|\, class_i = c) = ln\, \pi_c + ln \prod_{t=1}^{T} \prod_{j=1}^{J} \left(\frac{exp(\boldsymbol{\beta_c x}_{ijt})}{\sum_{k=1}^{J} exp(\boldsymbol{\beta_c x}_{ikt}))} \right)^{d_{ijt}} \tag{8}$$

The recursion in equation 6 can be split into the following steps:

1. Form the contribution to the likelihood $(L_i \,|\, class_i = c)$ as defined in equation 6 *for each class*;[3]

2. Form the *individual-specific* conditional probabilities of class membership as in equation 7;

3. Following equation 8, update the sets of $\boldsymbol{\beta_c},\ c = 1, 2, ..., C$ by maximizing - *for each class* - a conditional logit model with weighed observations, with weights given by the conditional probabilities of class membership:

$$\boldsymbol{\beta_c^{s+1}} = argmax_{\boldsymbol{\beta}} \sum_{i=1}^{N} C(\boldsymbol{\eta}^s) ln \prod_{t=1}^{T} \prod_{j=1}^{J} \left(\frac{exp(\boldsymbol{\beta_c x}_{ijt})}{\sum_{k=1}^{J} exp(\boldsymbol{\beta_c x}_{ikt}))} \right)^{d_{ijt}} \tag{9}$$

4. Maximize the other part of the log likelihood in equation 6 and get the updated vector of class shares:

$$\boldsymbol{\pi^{s+1}} = argmax_{\boldsymbol{\pi}} \sum_{i=1}^{N} \sum_{c=1}^{C} C_i(\boldsymbol{\eta}^s) ln\, \pi_c \tag{10}$$

   - If the class share probabilities depend on a vector of demographics - $\boldsymbol{z_i}$ - the relative parameters are updated as:

$$\boldsymbol{\alpha^{s+1}} = argmax_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \sum_{c=1}^{C} C_i(\boldsymbol{\eta}^s) ln \frac{exp(\boldsymbol{\alpha_c z_i})}{\sum_c exp(\boldsymbol{\alpha_c z_i})},\ \boldsymbol{\alpha_C = 0} \tag{11}$$

     This is a *grouped-data log likelihood*, where we have used a logit specification so as to constrain the estimated class share probabilities into the right range.[4] The updated class share probabilities - $\pi_c,\ c = 1, 2, ..., C$ - are then computed as:

$$\pi_c^{s+1} = \frac{exp(\boldsymbol{\hat{\alpha}_c^{s+1} z_i})}{\sum_c exp(\boldsymbol{\hat{\alpha}_c^{s+1} z_i})},\ c = 1, 2, ..., C \tag{12}$$

     Which, in turn, allows updating the conditional probabilities of class membership by means of the new vectors $\boldsymbol{\beta_c^{s+1}}$ and $\pi_c^{s+1},\ c = 1, 2, ..., C$.

   - If the class share probabilities do not depend on demographics the empirical maximization of the function in equation 11 can be avoided, as its analytical solution would be given by:

$$\pi_c^{s+1} = \frac{\sum_i C_i(\boldsymbol{\eta}^{s+1})}{\sum_i \sum_c C_i(\boldsymbol{\eta}^{s+1})},\ c = 1, ..., C \tag{13}$$

---

3. For the first iteration, starting values have to be used for the densities that enter the model. Note that these starting values must be different in every class. Otherwise, the recursion estimates the same set of parameters for all the classes.
4. Differently from the $\boldsymbol{\beta_c}$s, the vectors $\boldsymbol{\alpha_c},\ c = 1, 2, ..., C$ are jointly estimated. This is needed in order to ensure that $\sum_c \pi_c = 1$.

Where the updated conditional probabilities - $C_i(\boldsymbol{\eta}^{s+1})$ - are computed by using the updated values of $\boldsymbol{\beta}_c$, c=1,2,...,C and the *previous* values of the class shares.

5. Once the conditional probabilities of class membership have been updated - either in models with or without covariates on $\boldsymbol{z_i}$ - the recursion can start again from point 3 until convergence.

4 A Stata routine for the estimation of mixed logit models with discrete mixing distributions

In this section we show how the EM algorithm outlined above can be coded into Stata. We propose a detailed step-by-step procedure that can be easily implemented in a simple `do` file and work with accessible data from Huber and Train (2000) on household's choice of electricity supplier. Note that this is the same database used by Hole (2007) for an application of his Stata program `mixlogit`, which performs the estimation of *parametric* mixed logit models via Simulated Maximum Likelihood.

The data collects information on 100 residential electricity customers, who were asked up to 12 choice experiments.[5] In each experiment the customer was asked which of the 4 suppliers he/she would prefer among four hypothetical electricity suppliers.

The following characteristics of each offer were stated:

- The price of the contract (in cents per kWh) whenever the supplier offers a contract with a fixed rate (`price`)
- The length of contract that the supplier offered, expressed in years (`contract`)
- Whether the supplier is a local company (`local`)
- Whether the supplier is a well-known company (`wknown`)
- Whether the supplier offers a time-of-day rate instead of a fixed rate (`tod`)
- Whether the supplier offers a seasonal rate instead of a fixed rate (`seasonal`)

Each costumer is identified by the variable `pid`. For each costumer, the variable `gid` identifies a given choice occasion, while the dummy variable `y` identifies the stated choice in each choice occasion.

The data setup required for estimating the model is as follows:

```
. use http://fmwww.bc.edu/repec/bocode/t/traindata.dta, clear
. list in 1/12, sepby(gid)
```

5. Since some customers stopped before answering all 12 experiments, there is a total of 1195 choice occasions in the sample.

	y	price	contract	local	wknown	tod	seasonal	gid	pid
1.	0	7	5	0	1	0	0	1	1
2.	0	9	1	1	0	0	0	1	1
3.	0	0	0	0	0	0	1	1	1
4.	1	0	5	0	1	1	0	1	1
5.	0	7	0	0	1	0	0	2	1
6.	0	9	5	0	1	0	0	2	1
7.	1	0	1	1	0	1	0	2	1
8.	0	0	5	0	0	0	1	2	1
9.	0	9	5	0	0	0	0	3	1
10.	0	7	1	0	1	0	0	3	1
11.	0	0	0	0	1	1	0	3	1
12.	1	0	0	1	0	0	1	3	1

The next subsection presents the steps for estimating a model with covariates on the class share porbabilities. Alternatively, the optimization of the algorithm when only a constant term is included among the class probabilities is presented in subsection 4.2.

4.1 A model with covariates on the class share probabilities

In order to present a flexible routine we work with global variables, so that the code can be easily adapted to other databases. The dependent variable is called `$depvar`, the list of covariates that enter the probability of the observed choice `$varlist`; the list of variables that enter the grouped-data log likelihood `$varlist2`; the variable that identifies the panel dimension - i.e. the choice makers - `$id`; the variable that defines the choice situations for each choice maker `$group`. We also define the number of latent classes `$nclasses` and the number of maximum iterations `$niter`.[6]

```
. **(1) Set the estimation setup**
. global depvar "y"
. global varlist "price contract local wknown tod seasonal"
. gen _con=1
. global varlist2 "_con x2"
. global id "pid"
. global group "gid"
. global nclasses "2"
. global niter "35"
```

In order to compute the starting values, we randomly split the sample into C different sub-samples - one for each class - and estimate a separate `clogit` for each of them.[7]

6. In the following estimation setup the covariate that we included in `$varlist2` - `x2` - can be created manually, as the database does not contain individual-level variables.
7. If the same starting values were used for all the classes the EM algorithm would perform the same computations for each class and return the same results at each iteration.

After each `clogit` estimation we use the command `predict` to obtain the probability of every alternative in each class and we store them in the variables `_l1`, `_l2`,...,`_lC`.

As for the starting values for the probability of class membership we simply define equal shares, that is $\frac{1}{C}$:

```
. **(2) Split the sample**
. bysort $id: gen double _p=runiform() if _n==_N
. bysort $id: egen double _pr=sum(_p)
. local prop 1/$nclasses
. gen double _ss=1 if _pr<=`prop´
. forvalues s=2/$nclasses  {
.   replace _ss=`s´ if _pr>(`s´-1)*`prop´ & _pr<=`s´*`prop´
.                          }
. **(3) Get starting values for the beta coefficients and the class shares**
. forvalues s=1/$nclasses  {
.    gen double _prob`s´=1/$nclasses
.    clogit $depvar $varlist if _ss==`s´, group($group) technique(nr dfp)
.    predict double _l`s´
.                          }
```

In what follows, the steps to calculate the conditional probabilities of equation 7 from these starting values are presented.

First, for each latent class we multiply the variables `_l1`, `_l2`,...,`_lC` by the dummy variable that identifies the observed choice in each choice situation, i.e. `$depvar`. Note that this allows storing the probabilities of the observed choice for each class.

Second, for each latent class we multiply the probabilities of the observed choices *in each choice situation* in order to obtain the probability of the agent's sequence of choices:[8]

```
. **(4) Compute the probability of the agent´s sequence of choices**
. forvalues s=1/$nclasses  {
.   gen double _kbb`s´=_l`s´*$depvar
.   recode _kbb`s´ 0=.
.   bysort $id: egen double _kbbb`s´=prod(_kbb`s´)
.   bysort $id: replace _kbbb`s´=. if _n!=_N
.                          }
```

Third, we construct the denominator of equation 7, i.e. the unconditional choice probability for each choice maker. This is done by computing a weighted average of the probabilities of the agent's sequence of choices in each class, with weights given by the class shares, i.e. the variables `_prob1`, `_prob2`,...,`_probC`:

```
. **(5) Compute the choice probability**
```

8. This last step is done through the user-written program `_gprod`. Type `findit gprod` and install the package `dm71`.

```
. gen double _den=_prob1*_kbbb1
. forvalues s=2/$nclasses  {
.   replace _den=_den+_prob`s´*_kbbb`s´
.                          }
```

Finally, we compute the ratios defined in equation 7 and rearrange them in order to create individual-level variables. These are the conditional probabilities of class membership as defined in the previous section:

```
. **(6) Compute the conditional probabilities of class membership**
. forvalues s=1/$nclasses  {
.   gen double _h`s´=(_prob`s´*_kbbb`s´)/_den
.   bysort $id: egen double _H`s´=sum(_h`s´)
.                          }
```

Before starting the loop that iterates the EM recursion until convergence, we need to specify a Stata `ml` program that performs the estimation of the grouped-data model defined in equation 11:[9]

```
**(7) Provide Stata with the ML command for the grouped-data model**
. bysort $group: gen _alt=sum(1)
. su _alt
. bysort $id: gen double _id1=1 if _n<=r(mean)
. program logit_lf
.   args lnf a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17 a18 a19 a20
.   tempvar denom
.   gen double `denom´=1
.   forvalues c=2/$nclasses  {
.     replace `denom´=`denom´+exp(`a`c´´)
.                            }
.   replace `lnf´=_H1*ln(1/`denom´) if $depvar==1 & _id1==1
.   forvalues c=2/$nclasses  {
.     replace `lnf´=`lnf´+_H`c´*ln(exp(`a`c´´)/`denom´) if $depvar==1 & _id1==1
.                            }
.   replace `lnf´=0 if `lnf´==.
.   **Note: the ML command updates the class shares internally:**
.   capture drop _prob*
.   gen double _prob1=1/`denom´
.   forvalues c=2/$nclasses  {
.     gen double _prob`c´=exp(`a`c´´)/(`denom´)
.                            }
. end
```

There are two important remarks about the above-mentioned routine. First, we set to zero one vector of parameters for identification. Second, the class shares - `_prob1`,

9. The `ml` program presented in the following lines allows for up to 20 latent classes. However, the routine can be easily modified so as to account for more - or less - classes.

_prob2,...,_probC - are updated *internally* when the ml program is called.

We now present the loop that repeats the steps above until convergence:

```
. local i=1
. while `i´<= $niter{
```

To begin with, estimate again the C clogit models (one for each class) using the conditional probabilities of class membership as weights. Then, recompute the probability of the chosen alternative in each choice occasion so as to update probabilities of the agent's sequence of choices:

```
**(8) Update the probability of the agent´s sequence of choices**
. capture drop _l*  _kbbb*
. forvalues s=1/$nclasses  {
.   clogit $depvar $varlist [iw=_H`s´], group($group) technique(nr dfp)
.   predict double _l`s´
.   replace _kbb`s´=_l`s´*$depvar
.   recode _kbb`s´ 0=.
.   bysort $id: egen double _kbbb`s´=prod(_kbb`s´)
.   bysort $id: replace _kbbb`s´=. if _n!=_N
.                           }
```

Now launch the ml model for the estimation of the grouped-data log likelihood:

```
**(9) Update the class share probability**
. global classes="($varlist2)"
. forvalues s=3/$nclasses  {
.   global classes="$classes ($varlist2)"
.                           }
. ml model lf logit_lf  $classes, max
```

Once the class share probabilities have been updated *within* the ml program, the next step requires updating both the choice probability - i.e. the variable _den - and the conditional probabilities of class membership, i.e. the variables _H*:

```
**(10) Update the choice probability**
. replace _den=_prob1*_kbbb1
. forvalues s=2/$nclasses  {
.    replace _den=_den+_prob`s´*_kbbb`s´
.                           }
**(11) Update the conditional probabilities of class membership**
. drop _H*
. forvalues s=1/$nclasses  {
.     replace _h`s´=(_prob`s´*_kbbb`s´)/_den
.     bysort $id: egen double _H`s´=sum(_h`s´)
.                           }
```

Once the parameters and the conditional probabilities have been updated the routine computes the log likelihood and restarts the loop until convergence.

As pointed out in Train (2008), convergence of EM algorithms for nonparametric estimation is still controversial and constitutes an area of future research. As in Train (2008) and Weeks and Lange (1989) here we define convergence when the change in the log likelihood function from one iteration to the other is *sufficiently small*. Hence, the routine automatically stops the internal loop *before* the selected number of iterations provided that the log likelihood has not changed more than 0.001% during the last five iterations:[10]

```
. **(12) Update the log likelihood**
. capture drop _sumll
. egen _sumll=sum(ln(_den))
. sum _sumll
**Check if the log likelihood has increased from the previous 5 iterations:**
. global z=r(mean)
. local _sl`i´=$z
. if `i´>=6  {
.   local a=`i´-5
.           }
**Stop the loop if the LL has not change of more than 0.001% over the last 5 iterations:**
. if `i´>=6  {
.    local `_vpsl`i´´= -(`_sl`i´´ - `_sl`a´´)/`_sl`a´´
.    if `_vpsl`i´´<= 0.00001  {
.         local i=$niter
.                          }
.           }
**Restart the loop and display the log likelihood**
. local i=`i´ +1
. display as green "Iteration " `i´ ": log likelihood = " as yellow $z
. }
```

Once the algorithm has converged the results can be displayed by typing:

```
. forvalues s=1/$nclasses  {
.     clogit $depvar $varlist [iw=_H`s´], group($group) technique(nr dfp)
.                         }
```

4.2 A model without covariates on the class shares probabilities

When the model does not include demographics on the class share probabilities the maximization of the grouped-data log likelihood can be avoided. In fact, its solution can be provided analytically as it is shown in equation 13. This is important because the maximization of the grouped-data model slows down the overall estimation process, which could become time-consuming when the number of latent classes is high.

If there are no covariates on the class share probabilities the EM algorithm can be optimized by simply dropping out the 9th step from the loop presented in the previous

10. Such a small value is advisable because - as discussed in Dempster et al. (1977) - EM algorithms may move very slowly when they are close to a maximum.

subsection. Therefore - following the routine - the choice probability is updated by means of the *previous* values of the class shares and this, in turn, allows updating the conditional probabilities of class membership during the 11th step.

Once the conditional probabilities have been updated they are used to construct the numerator and the denominator of equation 13, so that also the class share probabilities can be updated to the next iteration. Hence, the following lines should be added after the 11th step:

```
**Compute the numerator of equation 13**
. forvalues s=1/$nclasses  {
.    capture drop _nums`s´
.    egen double _nums`s´=sum(_h`s´)
.                          }
**Compute the denominator of equation 13**
. capture drop _dens
. gen double _dens=_nums1
. forvalues s=2/$nclasses  {
.    replace _dens=_dens+_nums`s´
.                          }
. **Update the class shares**
. forvalues s=1/$nclasses  {
.    replace prob`s´=nums`s´/dens
.                          }
```

Subsequently, the loop continues as before from point 12 till the end.

5 The `lclogit` command

`lclogit` is a Stata command that generalizes the routine outlined above. Therefore, this command does not contain its own *ml* evaluator and it simply makes use of the Stata `clogit` estimation command at each maximization step. This reduces significantly the estimation time and - importantly - it bases the EM estimation on the quality and the efficiency of the `clogit` evaluator.

Following the routine outlined in the previous section, `lclogit` declares convergence when the variation in the last 5 values of the maximized log likelihood is smaller than a given threshold.[11] When this happens `lclogit` stops the internal loop and displays the estimated coefficients and the class shares.

The results are displayed in a formatted table, with the columns containing the results by classes.[12] When there are more than 5 latent classes the table of results is divided in blocks of 6 columns. This should allow for a better view of the results and it

11. Users are allowed to change this threshold, the default is 0.001%.

12. `lclogit` uses the built-in programs `estimates store` and `estimates table` in order to display the table of results. Moreover, the command `estimates store` allows displaying the last set of internal `clogit` estimations by simply typing `estimates dir` after `lclogit`.

also avoid visualization problems when the table becomes too wide.[13]

The syntax for `lclogit` is:

`lclogit` *depvar* [*varlist*] [*if*] [*in*] , `group(`*varname*`)` `id(`*varname*`)` `nclasses(`*#*`)` [`options`]

The options allowed by `lclogit` are:

- `group(`*varname*`)` is required; it specifies the variable for the choice occasions.
- `id(`*varname*`)` is required; it specifies the variable for the choice-makers
- `nclasses(`*#*`)` is required; it sets the number of latent classes to be estimated.
- `seed(`*#*`)` is used to set the starting values, the default is `1234567890`;[14]
- `niter(`*#*`)` specifies the number of maximum iterations, the default is `150`;
- `convergence(`*#*`)` specifies the minimum variation with respect to the last 5 values of the log likelihood in order to declare convergence, the default is `0.001%`;
- `cpname(`*newvarname*`)`, tells `lclogit` to save C new variables called *newvarname*1, *newvarname*2,.., *newvarname*C containing the individual conditional probabilities of class membership.

6 Application

For our application we use the data illustrated in the previous section and, therefore, a model without covariates on the class share probabilities. We begin by estimating a conditional logit model using the Stata command `clogit`:

```
. use http://fmwww.bc.edu/repec/bocode/t/traindata.dta, clear
. clogit y price contract local wknown tod seasonal, group(gid)
Iteration 0: log likelihood = -1379.3159
  (output omitted)
Iteration 4: log likelihood = -1356.3867
Conditional (fixed-effects) logistic regression   Number of obs   =       4780
                                                  LR chi2(6)      =     600.47
                                                  Prob > chi2     =     0.0000
Log likelihood = -1356.3867                       Pseudo R2       =     0.1812

-------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------------------------------------------------------------------------
       price |  -.6354853   .0439523   -14.46   0.000    -.7216302   -.5493403
```

13. For models with more than 20 latent classes `lclogit` does not show the results in a formatted table but in a plain matrix, which can be also found in the `ereturn` list.
14. Note that the seed is set internally only when computing the starting values. Thereafter, the seed is set back to the original value. This allows for comparable results and simulation-based inference.

```
    contract |    -.13964   .0161887    -8.63   0.000    -.1713693   -.1079107
       local |   1.430578   .0963826    14.84   0.000     1.241672    1.619485
      wknown |   1.054535    .086482    12.19   0.000     .8850338    1.224037
         tod |  -5.698954   .3494016   -16.31   0.000    -6.383769    -5.01414
    seasonal |  -5.899944     .35485   -16.63   0.000    -6.595437   -5.204451
------------------------------------------------------------------------------
```

From the results above we can see that - on average - costumers prefer lower prices, shorter contracts length, a local and well-known company and fixed rate plans.

The Stata command **mixlogit** from Hole (2007) can be used to estimate a *parametric* mixed logit model with independent, normally-distributed coefficients:[15]

```
. mixlogit y, id(pid) group(gid) rand(price contract local wknown tod seasonal)
>  nrep(300)
Iteration 0:   log likelihood = -1249.8219  (not concave)
  (output omitted)
Iteration 7:   log likelihood = -1101.6085
Mixed logit model                                 Number of obs   =       4780
                                                  LR chi2(6)      =     509.56
Log likelihood = -1101.6085                       Prob > chi2     =     0.0000
------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Mean         |
       price |  -1.004329   .0721185   -13.93   0.000    -1.145679   -.8629798
    contract |  -.2274985    .047386    -4.80   0.000    -.3203735   -.1346236
       local |   2.208746   .2439681     9.05   0.000     1.730578    2.686915
      wknown |   1.656329   .1707167     9.70   0.000      1.32173    1.990927
         tod |  -9.364151   .5858618   -15.98   0.000    -10.51242   -8.215883
    seasonal |  -9.496181   .5792009   -16.40   0.000    -10.63139   -8.360968
-------------+----------------------------------------------------------------
SD           |
       price |   .2151655   .0311095     6.92   0.000      .154192    .2761389
    contract |    .384136    .044778     8.58   0.000     .2963728    .4718992
       local |   1.788806   .2370063     7.55   0.000     1.324282     2.25333
      wknown |   1.185838   .1731652     6.85   0.000     .8464401    1.525235
         tod |     1.6553   .2094545     7.90   0.000     1.244777    2.065824
    seasonal |  -1.119371   .2836182    -3.95   0.000    -1.675252   -.5634893
------------------------------------------------------------------------------
```

As it can be seen from the maximized log likelihood, we can reject the conditional logit specification in favor of a random coefficient model. Moreover, the magnitude of the coefficients is significantly different when compared to the estimates from **clogit**.[16]

We now show how to use **lclogit** to estimate a nonparametric mixed logit model by means of the EM algorithm outlined in the previous sections. As explained in section 1, the main idea is to use a latent class model with a relatively high number of classes so as to approximate he mixing distribution nonparametrically.

15. The Simulated Maximum Likelihood estimation is done with 300 Halton draws; **mixlogit** can be installed in Stata by typing **findit mixlogit**.
16. This is an indication of the bias produced by the IIA property of standard conditional logit models. See Bhat (2000) for this point.

As stated in Greene and Hensher (2003) and Train (2008), the choice of the appropriate number of classes is made by means of some information criteria. Here we opt for the BIC and the CAIC, which penalize more heavily models with a large number of parameters.[17] The next lines show how to use `lclogit` to estimate a latent class model with an increasing number of classes:

```
. forvalues c=2/15  {
.   lclogit y price contract local wknown tod seasonal, gr(gid) id(pid) ncl(`c´)
.   display e(bic)
.   display e(caic)
.   display e(ll)
.                  }
```

The routine took about 30 minutes to estimate the whole set of 14 models on our standard-issue PC.[18] The next table shows - for an increasing number of latent classes - the maximized log likelihood, the number of parameters, the BIC and the CAIC:

Classes	Log Likelihood	N.param.	CAIC	BIC
Cl.1	-1356.39	6	2746.40	2740.40
Cl.2	-1211.35	13	2495.57	2482.57
Cl.3	-1118.23	20	2348.57	2328.57
Cl.4	-1085.30	27	2321.95	2294.95
Cl.5*	-1040.49	34	2271.55*	2237.55
Cl.6	-1028.56	41	2286.93	2245.93
Cl.7	-1006.37	48	2281.79	2233.79
Cl.8*	-990.24	55	2288.76	2233.76*
Cl.9	-983.64	62	2314.80	2252.80
Cl.10	-979.23	69	2345.22	2276.22
Cl.11	-965.76	76	2357.52	2281.52
Cl.12	-952.68	83	2370.58	2287.58
Cl.13	-947.24	80	2398.94	2308.94
Cl.14	-945.59	97	2434.89	2337.89
Cl.15	-943.42	104	2469.78	2365.78

From these results we can infer that the 5-class model is optimal according to the CAIC, whilst the Bayesian criterion points to a model with 8 latent classes.

The output below shows how the program works for a model with 8 latent classes:

```
. display c(current_time)
12:48:14
.  lclogit y price contract local wknown tod seasonal, id(pid) gr(gid) ncl(8)
Iteration 1:  log likelihood = -1252.5014
Iteration 2:  log likelihood = -1094.1094
Iteration 3:  log likelihood = -1043.4077
Iteration 4:  log likelihood = -1031.4924
Iteration 5:  log likelihood = -1024.0092
Iteration 6:  log likelihood = -1016.9219
```

17. `lclogit` returns in `e()` three different information criteria: the AIC, the CAIC and the BIC.
18. We used a PC with a 2.2GHz Intel core 2 duo and 4MB RAM.

```
Iteration 7:  log likelihood = -1008.2313
Iteration 8:  log likelihood = -1002.0564
Iteration 9:  log likelihood = -998.89545
Iteration 10:  log likelihood = -996.875
Iteration 11:  log likelihood = -995.84229
Iteration 12:  log likelihood = -995.19885
Iteration 13:  log likelihood = -994.69055
Iteration 14:  log likelihood = -994.31946
Iteration 15:  log likelihood = -994.08838
```

(*output omitted*)

```
Iteration 45:  log likelihood = -990.23932
Iteration 46:  log likelihood = -990.23895
Iteration 47:  log likelihood = -990.23877
Iteration 48:  log likelihood = -990.23865
Iteration 49:  log likelihood = -990.23859
Iteration 151:  log likelihood = -990.23853
Latent class model with 8 latent classes
```

Variable	Class1	Class2	Class3	Class4	Class5
price	-0.910	-0.737	-0.488	-2.110	-0.642
contract	-0.438	0.218	-0.592	-0.662	0.096
local	0.370	2.416	0.782	0.717	2.186
wknown	0.369	2.840	0.710	0.241	1.207
tod	-8.257	-6.690	-4.132	-14.191	-3.836
seasonal	-6.440	-7.213	-6.560	-17.207	-4.052
Prob	0.120	0.097	0.091	0.070	0.096

Variable	Class6	Class7	Class8
price	-1.208	-1.533	-0.082
contract	-0.198	-0.409	-0.156
local	6.578	0.621	4.937
wknown	5.103	0.930	3.444
tod	-14.847	-16.007	-1.088
seasonal	-15.334	-14.818	-1.060
Prob	0.111	0.236	0.178

```
Note: model estimated via EM algorithm
. display c(current_time)
12:49:20
```

As it can be seen, the EM algorithm took 49 iterations before reaching convergence, i.e. about one minute in our standard-issue PC. However, the routine is already close to the maximum at the 11th iteration, i.e. after less than 10 seconds. This is a common feature of EM algorithms and it actually suggests another useful application of `lclogit`, as it could be used to obtain good starting values for the estimation of latent class models via gradient-based algorithms.[19]

19. This could be particularly useful either to speed up the estimation process or to avoid convergence problems when estimating models with a high number of latent classes.

As Train (2008) points out, when the number of classes rises summary statistics for the distribution of each coefficient could be more informative than the single point estimates. For this reason, `lclogit` stores the predicted weighted average of each coefficient in the vector `e(PB)`:

```
. matrix list e(PB)
e(PB)[1,6]
       Average of  Average of  Average of  Average of  Average of  Average of
            price    contract       local      wknown         tod    seasonal
Coeff    -.94577     -.26901      2.3690      1.9184     -9.0036      -9.058
```

Interestingly, although the parameters are rather different from class to class their weighted averages are close to the correspondent values obtained from the parametric estimation via `mixlogit`.

7 Conclusions

This article has shown how to estimate a nonparametric mixed logit model in Stata with one of the methods proposed in Train (2008). The method makes use of an EM algorithm that - thanks to its desirable properties - allows estimating a latent class model with a high number of classes so that the unobserved distribution of the coefficients can be approximated nonparametrically. We have also shown how to use the Stata command `lclogit`, which performs the EM estimation for latent class logit models automatically.

8 Acknowledgements

I am grateful to Kenneth Train for helpful comments on the implementation of the EM algorithm. Thanks also to an anonymous referee for relevant comments and suggestions on how to improve both this paper and the `lclogit` routine.

9 References

Bhat, C., 2000. Flexible model structures for discrete choice analysis. Handbook of transport modelling 1, 71-90.

Boyles, R., 1983. On the convergence of the em algorithm, Journal of the Royal Statistical Society B 45, 47-50.

Dempster, A., Laird, N., Rubin, D., et al., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39 (1), 1-38.

Greene, W., Hensher, D., 2003. A latent class model for discrete choice analysis: contrasts with mixed logit. Transportation Research Part B 37 (8), 681-698.

Hole, A.R., 2007. Fitting mixed logit models by using maximum simulated likelihood. Stata Journal, 7 (3), 388-401.

Huber, J. and K. Train, 2001. On the similarity of classical and bayesian estimates of individual mean partworths, Marketing Letters 12, 259-269

McFadden, D., 1973. Conditional logit analysis of qualitative choice models. Frontiers of Econometrics, ed. P. Zarembka. New York: Academic Press.

McFadden, D., Train, K., 2000. Mixed MNL models for discrete responses. Journal of Applied Econometrics 15 (5), 447-470.

Train, K., 2003. Discrete choice methods with simulation. Cambridge Books.

Train, K., 2008. EM Algorithms for Nonparametric Estimation of Mixing Distributions. Journal of Choice Modelling 1 (1).

Weeks, D. and K. Lange 2007. Trials,tribulations, and triumphs of the EM algorithm in pegigree analysis. Journal of Mathematics Applied in Medicine and Biology 6, 209-232.

Wu, C., 1983. On the convergence properties of the EM algorithm. The Annals of Statistics 11 (1), 95-103.

About the author

Daniele Pacifico works with the Italian Department of the Treasury (Rome, Italy) and is a research fellow of the Centre for the Analysis of Public Policies (Modena, Italy).

“Materiali di Discussione” LATER PUBLISHED ELSEWHERE

N. 546 - M. Murat and B. Pistoresi, *Emigrants and immigrants networks in FDI*, Applied Economics letters, April 2008, **http://www.informaworld.com/ /content~content=a789737803~db=all~order=author** (electronic publication), **WP No. 546 (December 2006).**

N. 545 - M.Brunetti and C. Torricelli, *The Population Ageing in Italy:Facts and Impact on Household Portfolios*, in M. Balling & E. Gnan & F. Lierman (eds.), Money, Finance and Demography: The Consequences of Ageing, Vienna, Suerf (2007), **WP No. 545 (November 2006).**

N. 532 – M: Montanari, Between European Integration and Regional Autonomy: The Case of Italy from an Economic Perspective, Constitutional Political Economy, Vol. 17, 4, pp. 277-301 (2006), **WP No. 532 (March 2006).**

N. 529 - M. Montanari, *Knocking on the EU’s door: the Political Economy of EU-Ukraine Relations*, Journal of Contemporary European Research, Vol.3,1, pp.64-78 (2007), **WP No. 529 (February 2006).**

N. 518 - M.Brunetti and C. Torricelli, *Economic Activity and Recession Probabilities: information content and predictive power of the term spread in Italy*, Applied Economics (2009), **WP No. 518 (December 2005).**

N. 517 - M. Murat and S. Paba (2006), *I distretti industriali tra immigrazioni e internazionalizzazione produttiva*, in B. Quintieri (ed.) I distretti italiani dal locale al globale, Rubbettino (2006), **WP No. 517 (December 2005).**

N. 491 - V. Moriggia, S. Muzzioli and C. Torricelli, *On the no arbitrage condition in option implied trees*, European Journal of Operational Research (2009), **WP No. 491 (May 2005).**

N. 482 - G. Di Lorenzo and G. Marotta, *A less effective monetary transmission in the wake of EMU? Evidence from lending rates passthrough***,** ICFAI Journal of Monetary Economics, Vol. 4, 2, pp. 6-31 (2006), **WP No. 482 (February 2005).**

N. 472 - M.Brunetti and C. Torricelli, *The internal and cross market efficiency in index option markets: an investigation of the Italian market*, Applied Financial Economics, Vol. 17, 1, pp. 25-33 (2007), **WP No. 472 (November 2004).**

N. 466 - G. Marotta, *La finanza del settore non profit tra ritardi nei pagamenti e Basilea 2*, Banca Impresa Società , Vol. XXIV, 1, pp. 35-51 (2005), **WP No. 466 (September 2004).**

N. 453 - Pederzoli and C. Torricelli, *Capital requirements and Business Cycle Regimes: Forward-looking modelling of Default Probabilities* , Journal of Banking and Finance, Vl. 29, 12, pp. 3121-3140 (2005), **WP No. 453 (February 2004).**

N. 448 - V. Moriggia, S. Muzzioli, C. Torricelli, *Call and put implied volatilities and the derivation of option implied trees*, Frontiers In Finance and Economics, vol.4, 1, pp. 35-64 (2007), **WP No. 448 (November 2003).**

N. 436 - M.Brunetti and C. Torricelli, *Put-Call Parity and cross-market efficiency in the Index Options Markets: evidence from the Italian market*, International Review of Financial Analysis, Vl.14, 5, pp. 508-532 (2005), **WP No. 436 (July 2003).**

N. 429 - G. Marotta, *When do trade credit discounts matter? Evidence from Italian Firm-Level Data*, Applied Economics, Vol. 37, 4, pp. 403-416 (2005), **WP No. 429 (February 2003).**

N. 426 - A. Rinaldi and M. Vasta, *The Structure of Italian Capitalism, 1952-1972: New Evidence Using the Interlocking Directorates Technique*, Financial History Review, vol, 12, 2, pp. 173-198 (2005), **WP No. 426 (January 2003).**

N. 417 - A. Rinaldi, *The Emilian Model Revisited: Twenty Years After*, Business History, vol. 47, 2, pp. 244-226 (2005), **WP No. 417 (September 2002).**

N. 375 - G. Marotta, *La direttiva comunitaria contro i ritardi nei pagamenti tra imprese. Alcune riflessioni sul caso italiano*, Banca, Impresa, Società, Vol. XX, 3, pp. 451-71 (2001), **WP No. 375 (September 2001).**

N. 303 - G. Marotta and M. Mazzoli, *Fattori di mutamento nella domanda di prestiti ed effetti sulla trasmissione della politica monetaria*, in P. ALESSANDRINI (ed.) Il sistema finanziario italiano tra globalizzazione e localismo, Bologna, Il Mulino, pp. 223-260 (2001), **WP No. 303 (April 2000)**

N. 131 - G. Marotta, *Does trade credit redistribution thwart monetary policy? Evidence from Italy*, Applied Economics, Vol. 29, December, pp. 1619-29 (1997), **WP No. 131 (1996).**

N. 121 - G. Marotta, *Il credito commerciale in Italia: una nota su alcuni aspetti strutturali e sulle implicazioni di politica monetaria*, L'Industria, Vol. XVIII, 1, pp. 193-210 (1997), **WP No. 121 (1995)**

N. 105 - G. Marotta, *Credito commerciale e "lending view"*, Giornale degli Economisti e Annali di Economia, Vol. LIV, 1-3, gennaio-marzo, pp. 79-102; anche in G. Vaciago (a cura di) *Moneta e finanza*, Bologna, Il Mulino (1995), **WP No. 105 (1994).**

RECENTLY PUBLISHED "Materiali di Discussione"

N. 662 - *From SHIW to IT-SILC: construction and representativeness of the new CAPP_DYN first-year population*, by Emanuele Ciani and Donatella Fresu [July 2011].

N. 661 - *Trends and dynamics in the Italian labour market. An empirical evaluation using RFL data, 1993-2007*, by Sara Flisi and Marcello Morciano [July 2011].

N. 660 - *Estimation and Simulation of Earnings in IT-SILC*, by Emanuele Ciani e Marcello Morciano [July 2011].

N. 659 - *Growth, Colonization, and Institutional Development In and Out of Africa*, by Graziella Bertocchi [July 2011].

N. 658 - *Economia e finanza nelle imprese meccaniche delle regioni del Nord Italia 2000-2008. Primo rapporto di ricerca*, by Luciana Canovi, Margherita Russo, Monica Baracchi and Daniela Bigarelli [June 2011].

N. 657 - *Innovation networks: theory and policy implications at regional level. The case of Tuscany' innovation policies 2000-2006*, by Annalisa Caloffi, Federica Rossi and Margherita Russo [May 2011].

N. 656 - *Il sistema di istruzione nella promozione dello sviluppo economico. Strategie pubbliche e interventi privati a Modena*, by Paola Mengoli and Alberto Rinaldi [May 2011].

N. 655 - *Exports and Italy's economic development: a long-run perspective (1863-2004)*, by Barbara Pistoresi and Alberto Rinaldi [May 2011].

N. 654 - *Transnational social capital and FDI. Evidence from Italian associations worldwide*, by Marina Murat, Barbara Pistoresi and Alberto Rinaldi [May 2011].

N. 653 - *Le difficoltà d'accesso all'abitazione da parte degli immigrati. Una indagine di campo*, by Paola Bertolini, Francesco Pagliacci and Serena Giannuzzi [April 2011].

N. 652 - *Il social housing in Europa*, by Massimo Baldini Baldini and Marta Federici [April 2011].

N. 651 - *L'edilizia sociale di fronte all'immigrazione a Modena: politiche abitative e domanda potenziale (o inevasa)*, by Marta Federici and Giuseppe Fiorani [April 2011].

N. 650 - *Dynamic Adverse Selection and the Size of the Informed Side of the Market*, by Ennio Bilancini and Leonardo Boncinelli [March 2011].

N. 649 - *Present and Future of the Chinese labour Market*, by Michele Bruni and Claudio Tabacchi [March 2011].

N. 648 - *Lisbon strategy and EU countries' performance: social inclusion and sustainability*, by Paola Bertolini and Francesco Pagliacci [March 2011].

N. 647 - *Schools choices of foreign youth in Italian territorial areas*, by Paola Bertolini, Valentina Toscano and Linda Tosarelli [March 2011].

N. 646 - *Allocation of Time within Italian Couples: Exploring the Role of Institutional Factors and their Effects on Household's Wellbeing*, by Tindara Addabbo Antonella Caiumi and Anna Maccagnan [February 2011].