AECM Algorithm on Factor HMM-VB

Lixiang Zhang

Pennsylvania Stata University

lzz46@psu.edu

December 12, 2019

Preliminary

- Mixture Model, e.g. Gaussian Mixture Model.
  $f(x) = \sum_{k=1}^{K} \pi_k \phi(x|\theta_k = \mu_k, \Sigma_k)$.
- Hidden Markov Model (HMM) can be viewed as a special kind of Mixture Model.
- Hidden Markov Model on Variable Blocks (HMM-VB).
  High-dimensional data with sequential dependence structure among groups of variables.
- View the sequential ordering of the variable blocks as a "timeline", it is natural to employ a HMM-type model. Clustering using the underlying state sequence.
- Difference between HMM: Each variable block is of different dimensions, and follows their own Mixture Model.

Motivation

- The dimension of each variable block is still very high. Force covariance matrix to be diagonal to avoid singular matrix issue.
- It is tempting if we get a dimension reduction result at the same time.
- Factor Mixture Model. $Y_i = \mu + BU_i + \epsilon_i,\ i = 1, ..., n$, where $Y_i \in R^p, U_i \in R^q, q < p$. $U_i \sim N(0, I_q), \epsilon_i \sim N(0, D)$, D diagonal.
- Conditional on the factors U_i, Y_i are independently distributed from $N(\mu + BU_i, D)$. Unconditionally, Y_i are i.i.d from $N(\mu, BB^T + D)$.
- $Y_i = \mu_k + B_k U_{ki} + \epsilon_{ki},\ i = 1, ..., n$, with a mixture component prior probability π_k.
- We can write the density function for variable block $x^{(t)}$ as $f(x^{(t)}|\theta^{(t)}) = \sum_{k=1}^{M_t} \pi_k \phi(x^{(t)}|\mu_k^{(t)}, B_k^{(t)} {B_k^{(t)}}^T + D_k^{(t)})$.

Estimation and Challenge

- The density of Factor HMM-VB is
$f(\mathbf{x}) = \sum_{\mathbf{s}\in\hat{\mathcal{S}}} \left(\pi_{s_1} \prod_{t=1}^{T-1} a^{(t)}_{s_t,s_{t+1}}\right) \cdot \prod_{t=1}^{T} \phi\left(x^{(t)}|\mu^{(t)}_{s_t}, \Sigma^{(t)}_{s_t}\right).$
- Usually using EM to estimate HMM. For HMM-VB is has exponential complexity. Already solved by Baum-Welch (BW) algorithm.
- Challenge here for Factor HMM-VB. EM is painfully slow for Factor Mixture Model. Its missing data contains both component indicator vectors and latent factor vectors. It is even worse in Factor HMM-VB as each variable block follows their own Factor Mixture.
- Recipe: Alternating Expectation Conditional-Maximization (AECM) Algorithm.

High Level Main Idea of AECM

1. AECM is an extension of ECM. ECM algorithm replaces the M-step of the EM algorithm by a number of computationally simpler conditional maximization (CM) steps.
2. Alternating means that complete-data is allowed to be different on each CM-step.

Algorithm Details

- Estimating the transition probability matrices between variable blocks and the posterior probability are not the focus of this talk. I would just illustrate the update about one variable block $x^{(t)}$.
- We partition the parameter vector Ψ into (Ψ_1, Ψ_2). Ψ_1 contains component means μ_i and mixing proportions π_i. Ψ_2 contains elements from loading matrix B_i and diagonal matrix D_i.
- In each iteration, we can use BW algorithm to get $L_k(x_j, t)$ as the posterior probability for variable block $x^{(t)}$.

First CM

- Updating only Ψ_1 and specify the missing data to be just component id vectors z_{ij}.
- Now the complete log-likelihood for variable block t is: $\log L_c(\Psi) = \sum_{i=1}^{g} \sum_{j=1}^{n} z_{ij} \log \left\{ \pi_i \phi \left(x_j^{(t)}; \mu_i^{(t)}, \Sigma_i^{(t)} = B_i^{(t)} {B_i^{(t)}}^T + D_i^{(t)} \right) \right\}$.
- On the $(k+1)$th iteration, $Q_1\left(\Psi; \Psi^{(k)}\right) = E_{\Psi^{(k)}} \{\log L_c(\Psi) | x_o\}$.
- Simply update: $\mu_i^{(t)} = \frac{\sum_{j=1}^{n} L_i^{(k)}(\mathbf{x}_j, t) x_j^{(t)}}{\sum_{j=1}^{n} L_i^{(k)}(\mathbf{x}_j, t)}$ and $\pi_i \propto \sum_{j=1}^{n} L_i^{(k)}(\mathbf{x}_j, 1)$
- Now we set $\Psi^{(k+1/2)} = (\Psi_1^{(k+1)}, \Psi_2^{(k)})$.

Second CM

- Updating Ψ_2 and specify the missing data to be both component id vectors and factors U_{ij}.
- Now $Q_2\left(\Psi; \Psi^{(k+1/2)}\right) = E_{\Psi^{(k+1/2)}}\left\{\log L_c(\Psi)|x_o\right\}$.
- Omit many calculation details. The updates:
$B_i^{(k+1)} = V_i^{(k+1/2)}\gamma_i^{(k)}\left(\gamma_i^{(k)^T} V_i^{(k+1/2)}\gamma_i^{(k)} + \omega_i^{(k)}\right)^{-1}$,
$D_i^{(k+1)} = \text{diag}\left\{V_i^{(k+1/2)} V_i^{(k+1/2)}\gamma_i^{(k)} B_i^{(k+1)^T}\right\}$.
- V, γ, ω are all some intermediate matrix results. But many inverse has the form: $(BB^T + D)^{-1}, B \in R^{p*q}, D \in R^{p*p} diagonal$. It equals $D^{-1} - D^{-1}B\left(I_q + B^T D^{-1} B\right)^{-1} B^T D^{-1}$.

Something Interesting

1. By the construction of AECM algorithm, it still has the ascending property.
2. Initialization issue. Can not use $B = 0, D =$ diagonal sample covariance matrix as the initialization.
   So we need a good initial estimate of B and D. In factor analysis, estimate using eigenvalue decomposition.
   $S \approx \lambda_1 l_1 l_1' + \cdots + \lambda_q l_q l_q' + D$
   $$= (\sqrt{\lambda_1} l_1, \cdots, \sqrt{\lambda_q} l_q) \begin{pmatrix} \sqrt{\lambda_1} l_1' \\ \vdots \\ \sqrt{\lambda_q} l_q' \end{pmatrix} + \begin{pmatrix} \sigma_1^2 \cdots & 0 \\ \ddots & \\ 0 & \cdots \sigma_p^2 \end{pmatrix}.$$

Simulation

- The set up of the simulation is to connect 50 10-dimensional Factor mixtures into a 500 dimension vector to generate sample.
- HMM-VB was first written in a package by Prof. Jia Li, using more than 6000 lines of code in C.
- Previous student used Rcpp to make it into an R package called HDclust. I have finished modified C code into Factor HMM-vb version and the corresponding changes in HDclust. But when I call the function in R: