Alternative methods for parameter estimation in discrete latent variable models

Luca Brusa[1], Francesco Bartolucci[2], Fulvia Pennoni[1]
(luca.brusa@unimib.it)

[1]University of Milano-Bicocca - Department of Statistics, and Quantitative Methods
[2]University of Perugia - Department of Economics

December 18, 2022

Outline

Latent variable models and EM algorithm

- Maximum likelihood estimation of latent variable model parameters $\boldsymbol{\theta}$ is based on the complete data log-likelihood function $\ell^*(\boldsymbol{\theta})$
- It is often carried out through the **Expectation-Maximization (EM)** algorithm (Dempster et al., 1977)
- Alternate the following steps until a suitable convergence condition:
  - **E-step**: compute the conditional expected value of $\ell^*(\boldsymbol{\theta})$, given the observed data and the value of the parameters at the previous step
  - **M-step**: maximize the expected value of $\ell^*(\boldsymbol{\theta})$ and so update the model parameters

Convergence to local maxima

- Is straightforward to implement, is able to converge in a stable way to a local maximum of the log-likelihood function and is used for parameter estimation in many available packages
- A well-known drawback is related to the **multimodality** of the log-likelihood function, especially when the model has many latent components; therefore the global maximum is not ensured to be reached
- In the following we address this problem by considering two special classes of discrete latent variable models, namely **latent class** (LC, Goodman, 1974) and **hidden Markov** (HM, Bartolucci et al., 2013)

Solutions

- Current approach: **multi-start strategy** based on deterministic and random rules. It is computationally intensive and does not guarantee convergence to the global maximum

Proposed approaches

- **Tempering** and **annealing** techniques: the objective function is re-scaled on the basis of a variable, known as **temperature**, which controls the prominence of global and local maxima
- **Evolutionary** algorithms: many candidate solutions are considered and evaluated. The best candidates, according to some quality measure, are selected for successive steps, the worst ones are discarded

Outline

Derivation

- We implement the **tempered EM** (**T-EM**) algorithm, by adjusting the computation of the conditional expected frequencies in the E-step on the basis of the temperature, denoted with τ in the following
- By properly **tuning the sequence of temperature** values, the procedure is gradually attracted towards the global maximum, escaping local sub-optimal solutions:
  - **high temperatures** allow exploring wide regions of the parameter space, avoiding being trapped in non-global maxima
  - **low temperatures** guarantee a sharp optimization in a local region of the solution space

Tempering profile

- We define a sequence of temperatures $(\tau_h)_{h \geq 1}$, such that:
  - τ_1 is sufficiently large so that the optimization function is relatively flat
  - τ_h tends towards 1 as the algorithm iteration counter increases, to recover the original function
- The resulting sequence, known as **tempering profile**, guarantees a proper convergence of the T-EM algorithm
- To ensure flexibility to the tempering profile, it depends on a set of constants; a suitable grid-search procedure is employed to select the optimal configuration of tempering constants

Tempering profile

In particular, we consider two classes of tempering profiles:

- a **monotonically decreasing exponential** (M-T-EM) profile:

$$\tau_h = 1 + e^{\beta - h/\alpha}$$

- a **non-monotonic** profile with **gradually smaller oscillations** (O-T-EM)

$$\tau_h = \tanh\left(\frac{h}{2\rho}\right) + \left(T_0 - \beta\,\frac{2\sqrt{2}}{3\pi}\right)\alpha^{h/\rho} + \beta\,\mathrm{sinc}\left(\frac{3\pi}{4} + \frac{h}{\rho}\right)$$

Outline

Derivation

- We implement the **evolutionary EM** (**E-EM**) algorithm, by incorporating the E-step and the M-step into an evolutionary framework
- Inspired by the basic principles of the Darwinian theory of biological evolution:
  - The individuals in a population are represented by candidate solutions of the optimization problem (model parameters)
  - Reproduction and mutation of the individuals ensure an adequate exploration of the solution space
  - Selection of the best individuals is performed through a fitness function that determines the quality of each candidate solution

Main steps

- A population P_0 is initialized with n_p sets of random starting values
- The following steps are iterated until a suitable convergence criterion is satisfied:
  1. $P_1 \leftarrow$ **Update**(P_0)
  2. $P_2 \leftarrow$ **Crossover**(P_1)
  3. $P_3 \leftarrow$ **Update**(P_2)
  4. $P_4 \leftarrow$ **Select**(P_3)
  5. $P_5 \leftarrow$ **Mutate**(P_4)
- The best result from population P4 is selected and updated through a complete run of the EM algorithm until convergence

Main steps

- A population P_0 is initialized with n_p sets of random starting values
- The following steps are iterated until a suitable convergence criterion is satisfied:
  1. $P_1 \leftarrow$ **Update**(P_0): each individual from population P_0 is updated by performing a small number R of steps of the standard EM algorithm
  2. $P_2 \leftarrow$ **Crossover**(P_1)
  3. $P_3 \leftarrow$ **Update**(P_2)
  4. $P_4 \leftarrow$ **Select**(P_3)
  5. $P_5 \leftarrow$ **Mutate**(P_4)
- The best result from population P4 is selected and updated through a complete run of the EM algorithm until convergence

Main steps

- A population P_0 is initialized with n_p sets of random starting values
- The following steps are iterated until a suitable convergence criterion is satisfied:
  1. $P_1 \leftarrow$ **Update**(P_0)
  2. $P_2 \leftarrow$ **Crossover**(P_1): pairs of distinct individuals are randomly selected from population P_1 and combined to obtain the n_c offspring of new population P_2
  3. $P_3 \leftarrow$ **Update**(P_2)
  4. $P_4 \leftarrow$ **Select**(P_3)
  5. $P_5 \leftarrow$ **Mutate**(P_4)
- The best result from population P4 is selected and updated through a complete run of the EM algorithm until convergence

Main steps

- A population P_0 is initialized with n_p sets of random starting values
- The following steps are iterated until a suitable convergence criterion is satisfied:
  1. $P_1 \leftarrow$ **Update**(P_0)
  2. $P_2 \leftarrow$ **Crossover**(P_1)
  3. $P_3 \leftarrow$ **Update**(P_2): each individual from population P_2 is updated by performing a small number R of steps of the standard EM algorithm
  4. $P_4 \leftarrow$ **Select**(P_3)
  5. $P_5 \leftarrow$ **Mutate**(P_4)
- The best result from population P4 is selected and updated through a complete run of the EM algorithm until convergence

Main steps

- A population P_0 is initialized with n_p sets of random starting values
- The following steps are iterated until a suitable convergence criterion is satisfied:
  1. $P_1 \leftarrow$ **Update**(P_0)
  2. $P_2 \leftarrow$ **Crossover**(P_1)
  3. $P_3 \leftarrow$ **Update**(P_2)
  4. $P_4 \leftarrow$ **Select**(P_3): individuals from populations P_1 and P_3 are considered jointly and the n_p with the highest value of the complete log-likelihood function are selected for the next generation P_4
  5. $P_5 \leftarrow$ **Mutate**(P_4)
- The best result from population P4 is selected and updated through a complete run of the EM algorithm until convergence

Main steps

- A population P_0 is initialized with n_p sets of random starting values
- The following steps are iterated until a suitable convergence criterion is satisfied:
  1. $P_1 \leftarrow$ **Update**(P_0)
  2. $P_2 \leftarrow$ **Crossover**(P_1)
  3. $P_3 \leftarrow$ **Update**(P_2)
  4. $P_4 \leftarrow$ **Select**(P_3)
  5. $P_5 \leftarrow$ **Mutate**(P_4): single individuals are randomly selected from population P_4 and minor changes are introduced with a certain probability
- The best result from population P4 is selected and updated through a complete run of the EM algorithm until convergence

Outline

Simulation study

- **Different scenarios** (5 for LC, 6 for categorical HM, 5 for continuous HM) to explore the performance for many combinations of parameters
- Extensive **Monte Carlo simulation study**:
  - draw 50 samples for each scenario
  - select 100 random starting values for each sample
  - consider both correctly specified and misspecified models
  - fit the model using EM, T-EM, and E-EM algorithms
- Criteria to compare the behavior of the three algorithms:
  - **mean** and **median** of the maximized log-likelihood values
  - **ability to reach the global maximum**
  - normalized **mean distance** from the global maximum

Settings of the algorithms

- The convergence of the EM algorithm is checked on the basis of both the relative change in the log-likelihood of two consecutive steps, and the distance between the corresponding parameter vectors
- T-EM algorithm:
  - Only the monotonic tempering profile is used
  - The tempering parameters α and β are optimally tuned through a grid-search procedure (time consuming)
- E-EM algorithm:
  - $R = 20$, $n_p = 15$, $n_c = 30$

Correctly specified models

- Both the T-EM and E-EM algorithm show a clear advantage with respect to the standard EM algorithm
- The improvement obtained with the two proposed algorithms is generally very similar when models with a few latent components ($k = 3$) are considered
- Focusing on the cases with many latent components ($k = 6$), the performance of the E-EM algorithm is considerably superior also with respect to the tempered approach
- In particular the probability to reach the global maximum is, on average, very close to 100% when the E-EM algorithm is used

Correctly specified models

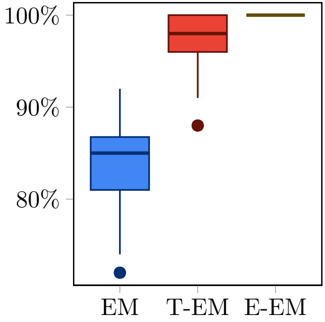

Figure: Hidden Markov model (with continuous response variables); correctly specified model with $k = 3$ latent components

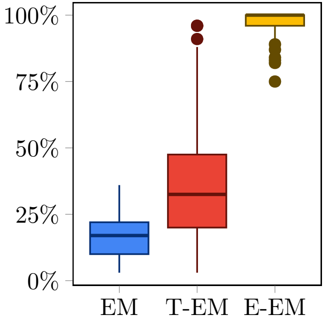

Figure: Latent class model (with categorical response variables); correctly specified model with $k = 6$ latent components

Misspecified models

- In this case the E-EM algorithm ensures a significant advantage over both the standard and the tempered versions
- In the cases with few latent components, the E-EM algorithm reaches the global maximum with a very high frequency (close to 100%), being able to avoid all local maxima
- Considering the models with many latent components, the E-EM algorithm ensures the best results, even though the frequencies are lower than in the previous cases
- The T-EM algorithm provides an improvement with respect to the standard version, but the performance is inferior to the one of the evolutionary approach

Misspecified models

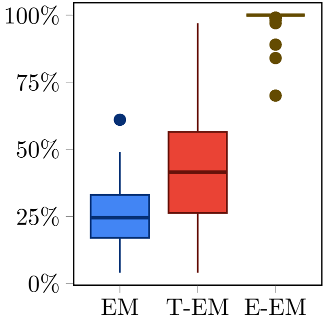

Figure: Hidden Markov model (with categorical response variables); misspecified model with $k = 4$ latent components

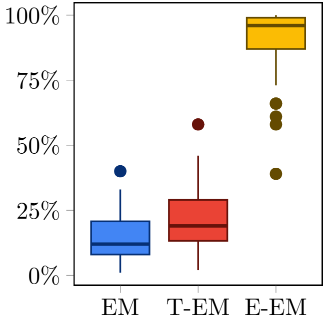

Figure: Hidden Markov model (with continuous response variables); correctly specified model with $k = 7$ latent components

Outline

Conclusion and comparison

- Both proposed approaches clearly show superior performance with respect to the standard EM algorithm
- The E-EM algorithm generally provides the best results, even compared to the tempered version
- The evolutionary approach ensures a simple interpretation: n_p and n_c represent the number of parent and children individuals, respectively. None of the tempering parameters has such a clear meaning
- The E-EM algorithm does not need a tuning procedure for the parameters, thus resulting in much less time consuming than the T-EM algorithm

Outline

References I

ASHLOCK, D. (2004). *Evolutionary Computation for Modeling and Optimization*. Springer, New York.

BARTOLUCCI, F., FARCOMENI, A., AND PENNONI, F. (2013). *Latent Markov models for longitudinal data*. Chapman & Hall/CRC, Boca Raton.

BARTOLUCCI, F., PANDOLFI, S., AND PENNONI, F. (2022). Discrete latent variable models. *Annual Review of Statistics and its Application*, **6**, 1–31.

BRUSA, L., BARTOLUCCI, F., AND PENNONI, F. (2022). Tempered Expectation-Maximization algorithm for discrete latent variable models. *Computational Statistics*, pages 1–34.

DEMPSTER, A., LAIRD, N., AND RUBIN, D. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, **39**, 1–38.

References II

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

McNicholas, S., McNicholas, P., and Ashlock, D. (2021). An evolutionary algorithm with crossover and mutation for model-based clustering. *Journal of Classification*, **38**, 264–279.

Pernkopf, F. and Bouchaffra, D. (2005). Genetic-based em algorithm for learning gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 1344–1348.

Sambridge, M. (2013). A parallel tempering algorithm for probabilistic sampling and multimodal optimization. *Geophysical Journal International*, **196**, 357–374.

Ueda, N. and Nakano, R. (1998). Deterministic annealing EM algorithm. *Neural Networks*, **11**, 271–282.

Zhou, H. and Lange, K. (2010). On the bumpy road to the dominant mode. *Scandinavian Journal of Statistics*, **37**, 612–631.