Particle Learning for General Mixtures

Hedibert Freitas Lopes[1]

Booth School of Business
University of Chicago

Dipartimento di Scienze delle Decisioni
Università Bocconi, Milano

[1] Joint work with Nicholas Polson, Carlos Carvalho and Matt Taddy

Contributions

1. Efficiently and sequentially learn from general mixture models

$$f(z) = \int k(z;\theta)dG(\theta)$$

   where G is almost surely discrete.

2. When we say general, we mean general
   - Finite mixture models;
   - Dirichlet process mixture models;
   - Indian buffet processes;
   - Probit stick-breaking models.
3. An alternative to MCMC
   - On-line model fitting and marginal likelihood estimation;
   - Posterior cluster allocation;
   - Handling high dimensional data-sets.

Particle Learning (PL)

1. General framework of sequential parameter learning;
2. Practical alternative to MCMC[2];
3. Sequential Monte assessment;
4. Resample-sample is key[3];
5. *Essential state vector* generalizes sufficient statistics[4];
6. Connection to Rao-Blackwellization[5];
7. PL is not sequential importance sampling;
8. Smoothing offline.

[2] Chen and Liu (1999).

[3] Pitt and Shephard (1999).

[4] Storvik (2002) and Fearnhead (2002)

[5] Kong, Liu and Wong (1994).

The general PL algorithm

- Posterior at t: $\Phi_t \equiv \left\{(x_t, \theta)^{(i)}\right\}_{i=1}^N \sim p(x_t, \theta | y^t)$.
- Compute, for $i = 1, \ldots, N$,

$$w_{t+1}^{(i)} \propto p(y_{t+1} | x_t^{(i)}, \theta^{(i)})$$

- Resample from Φ_t with weights w_{t+1}: $\tilde{\Phi}_t \equiv \left\{(\tilde{x}_t, \tilde{\theta})^{(i)}\right\}_{i=1}^N$.
- Propagate states

$$x_{t+1}^{(i)} \sim p(x_{t+1} | \tilde{x}_t^{(i)}, \tilde{\theta}^{(i)}, y_{t+1})$$

- Update sufficient statistics

$$s_{t+1}^{(i)} = \mathcal{S}(s_t^{(i)}, x_{t+1}^{(i)}, y_{t+1})$$

- Sample parameters

$$\theta^{(i)} \sim p(\theta | s_{t+1}^{(i)})$$

Example: Nonlinear, nonlinear dynamic model

Heavy-tail observation errors can be introduced as follows:

$$y_{t+1} = x_{t+1} + \sigma\sqrt{\lambda_{t+1}}\epsilon_{t+1}$$
$$x_{t+1} = \beta\frac{x_t}{1+x_t^2} + \tau u_{t+1}$$

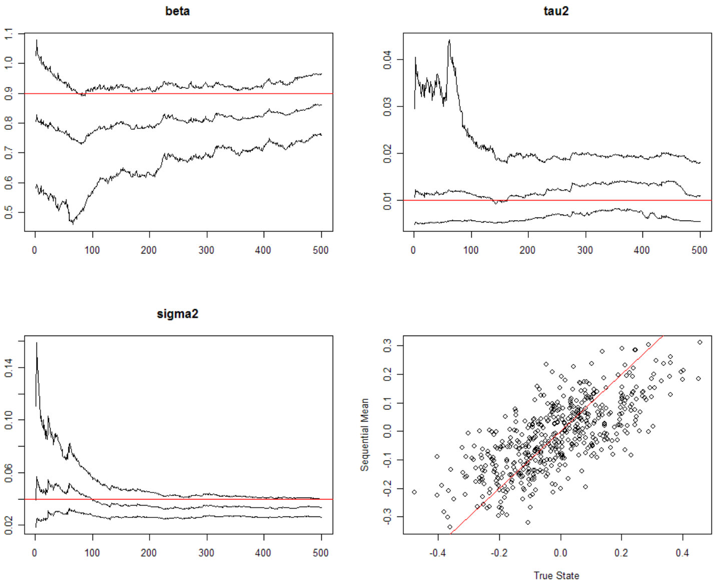
where

$$\lambda_{t+1} \sim \mathcal{IG}(\nu/2, \nu/2)$$

and ϵ_{t+1} and u_{t+1} are N(0,1) and ν is known.

The observation error term is non-normal $\sqrt{\lambda_{t+1}}\epsilon_{t+1} \sim t_\nu$.

Sequential inference



Example: Dynamic multinomial logit model

Let us study the multinomial logit model

$$P\left(y_{t+1}=1|\beta_{t+1}\right)=\frac{e^{F_t\beta_t}}{1+e^{F_t\beta_t}} \text{ and } \beta_{t+1} \quad =\phi\beta_t+\sigma_x\epsilon^{\beta}_{t+1}$$

where $\beta_0 \sim N(0,\sigma^2/(1-\rho^2))$. Scott's (2007) data augmentation structure leads to a mixture Kalman filter model

$$\begin{aligned} y_{t+1} &= \mathbb{I}(z_t \geq 0) \\ z_{t+1} &= Z_t\beta+\epsilon_{t+1} \text{ where } \epsilon_{t+1} \sim -ln\mathcal{E}(1) \end{aligned}$$

Here ϵ_t is an extreme value distribution of type 1 where $\mathcal{E}(1)$ is an exponential of mean one. The key is that it is easily to simulate $p(z_t|\beta,y_t)$ using

$$z_{t+1}=-\ln\left(\frac{\ln U_i}{1+e^{\beta_i\beta}}-\frac{\ln V_i}{e^{\beta_i\beta}}\mathcal{I}_{y_{t+1}=0}\right)$$

10-component mixture of normals

Frunwirth-Schnatter and Schnatter (2007) uses a 10-component mixture of normals:

$$p(\epsilon_t) = e^{-\epsilon_t - e^{-\epsilon_t}} \approx \sum_{j=1}^{10} w_j \mathcal{N}(\mu_j, s_j^2)$$

Hence conditional on an indicator λ_t we can analyze

$$y_t = \mathbb{I}(z_t \geq 0) \quad \text{and} \quad z_t = \mu_{\lambda_t} + Z_t \beta + s_{\lambda_t} \epsilon_t$$

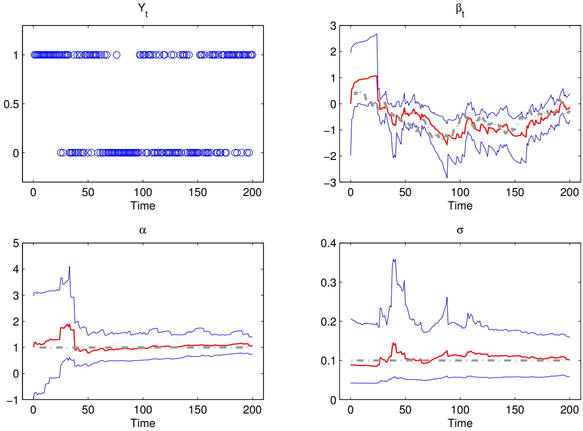
where $\epsilon_t \sim N(0,1)$ and $Pr(\lambda_t = j) = w_j$. Also,

$$\begin{aligned} s_{t+1}^{\beta} &= \mathcal{K}\left(s_t^{\beta}, z_{t+1}, \lambda_{t+1}, \theta, y_{t+1}\right) \\ p(y_{t+1}|s_t^{\beta}, \theta) &= \sum_{\lambda_{t+1}} p(y_{t+1}|s_t^{\beta}, \lambda_{t+1}, \theta) \end{aligned}$$

for re-sampling. Propagation now requires

$$\begin{aligned} \lambda_{t+1} &\sim p\left(\lambda_{t+1}|(s_t^{\beta}, \theta)^{k(i)}, y_{t+1}\right) \\ z_{t+1} &\sim p\left(z_{t+1}|(s_t^{\beta}, \theta)^{k(i)}, \lambda_{t+1}, y_{t+1}\right) \\ \beta_{t+1} &\sim p\left(z_{t+1}|(s_t^{\beta}, \theta)^{k(i)}, \lambda_{t+1}, z_{t+1}\right) \end{aligned}$$

Simulated exercise



PL based on 30,000 particles.

Example: Sequential Bayesian Lasso

We develop a sequential version of Bayesian Lasso[6] for a simple problem of signal detection. The model takes the form

$$\begin{aligned}(y_t|\theta_t) &\sim N(\theta_t, 1)\\ p(\theta_t|\tau) &= (2\tau)^{-1}\exp\left(-|\theta_t|/\tau\right)\end{aligned}$$

for $t = 1, \ldots, n$ and $\tau^2 \sim IG(a_0, b_0)$.

Data augmentation: It is easy to see that

$$p(\theta_t|\tau) = \int p(\theta_t|\tau, \lambda_t)p(\lambda_t)d\lambda_t$$

where

$$\begin{aligned}\lambda_t &\sim Exp(2)\\ \theta_t|\tau, \lambda_t &\sim N(0, \tau^2\lambda_t)\end{aligned}$$

[6] Hans (2008)

Data augmentation

The natural set of latent variables is given by the augmentation variable λ_{n+1} and conditional sufficient statistics leading to

$$Z_n = (\lambda_{n+1}, a_n, b_n)$$

The sequence of variables λ_{n+1} are i.i.d. and so can be propagated directly with $p(\lambda_{n+1})$.

The conditional sufficient statistics (a_{n+1}, b_{n+1}) are deterministically determined based on parameters $(\theta_{n+1}, \lambda_{n+1})$ and previous values (a_n, b_n).

PL algorithm

1. After n observations: $\{(Z_n, \tau)^{(i)}\}_{i=1}^{N}$.
2. Draw $\lambda_{n+1}^{(i)} \sim Exp(2)$.
3. Resample old particles with weights

$$w_{n+1}^{(i)} \propto p(y_{n+1}; 0, 1+\tau^{2(i)}\lambda_{n+1}^{(i)}).$$

4. Sample $\theta_{n+1}^{(i)} \sim N(m_n^{(i)}, C_n^{(i)})$, where $m_n^{(i)} = C_n^{(i)} y_{n+1}$ and $C_n^{-1} = 1 + \tilde{\tau}^{-2(i)}\tilde{\lambda}_{n+1}^{-1(i)}$.
5. Suff. stats: $a_{n+1}^{(i)} = \tilde{a}_n^{(i)} + 1/2$, $b_{n+1}^{(i)} = \tilde{b}_n^{(i)} + \theta_{n+1}^{2(i)}/(2\tilde{\lambda}_{n+1}^{(i)})$.
6. Sample (offline) $\tau^{2(i)} \sim IG(a_{n+1}, b_{n+1})$.
7. Let $Z_{n+1}^{(i)} = (\lambda_{n+1}^{(i)}, a_{n+1}^{(i)}, b_{n+1}^{(i)})$.
8. After $n+1$ observations: $\{(Z_{n+1}, \tau)^{(i)}\}_{i=1}^{N}$.

Sequential Bayes factor

As the Lasso is a model for sparsity we would expect the evidence for it to increase when we observe $y_t = 0$.

We can sequentially estimate $p(y_{n+1} \mid y^n, \text{lasso})$ via

$$p(y_{n+1} \mid y^n, \text{lasso}) = \frac{1}{N} \sum_{i=1}^{N} p(y_{n+1} \mid (\lambda_n, \tau)^{(i)})$$

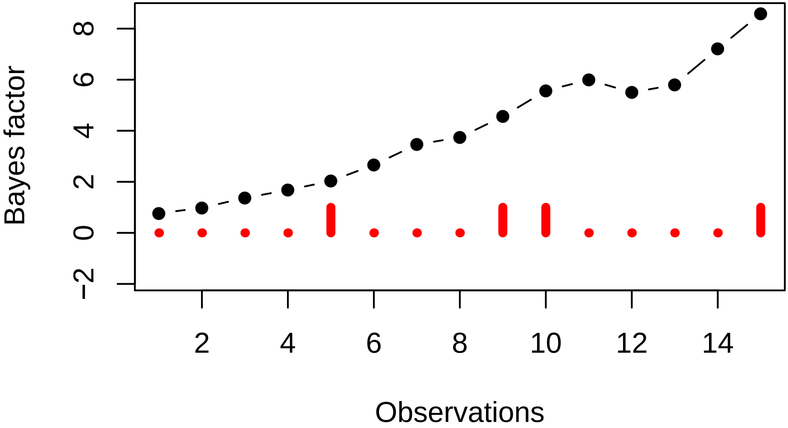
with predictive $p(y_{n+1} \mid \lambda_n, \tau) \sim N(0, \tau^2 \lambda_n + 1)$.

This leads to a sequential Bayes factor

$$BF_{n+1} = \frac{p(y^{n+1} \mid \text{lasso})}{p(y^{n+1} \mid \text{normal})}.$$

Simulated data

Data based on $\theta = (0,0,0,0,1,0,0,0,1,1,0,0,0,0,1)$ and priors $\tau^2 \sim IG(2,1)$ for the double exponential case and $\tau^2 \sim IG(2,3)$ for the normal case, reflecting the ratio of variances between those two distributions.



PL for finite mixture models

1. **Resample**: Generate an index $\zeta \sim \mathrm{MN}(\boldsymbol{\omega}, N)$ where

$$\omega(i) = \frac{\mathrm{p}\left(y_{t+1} \mid (\mathbf{s}_t, \mathbf{n}_t)^{(i)}\right)}{\sum_{i=1}^{N} \mathrm{p}\left(y_{t+1} \mid (\mathbf{s}_t, \mathbf{n}_t)^{(i)}\right)}$$

2. **Propagate**:

$$\begin{aligned} k_{t+1} &\sim \mathrm{p}\left(k_{t+1} \mid (\mathbf{s}_t, \mathbf{n}_t)^{\zeta(i)}, y_{t+1}\right) \\ \mathbf{s}_{t+1} &= \mathcal{S}\left(\mathbf{s}_t^{\zeta(i)}, k_{t+1}, y_{t+1}\right) \\ n_{t+1,k_{t+1}} &= n_{t,k_{t+1}}^{\zeta(i)} + 1, \quad n_{t+1,j} = n_{t,j}^{\zeta(i)} \quad \text{for} \quad j \neq k_{t+1} \end{aligned}$$

3. **Learn**:

$$\mathrm{p}(\mathbf{p}, \boldsymbol{\theta}^\star \mid y^t) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{p}\left(\mathbf{p}, \boldsymbol{\theta}^\star \mid (\mathbf{s}_t, \mathbf{n}_t)^{(i)}\right)$$

Finite mixture of Poisson

Model: an m component mixture of Poisson densities

$$\mathrm{p}(y_t) = \sum_{i=1}^{m} p_j \mathrm{Po}(y_t; \theta_j^\star).$$

Prior:

$$\begin{aligned} \pi(\theta_j^\star) &= \mathrm{ga}(\alpha_j, \beta_j) \qquad \text{for } j = 1, \ldots, m \\ \pi(\mathbf{p}) &\sim \mathrm{Dir}(\boldsymbol{\gamma}). \end{aligned}$$

The form of the conditional posterior given y^t, given the latent allocation k^t, is completely defined by $\mathbf{n}_t$, the number of samples in each component, and sufficient statistics $\mathbf{s}_t = (s_{t,1}, \ldots, s_{t,m})$, where $s_{t,j} = \sum_{r=1}^{t} y_r \mathbb{1}_{[k_r = j]}$.

Resample-propagate

Resample:

$$\begin{aligned}
\mathrm{p}(y_{t+1} \mid \mathbf{s}_t, \mathbf{n}_t) &= \sum_{k_{t+1}=j=1}^{m} \int\int p_j \mathrm{p}(y_{t+1} \mid \theta_j^\star) p(\boldsymbol{\theta}^\star, \mathbf{p}) d(\boldsymbol{\theta}^\star, \mathbf{p}) \\
&= \sum_{k_{t+1}=j=1}^{m} \frac{\Gamma\left(s_{t,j} + y_{t+1} + \alpha_j\right)}{\Gamma\left(s_{t,j} + \alpha_j\right)} \frac{\left(\beta_j + n_{t,j}\right)^{s_{t,j}+\alpha_j}}{\left(\beta_j + n_{t,j} + 1\right)^{s_{t,j}+y_{t+1}+\alpha_j}} \frac{1}{y_{t+1}!} \left(\frac{\gamma_j + n_{t,j}}{\sum_{i=1}^m \gamma_i + n_{t,i}}\right).
\end{aligned}$$

Propagate:

$$p(k_{t+1} = j \mid \mathbf{s}_t, \mathbf{n}_t, y_{t+1}) \propto \frac{\Gamma\left(s_{t,j} + y_{t+1} + \alpha_j\right)}{\Gamma\left(s_{t,j} + \alpha_j\right)} \frac{\left(\beta_j + n_{t,j}\right)^{s_{t,j}+\alpha_j}}{\left(\beta_j + n_{t,j} + 1\right)^{s_{t,j}+y_{t+1}+\alpha_j}} \left(\frac{\gamma_j + n_{t,j}}{\sum_{i=1}^m \gamma_j + n_{t,i}}\right).$$

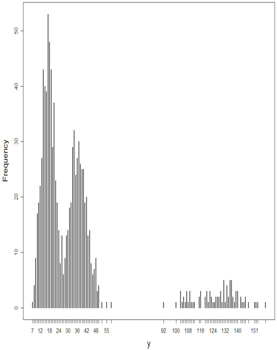
Given k_{t+1},

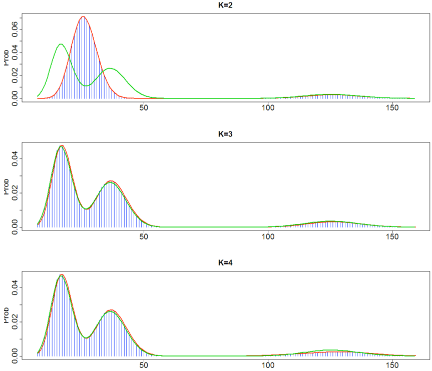
$$\begin{aligned}
s_{t+1,j} &= s_{t,j} + y_{t+1}\mathbb{1}_{\{k_{t+1}=j\}} \\
n_{t+1,j} &= n_{t,j} + \mathbb{1}_{\{k_{t+1}=j\}}
\end{aligned}$$

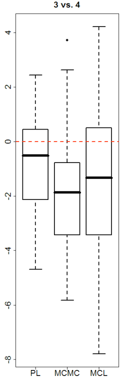
for $j = 1, \ldots, m$.

PL and MCMC

Left: data from a $m = 4$ mixture of Poisson.
Central: PL (red), MCMC (Blue), TRUE (green).
Right: MC study of BF(m=3;m=4).







Dirichlet Process (DP) mixtures

The DP mixtures is the most commonly used nonparametric prior for random mixture models.

Constructive definition: a random distribution G generated from $\mathrm{DP}(\alpha, G_0(\psi))$ is almost surely of the form

$$dG(\cdot) = \sum_{l=1}^{\infty} p_l \, \delta_{\vartheta_l}(\cdot) \tag{1}$$

with $\vartheta_l \overset{iid}{\sim} G_0(\vartheta_l; \psi)$, $p_l = (1 - \sum_{j=1}^{l-1} p_j) v_l$ and $v_l \overset{iid}{\sim} \mathrm{beta}(1, \alpha)$, for $l = 1, 2, \ldots$, centering distribution $G_0(\vartheta; \psi)$ and independent sequences $\{\vartheta_l\}_{l=1}^{\infty}$ and $\{v_k\}_{k=1}^{\infty}$.

The discreteness of DP realizations is explicit in this definition.

PL for DP mixture models

1. **Resample**: Generate an index $\zeta \sim \mathrm{MN}(\boldsymbol{\omega}, N)$ where

$$\omega(i) = \frac{\mathrm{p}\left(y_{t+1} \mid (\mathbf{s}_t, \mathbf{n}_t, m_t)^{(i)}\right)}{\sum_{i=1}^{N} \mathrm{p}\left(y_{t+1} \mid (\mathbf{s}_t, \mathbf{n}_t, m_t)^{(i)}\right)}$$

2. **Propagate**:
   - $k_{t+1} \sim \mathrm{p}\left(k_{t+1} \mid (\mathbf{s}_t, \mathbf{n}_t, m_t)^{\zeta(i)}, y_{t+1}\right)$,
   - For $j \neq k_{t+1}$, $n_{t+1,j} = n_{t,j}$.
   - If $k_{t+1} \leq m_t$, $n_{t+1,k_t} = n_{t,k_t} + 1$ and $m_{t+1} = m_t$. Otherwise, $m_{t+1} = m_t + 1$ and $n_{t,m_{t+1}} = 1$.

3. **Estimation**:

$$\mathrm{p}\left(\mathbb{E}[f(y; G)] \mid y^t\right) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{p}\left(y \mid (\mathbf{s}_t, \mathbf{n}_t, m_t)^{(i)}\right)$$

DP mixture of multivariate normals

The d-dimensional DP multivariate normal mixture (DP-MVN) model has density function

$$f(y_t; G) = \int \mathrm{N}(y_t|\mu_t, \Sigma_t) dG(\mu_t, \Sigma_t)$$

and

$$G \sim DP(\alpha, G_0(\mu, \Sigma)),$$

with conjugate centering distribution

$$G_0 = N(\mu; \lambda, \Sigma/\kappa) W(\Sigma^{-1}; \nu, \Omega),$$

where $W(\Sigma^{-1}; \nu, \Omega)$ denotes a Wishart distribution such that

$$\mathbb{E}[\Sigma^{-1}] = \nu\Omega^{-1} \quad \text{and} \quad \mathbb{E}[\Sigma] = (\nu - (d+1)/2)^{-1}\Omega.$$

Conditional sufficient statistics

For each unique mixture component (the $s_{t,j}$) are

$$\bar{y}_{t,j} = \sum_{r:k_r=j} y_r / n_{t,j}$$

and

$$S_{t,j} = \sum_{r:k_r=j} (y_r - \bar{y}_{t,j})(y_r - \bar{y}_{t,j})' = \sum_{r:k_r=j} y_r y_r' - n_{t,j}\bar{y}_{t,j}\bar{y}_{t,j}'.$$

The initial sufficient statistics are $n_1 = 1$ and $s_1 = \{y_1, 0\}$, such that the algorithm is populated with N identical particles.

Conditional on existing particles $\{(\mathbf{n}_t, \mathbf{s}_t)^i\}_{i=1}^N$, uncertainty is updated through the familiar resample/propagate approach.

Resample

The predictive probability function for resampling is

$$\begin{aligned} \mathrm{p}(y_{t+1}|\mathbf{s}_t, \mathbf{n}_t, m_t+1) &= \frac{\alpha}{\alpha+t}\mathrm{St}(y_{t+1}; a_0, B_0, c_0) \\ &+ \sum_{j=1}^{m_t} \frac{n_{t,j}}{\alpha+t}\mathrm{St}\left(y_{t+1}; a_{t,j}, B_{t,j}, c_{t,j}\right) \end{aligned}$$

where the Student's t distributions are parametrized by $a_0 = \lambda$, $B_0 = \frac{2(\kappa+1)}{\kappa c_0}\Omega$, $c_0 = 2\nu - d + 1$, $a_{t,j} = \frac{\kappa\lambda + n_{t,j}\bar{y}_{t,j}}{\kappa + n_{t,j}}$, $B_{t,j} = \frac{2(\kappa+n_{t,j}+1)}{(\kappa+n_{t,j})c_{t,j}}\left[\Omega + \frac{1}{2}D_{t,j}\right]$, $c_{t,j} = 2\nu + n_{t,j} - d + 1$, and $D_{t,j} = S_{t,j} + \frac{\kappa n_{t,j}}{(\kappa+n_{t,j})}(\lambda - \bar{y}_{t,j})(\lambda - \bar{y}_{t,j})'$.

Propagate

Sample the component state k_{t+1} such that,

$$
\begin{aligned}
\mathrm{p}(k_{t+1} = j) &\propto \frac{n_{t,j}}{\alpha + t} \mathrm{St}(y_{t+1};\ a_{t,j}, B_{t,j}, c_{t,j})\ j = 1, \ldots, m_t \\
\mathrm{p}(k_{t+1} = m_t + 1) &\propto \frac{\alpha}{\alpha + t} \mathrm{St}(y_{t+1};\ a_0, B_0, c_0).
\end{aligned}
$$

If $k_{t+1} = m_t + 1$, the new sufficient statistics are defined by $m_{t+1} = m_t + 1$ and $s_{t+1,m_{t+1}} = [y_{t+1}, 0]$.

If $k_{t+1} = j$, $n_{t+1,j} = n_{t,j} + 1$ and we update $s_{t+1,j}$ such that
$\bar{y}_{t+1} = (n_{t,j}\bar{y}_{t,j} + y_{t+1})/n_{t+1,j}$ and
$S_{t+1,j} = S_{t,j} + y_{t+1}y'_{t+1} + n_{t,j}\bar{y}_{t,j}\bar{y}^{i\prime}_{t,j} - n_{t+1,j}\bar{y}_{t+1,j}\bar{y}^{i\prime}_{t+1,j}$.

Parameter update

Assuming a $\mathrm{W}(\gamma_\Omega, \Psi_\Omega^{-1})$ prior for Ω and a $\mathrm{N}(\gamma_\lambda, \Psi_\lambda)$ prior for λ, the sample at time t is augmented with draws for the auxiliary variables $\{\mu_j^\star, \Sigma_j^\star\}$, for $j = 1, \ldots, m_t$, from their posterior full conditionals,

$$\mathrm{p}(\mu_j^\star, \Sigma_j^\star \mid \mathbf{s}_t, \mathbf{n}_t) = \mathrm{N}(\mu_j^\star;\ a_{t,j}, \frac{1}{\kappa + n_{t,j}} \Sigma_j^\star) \mathrm{W}(\Sigma_j^{\star -1};\ \nu + n_{t,j}, \Omega + D_{t,j}).$$

The parameter updates are then

$$\begin{aligned} \lambda &\sim \mathrm{N}\left(R(\gamma_\lambda \Psi_\lambda^{-1} + \kappa \sum_{j=1}^{m_t} \Sigma_j^{\star -1} \mu_j^\star),\ R \right) \\ \Omega &\sim \mathrm{W}\left(\gamma_\Omega + m_t \nu, R^{-1} \right), \end{aligned}$$

where $R^{-1} = \sum_{j=1}^{m_t} \Sigma_j^{\star -1} + \Psi_\Omega^{-1}$.

Similarly, if α is assigned the usual gamma hyperprior, it can be updated for each particle using the auxiliary variable method from Escobar and West (1995).

Illustration

Four datasets: $d = 2, 5, 10$ and $d = 25$.

Sample size: $t = 1, \dots, T = 500d$.

The d-dimensional y_t was generated from a $\mathrm{N}(\mu_t, \mathrm{AR}(0.9))$ density, where AR(0.9) denotes the correlation matrix implied by an autoregressive process of lag one and correlation 0.9.

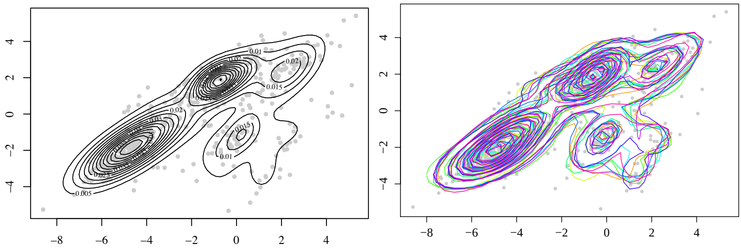
$\mu_t \overset{ind}{\sim} G_\mu$, where G_μ is the realization of a $\mathrm{DP}(4, N(0, 4\mathrm{I}))$ process.

Thus the simulated data is clustered around a set of distinct means, and highly correlated within each cluster.

Dimension $d = 2$

Data and density estimates for PL fit with 1000 particles (left) and each of ten PL fits with 500 particles (right).

A random ordering of the 1000 observations.

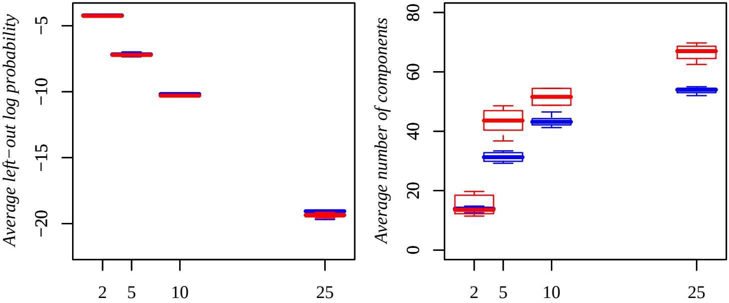


PL and Gibbs sampler

Left: average log posterior predictive score for validation sets of 100 observations.

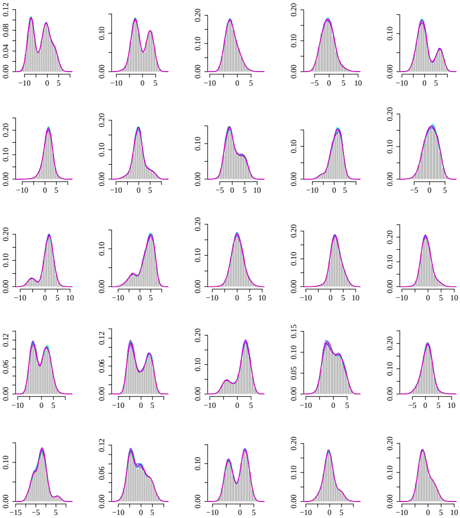
Right: posterior averages for m_T, the total number of allocated mixture components.

Boxplots of the distribution over ten repetitions of the algorithm. Red boxplots correspond to PL, and the blue correspond to Gibbs.



Dimension $d = 25$

Data and marginal density estimates. Curves are posterior means of ten PL fits, 500 particles, random ordering of the data.



Complete class of mixture models

Mixture models:

$$\begin{aligned} \text{likelihood} &: \mathrm{p}(y_{t+1}|k_{t+1},\theta) \\ \text{transition equation} &: \mathrm{p}(k_{t+1}|k^t,\theta) \\ \text{parameter prior} &: \mathrm{p}(\theta). \end{aligned}$$

with k_t states refer to a latent allocation of observations to mixture components, and $k^t = \{k_1, \ldots, k_t\}$.

State-space representation:

$$y_{t+1} = \mathrm{f}(k_{t+1},\theta) \quad (2)$$

$$k_{t+1} = \mathrm{g}(k^t,\theta) \quad (3)$$

where (2) is the observation equation and (3) is the evolution for states k_{t+1}.

General class of hidden Markov models

This structure establishes a direct link to the general class of hidden Markov models, which encompasses a vast number of widely used models.

Density estimation: $y_t \sim \int \mathrm{k}(y_t;\theta)dG(\theta)$, the allocation states break the mixture such that

$$y_t \sim \mathrm{k}(y;\theta_{k_t})$$

and

$$\theta_{k_t} \sim G(\theta).$$

Latent feature models: Multivariate k_t allows allocation of a single observation to multiple mixture components.

Essential state vector

In order to describe the PL algorithm, we begin by defining

$\mathcal{Z}_t$ as an essential state vector

that will be tracked in time.

We also assume that

$\mathcal{Z}_t$ is sufficient for sequential inference

that is, it allows for the computation of:

(a) Posterior predictive $\mathrm{p}(y_{t+1}|\mathcal{Z}_t)$,

(b) Posterior updating rule $\mathrm{p}(\mathcal{Z}_{t+1}|\mathcal{Z}_t, y_{t+1})$,

(c) Parameter learning via $\mathrm{p}(\theta|\mathcal{Z}_{t+1})$.

Particle learning (PL)

The posterior density $\mathrm{p}(\mathcal{Z}_t|y^t)$ is approximated by the equally weighted particle set

$$\{\mathcal{Z}_t^{(i)}\}_{i=1}^N = \{\mathcal{Z}_t^{(1)}, \ldots, \mathcal{Z}_t^{(N)}\}.$$

Then, given $\{\mathcal{Z}_t^{(i)}\}_{i=1}^N$, the generic particle learning update for a new observation y_{t+1} proceeds in two steps:

Step 1: Resample

$$\mathcal{Z}_t^{(i)} \propto \mathrm{p}(y_{t+1}|\mathcal{Z}_t^{(i)})$$

Step 2: Propagate

$$\mathcal{Z}_{t+1}^{(i)} \sim \mathrm{p}(\mathcal{Z}_{t+1}|\mathcal{Z}_t^{(i)}, y_{t+1})$$

Bayes' theorem

This process can be understood by re-writing Bayes' theorem as

$$\begin{aligned} \mathrm{p}(\mathcal{Z}_t|y^{t+1}) &\propto \mathrm{p}(y_{t+1}|\mathcal{Z}_t)\mathrm{p}(\mathcal{Z}_t|y^t) & (4) \\ \mathrm{p}(\mathcal{Z}_{t+1}|y^{t+1}) &= \int \mathrm{p}(\mathcal{Z}_{t+1}|\mathcal{Z}_t, y_{t+1})d\mathrm{P}(\mathcal{Z}_t|y^{t+1}), & (5) \end{aligned}$$

where $\mathrm{P}(\cdot)$ refers to the appropriate measure.

After resampling the initial particles with weights proportional to $\mathrm{p}(y_{t+1}|\mathcal{Z}_t)$ we have particles from $\mathrm{p}(\mathcal{Z}_t|y^{t+1})$.

These particles are then propagated through $\mathrm{p}(\mathcal{Z}_{t+1}|\mathcal{Z}_t, y_{t+1})$, leading to particles $\{\mathcal{Z}_{t+1}^{(i)}\}_{i=1}^N$ approximating $\mathrm{p}(\mathcal{Z}_{t+1}|y^{t+1})$.

PL algorithm for general mixture models

For $t = 1, \dots, T$

Resample: Draw indexes $\zeta^{(1)}, \dots, \zeta^{(N)} \sim \mathrm{Multinomial}(\boldsymbol{\omega}_t, N)$, with unnormalized weights given by

$$\boldsymbol{\omega}_t = \{\mathrm{p}(y_{t+1}|\mathcal{Z}_t^{(1)}), \dots, \mathrm{p}(y_{t+1}|\mathcal{Z}_t^{(N)})\},$$

Propagate:

$$\mathcal{Z}_{t+1}^{(j)} \sim \mathrm{p}(\mathcal{Z}_{t+1}|\mathcal{Z}_t^{\zeta^{(j)}}, y_{t+1}) \qquad j = 1, \dots, N.$$

Learning θ:

$$\theta^{(j)} \sim \mathrm{p}(\theta|\mathcal{Z}_{t+1}^{(j)}) \qquad j = 1, \dots, N.$$

Nature of the essential state vector $\mathcal{Z}_t$

In general, $\mathcal{Z}_t$ does not need to include k^t in order to be sufficient for the distributions listed in (a) to (c).

This, along with moving the propagation step to the end, is what makes PL distinct from the present state of the art in particle filtering for mixtures.

However, it is straightforward to obtain smoothed samples of k^t from the full posterior, through an adaptation of the particle smoothing algorithm of Godsill, Doucet and West (2004).

Allocation

PL provides a vehicle for drawing from the full posterior distribution of the allocation vector, $\mathrm{p}(k^t|y^t)$, through the backwards uncertainty update equation

$$\begin{aligned} p(k^t|y^t) &= \int \mathrm{p}(k^t|\mathcal{Z}_t, y^t) d\mathrm{P}(\mathcal{Z}_t|y^t) \\ &= \int \prod_{r=1}^{t} \mathrm{p}(k_r|\mathcal{Z}_t, y_r) d\mathrm{P}(\mathcal{Z}_t|y^t). \end{aligned}$$

We can directly approximate $\mathrm{p}(k^t|y^t)$ by sampling, for each particle $\mathcal{Z}_t^{(i)}$ and for $r = t, \ldots, 1$, k_r with probability

$$\mathrm{p}(k_r = j|\mathcal{Z}_t, y_r) \propto \mathrm{p}(y_r|k_r = j, \mathcal{Z}_t)\mathrm{p}(k_r = j|\mathcal{Z}_t)$$

leading to an $\mathcal{O}(N)$ algorithm for full posterior allocation.

Marginal likelihood

Marginal likelihoods are of key importance in Bayesian model assessment, particularly when computing Bayes factors.

MCMC: It is known that marginal likelihoods are hard to compute via MCMC schemes, mainly because most approximations are based on one of the following identities (or extensions)

$$p(y^n) = \int p(y^n|\theta)p(\theta)d\theta = \frac{p(y^n|\theta)p(\theta)}{p(\theta|y^n)}$$

SMC: Particle learning, on the other hand, can directly approximate the product rule of probabilities, i.e.

$$\mathrm{p}^N(y^n) = \prod_{t=1}^{n} \mathrm{p}^N(y_t|y^{t-1})$$

with

$$\mathrm{p}^N(y_t|y^{t-1}) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{p}(y_t|\mathcal{Z}_{t-1}^{(i)}). \qquad (6)$$

Density Estimation

We consider density functions of the form,

$$f(y; G) = \int \mathrm{k}(y; \theta) dG(\theta).$$

There are many possibilities for the prior on G:

Finite mixture models: finite dimensional models.
Dirichlet process: stick-breaking (Ferguson, 1973).
Beta two-parameter processes (Ishawaran and Zarepour, 2000).
Kernel stick-breaking processes (Dunson and Park, 2008).

See Walker, Damien, Laud and Smith (1999) or Müller and Quintana (2004) for more complete overviews of the major modeling frameworks.

Collapsed state-space model

An informal formulation of the collapsed state-space model is

$$\begin{aligned} \mathbb{E}\left[f(y_{t+1}; G) \mid \mathcal{Z}_t\right] &= \int \mathrm{k}(y_{t+1}; \theta) d\mathbb{E}\left[G(\theta)\right] \qquad (7) \\ \mathbb{E}\left[dG(\theta) \mid \mathcal{Z}_t\right] &= \int dG(\theta) d\mathrm{P}\left(dG(\theta) \mid \mathcal{Z}_t\right). \qquad (8) \end{aligned}$$

With t observations allocated to m_t mixture components

$$\begin{aligned} \mathbb{E}\left[dG(\theta) \mid \mathcal{Z}_t\right] &= p_0 dG_0(\theta) + \sum_{j=1}^{m_t} p_j \mathbb{E}\left[\delta_{\theta_j^\star} | \mathcal{Z}_t\right] \\ \mathrm{p}\left(y_{t+1} \mid \mathcal{Z}_t\right) &= p_0 \int \mathrm{k}(y_t; \theta) dG_0(\theta) + \sum_{j=1}^{m_t} p_j \int \mathrm{k}(y_t; \theta_j^\star) d\mathrm{P}(\theta_j^\star | \mathcal{Z}_t) \end{aligned}$$

with $\theta_j^\star$ the parameters for each of the m_t components.

Inference about the random mixing distribution

All of the inference so far is based on the marginal posterior predictive, thus avoiding direct simulation of the infinite dimensional random mixing distribution.

In some situations, however, it is necessary to obtain inference about the actual posterior for the random density $f(y; G)$, and hence about G itself, rather than about $\mathbb{E}[f(y; G)]$.

For example, functionals of the conditional density $f(x, y; G)/f(x; G)$ are the objects of inference in implied conditional regression (e.g., Taddy and Kottas, 2009).

Truncation

The standard approach to sampling G is to apply a truncated version of the constructive definition to draw from

$$DP\left(\alpha + t, G_0^t(\theta; \mathbf{n}_t, \boldsymbol{\theta}_t^\star)\right),$$

the conjugate posterior for G given $\boldsymbol{\theta}_t^\star$ and $\mathbf{n}_t$ (Gelfand and Kottas, 2002).

The approximate posterior draw $G_L \equiv \{p_l, \vartheta_l\}_{l=1}^L$ is built from i.i.d. point mass locations $\vartheta_l \sim G_0^t(\vartheta_l; \mathbf{n}_t, \boldsymbol{\theta}_t^\star)$ and the probability vector $\mathbf{p} = (p_1, \ldots, p_L)$ from the finite stick- breaking process $p_l = v_l(1 - \sum_{j=1}^{l-1})$ for $l = 1, \ldots, L$, with $v_l \sim \mathrm{beta}(1, \alpha + t)$ and $v_L = 1$.

Illustration

Data was simulated from

$$y_t \sim N(0.3 + 0.4x_t + 0.5\sin(2.7x_t) + 1.1(1 + x_t^2)^{-1}, \sigma_t^2)$$

where $x_t \sim \mathrm{N}(0,1)$ and

$$\sigma_t = \begin{cases} 0.5 & w.p.\ \Phi(x_t) \\ 0.25 & w.p.\ 1 - \Phi(x_t) \end{cases}$$

The joint distribution of x and y is modeled as arising from DP-MVN with parameter learning and $\alpha = 2$, $\nu = 3$, $\kappa = 0.1$, $\gamma_\lambda = 0$, $\Psi_\lambda = 1.5\mathrm{I}$, $\gamma_\Omega = 3$, and $\Psi_\Omega = 0.1\mathrm{I}$.

Conditional densities

After applying PL with $N = 1000$ particles to filter the posterior, truncated approximations G_L with $L = 300$ were drawn (Taddy and Kottas, 2009).

In particular, the conditional density is available at any location (x, y) as

$$\frac{\sum_{l=1}^{L} p_l \mathrm{N}\left(x, y; \mu_l, \Sigma_l\right)}{\sum_{l=1}^{L} p_l \mathrm{N}(x; \mu_l^x, \sigma_l^x)},$$

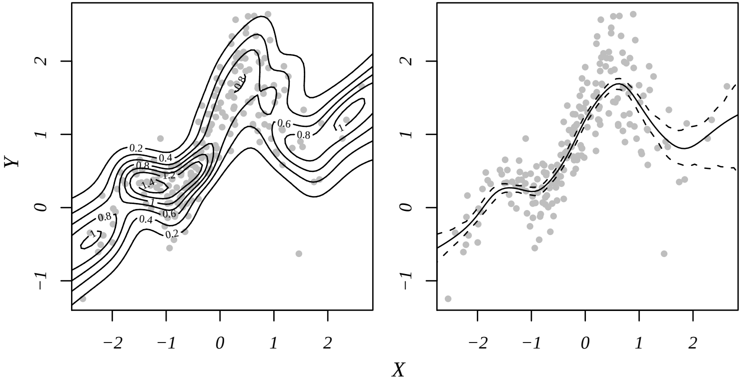
and the conditional mean at x is

$$\mathbb{E}\left[Y \mid x; G_L\right] = \frac{\sum_{l=1}^{L} p_l \mathrm{N}(x; \mu_l^x, \sigma_l^x)\left[\mu_l^y + \rho_l^{xy}(\sigma_l^x)^{-1}(x - \mu_l^x)\right]}{\sum_{l=1}^{L} p_l \mathrm{N}(x; \mu_l^x, \sigma_l^x)}.$$

Regression

Left: filtered posterior mean estimate for the conditional density $f(x, y; G_L)/f(x; G_L)$,

Right: posterior mean and 90% interval for the mean $\mathbb{E}[y|x; G_L]$.



Conclusion

We have proposed a new estimation method for general mixture models.

The approach is easy to understand, simple to implement, and computationally fast.

A vast body of empirical and theoretical evidence of the robust behavior of the resample/propagate PL procedure in states space models appear in Carvalho, Johannes, Lopes and Polson (2009).

Conditioning on sufficient statistics for states and parameters whenever possible creates a Rao-Blackwellized filter with more uniformly distributed resampling weights.

PL does not attempt to approximate the ever increasing joint posterior distribution for k^t.

It is self evident that any importance sampling approximation to the entire vector of allocations will eventually fail, due to the curse of dimensionality, as t grows.

But we show that this is an irrelevant target, since the allocation problem can be effectively solved *after* filtering relevant sufficient information.

Finally, we include an efficient framework for marginal likelihood estimation, providing a valuable tool for real-time sequential model selection.

The framework is especially appealing in the large class of nonparametric mixture priors where the predictive probability function is either available analytically or possible to approximate.

To enable understanding, we have focused on a limited set of concrete models, while pointing to a more general applicability available with little change to the algorithm.

It is thus hoped that this article will facilitate a wider adoption of sequential particle methods in nonparametric mixture model applications.