Estimate Exponential Forgetting rate in HMM

Felix Ye
(joint work with Y.-A. Ma and H. Qian)

University of Washington

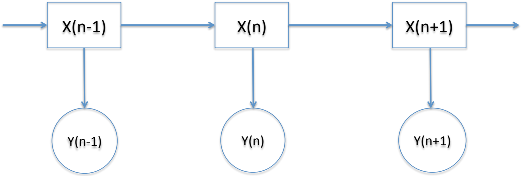
March 29, 2018

[1]

[1] Thanks to Anthony, Alex, Cecilia and U of H summer school

Hidden Markov Model(HMM) are a class of discrete time stochastic processes consisting of

- Latent state sequence $\{X_n\}$ in the finite set with K states follows an irreducible and aperiodic Markov Chain M. (assume Markov Chain reaches stationary)
- The pdf for observation sequences $\{Y_n\}$ in $\mathbb{R}^d$ is $p(y|x_n = i, \phi_i) = b_i(y)$. It could be a Gaussian or your favorite distribution. ϕ is emission parameter.
- The parameter of an HMM is $\theta = (M, \phi)$.



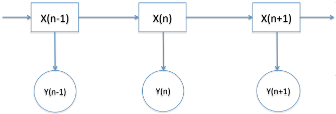
Say I have a very long observation sequence $y_{1:n}$ (for instance, $n = 10^8$), what is the 'suitable' or 'best' parameter θ (M and ϕ) for this HMM?
One could use log-likelihood or log posterior function to find local maximum(EM algorithm) or conduct MCMC method

$$\ln p(\theta|y_{1:n}) \propto \ln p(y_{1:n}|\theta) + \ln p(\theta) \tag{1}$$

$$= \ln(\sum_{i=1}^{K} p(x_n = i, y_{1:n}|\theta)) + \ln p(\theta) \tag{2}$$

where $p(\theta)$ is prior function. BUT, calculate the gradient of $\ln p(\theta|y_{1:n})$ is computationally expensive.
Scalability issue.

Forward Algorithm (filtering): essential part in inference



Y_n is conditionally independent of everything but X_n.
X_n is conditionally independent of everything but X_{n-1}.

Fixed M and emission parameter ϕ, by Bayes's rule,

$$p(x_n, y_{1:n}|\theta) = \sum_{x_{n-1}} \underbrace{p(y_n|x_n, x_{n-1}, y_{1:n-1})}_{=p(y_n|x_n)} \underbrace{p(x_n|x_{n-1}, y_{1:n-1})}_{=p(x_n|x_{n-1})} p(x_{n-1}, y_{1:n-1}|\theta)$$

Rewrite into matrix form, $\mathbf{p}_n = [p(x_n = 1, y_{1:n}|\theta), \ldots, p(x_n = K, y_{1:n}|\theta)]$

$$\mathbf{p}_n = \mathbf{p}_{n-1} M D_n$$

where $D_n = \text{diag}(b_i(y_n))$, $b_i(y_n) = P(y_n|x_n = i)$

Forward Process (filtering)

The forward algorithm gives

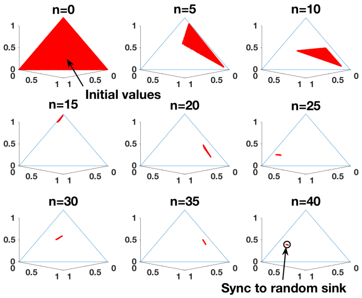
$$\mathbf{p}_n = P(x_n, y_{1:n}|\theta) = \mathbf{p}_0 \underbrace{MD_1}_{A_1} \dots \underbrace{MD_n}_{A_n} \tag{3}$$

One could think A_i are random matrices sampled in i.i.d manner with pdf $p(y) = \sum_i \pi_i p(y|x = i, \phi_i)$, where π is invariant density.
The filtered state probability $\boldsymbol{\rho}_n = P(x_n|y_{1:n}, \theta) = \mathbf{p}_n/(\mathbf{p}_n \cdot \mathbb{1})$.
Now $\boldsymbol{\rho}$ is in the projective space, a simplex S^{K-1}. However, $\mathbf{p}$ is in $\mathbb{R}^K$.

Exponential Forgetting

For a fixed sequence $\omega = (A_1, A_2, A_3, \dots)$, and two different initial conditions $\mathbf{p}_0$, $\mathbf{p}'_0$, two sequences of filtered state probability $\boldsymbol{\rho}_n$ and $\boldsymbol{\rho}'_n$ will synchronize almost surely (with some conditions),

$$\|\boldsymbol{\rho}_n - \boldsymbol{\rho}'_n\| \to_{n\to+\infty} 0 \tag{4}$$



exponential forgetting rate

$$\lim_{n \to +\infty} \frac{1}{n} \log \|\boldsymbol{\rho}_n - \boldsymbol{\rho}'_n\| = ? \tag{5}$$

Atar and Zeitouni (1997), Collet and Leonardi (2014) showed (5) is the gap of the second and first Lyapunov exponents almost surely, $\lambda_2 - \lambda_1$.
Moreover, if the transition matrix M is primitive and emission distribution is positive, then the gap is strictly negative.
But the analytical estimation is practically not useful. We need an efficient algorithm to estimate the gap

Numerical Calculation: Soft-Max parametrization

If $\boldsymbol{\rho} = [a_1, a_2, \ldots, a_K]$, define projection Π, $S^{K-1} \to \mathbb{R}^{K-1}$

$$\Pi : \boldsymbol{\rho} = \left[a_1, a_2, \ldots, a_K\right] \to \mathbf{r} = \left[\underbrace{\log(\frac{a_1}{a_K})}_{r_1}, \underbrace{\log(\frac{a_2}{a_K})}_{r_2}, \ldots, \underbrace{\log(\frac{a_{K-1}}{a_K})}_{r_{K-1}}, 0\right]$$

It is a 1-to-1 mapping for interior points. The derivative of the map and its inverse exist. So it preserves the Lyapunov exponent.

Consider first $K-1$ components and it removes the top Lyapunov exponent $\lambda_1' = 0$.
The dynamics for $\boldsymbol{\rho}$ in S^{K-1} induces a **random map** for $\mathbf{r}$ in $\mathbb{R}^{K-1}$.

$$\mathbf{r}_n = \underbrace{\mathbf{d}_n}_{\text{random translation}} + \underbrace{F(\mathbf{r}_{n-1})}_{\text{deterministic map}} \tag{6}$$

$$\mathbf{d}_n = \left[\ln \frac{b_1(y_n)}{b_K(y_n)}, \ldots, \ln \frac{b_{K-1}(y_n)}{b_K(y_n)}\right] \tag{7}$$

$$F_i(\mathbf{r}) = \ln\left(\frac{\exp(\mathbf{r}) \cdot \mathbf{m}_i}{\exp(\mathbf{r}) \cdot \mathbf{m}_K}\right),\ 1 \le i \le K-1 \tag{8}$$

$\mathbf{m}_i$ is i-th column of M.
It naturally defines an induced i.i.d random dynamical system (RDS) in $\mathbb{R}^{K-1}$.
The Jacobian $J(\mathbf{r}) = \nabla F(\mathbf{r})$ doesn't dependent on $\mathbf{d}$. We proved the following theorem,

Theorem

$$\lambda_2 - \lambda_1 = \lambda_{max} = \limsup_{n \to +\infty} \frac{1}{n} \log \|J(\mathbf{r}_{n-1}) \ldots J(\mathbf{r}_0)\|$$

$$\lambda_{\text{max}} = \limsup_{n \to +\infty} \frac{1}{n} \log \|J(\mathbf{r}_{n-1}) \dots J(\mathbf{r}_0)\|$$

- Calculating finite time maximum Lyapunov exponent for $\mathbf{r}$ is faster than QR.
- Only the information of $\mathbf{p}_n$ or $\boldsymbol{\rho}_n$ is needed. The observation sequence $y_{1:n}$ is not used directly.

The distance of two sequences with different initial conditions,

$$\|\boldsymbol{\rho}_n - \boldsymbol{\rho}_n'\| \leq C \exp((\lambda_2 - \lambda_1)n) \|\mathbf{p}_0 - \mathbf{p}_0'\|$$

For given an error ϵ, the memory length L needed is roughly about

$$L \approx \frac{\log(\epsilon)}{(\lambda_2 - \lambda_1)} \tag{9}$$

How to help?

Remind the posterior function and its gradient are

$$\ln p(\theta|y_{1:n}) \propto \ln p(y_{1:n}|\theta) + \ln p(\theta)$$

$$\frac{\partial \ln p(\theta|y_{1:n})}{\partial \theta_i} = \sum_{j=1}^{n} \frac{\mathbf{p}_0 M D_1 \dots \frac{\partial (M D_j)}{\partial \theta_i} \dots M D_n \mathbb{1}}{\mathbf{p}_0 M D_1 \dots M D_j \dots M D_n \mathbb{1}} + \frac{\partial \ln p(\theta)}{\partial \theta_i} \tag{10}$$

If $n = 10^8$, the number of matrix product required here is $2 * 10^8$ and the space need to store is $2 * 10^8$. But we don't need to use the full data, just pick small portion of them smartly!

Mini-batch gradient descent

Step 1: Uniformly sample a subset of S with cardinality s,

$$\frac{\partial \ln p(\theta|y_{1:n})}{\partial\theta_i} \approx \frac{n}{s}\sum_{j\in S}\frac{\mathbf{p}_0 MD_1 \ldots \frac{\partial(MD_j)}{\partial\theta_i}\ldots MD_n\mathbb{1}}{\mathbf{p}_0 MD_1\ldots MD_j\ldots MD_n\mathbb{1}} + \frac{\partial \ln p(\theta)}{\partial\theta_i} \tag{11}$$

Step 2: Use subsequences to approximate

$$\frac{n}{s}\sum_{j\in S}\frac{\mathbf{p}_0 MD_1\cdots MD_{j-1}\frac{\partial(MD_j)}{\partial\theta_i}MD_{j+1}\cdots MD_n\mathbb{1}}{\mathbf{p}_0 MD_1\cdots MD_{j-1}MD_jMD_{j+1}\cdots MD_n\mathbb{1}} \tag{12}$$

$$\approx\frac{n}{s}\sum_{j\in S}\frac{\mathbf{p}_0 \overbrace{MD_{j-L}\cdots MD_{j-1}}^{L_1=L}\frac{\partial(MD_j)}{\partial\theta_i}\overbrace{MD_{j+1}\cdots MD_{j+L}}^{L_1=L}\mathbb{1}}{\mathbf{p}_0 MD_{j-L}\cdots MD_{j-1}MD_jMD_{j+1}\cdots MD_{j+L}\mathbb{1}}$$

1. For given controlled error ϵ, what is the memory length $L(\theta)$?

   $L = \log(\epsilon)/(\lambda_2 - \lambda_1)$ where λ is Lyapunov exponent.

2. $L_1 = L_2$?

   Yes

3. Can I reuse L for nearby parameter θ? (The continuity of L with respect to θ)

   Yes. This is very delicate. proved by Avila in 2015

Summary

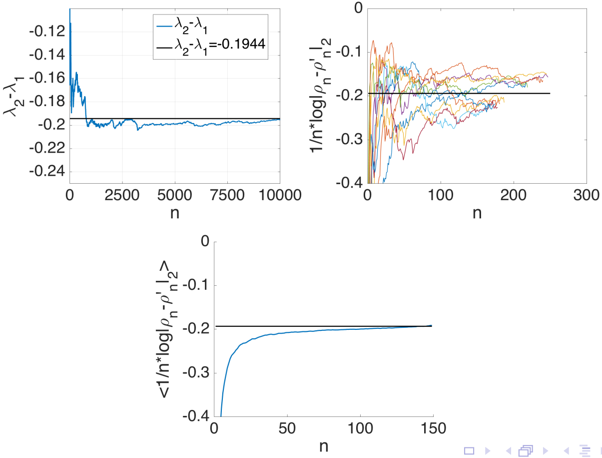
$$\text{Inference} \Leftarrow \text{HMM} \underset{\text{gap of LE}}{\overset{\text{random matrix production}}{\rightleftharpoons}} \text{RDS} \Leftrightarrow \text{MET}$$

Ye, F. X.-F., Ma, Y.-A., and Qian, H. Estimate exponential memory decay in Hidden Markov Model and its applications. Preprint: https://arxiv.org/pdf/1710.06078.pdf

Ye, F. X.-F., Wang, Y. and Qian, H, *Stochastic dynamics: Markov chains and random transformations*, DCDS-B, 21, 7, 2337-2361, 2016

Now I can answer the question for the rate of exponential forgetting

$$M = \begin{bmatrix} 0.005 & 0.99 & 0.005 \\ 0.01 & 0.03 & 0.96 \\ 0.95 & 0.005 & 0.045 \end{bmatrix}, \mu = [0, 0.5, -0.5], \sigma = [1, 1, 1]$$



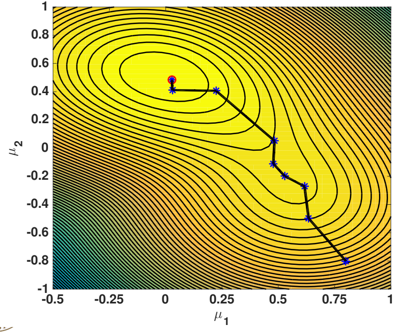
$$P(y_{LM}) = MD_{j-L}\cdots MD_{j-1},\ P(y_{RM}) = MD_{j+1}\cdots MD_{j+L}$$

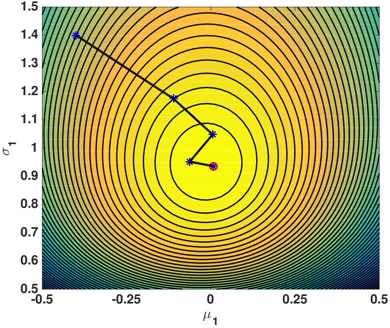
$$\frac{n}{s}\sum_{j\in S}\frac{\mathbf{p}_0 P(y_{LM})\frac{\partial(MD_j)}{\partial\theta_i}P(y_{RM})\mathbb{1}}{\mathbf{p}_0 P(y_{LM})MD_j P(y_{RM})\mathbb{1}} + \frac{\partial \ln p(\theta)}{\partial\theta_i}$$

In the last example, $L = 67$ for $\epsilon = 10^{-10}$ and I use $s = 1000$. The total matrix production needed is $2Ls = 1.34 * 10^5 \ll 2 * 10^8$. The space needed is $s = 1000 \ll 2 * 10^8$.
We only use less than 1% data!

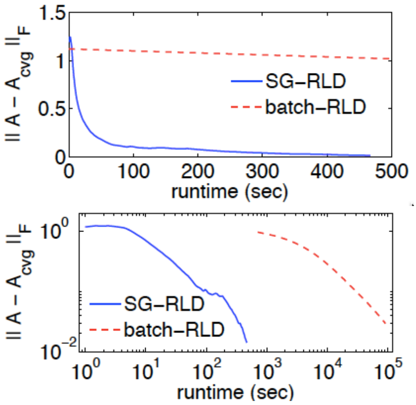
Finding local maximum: Mini-batch gradient descent

Fix other parameters except μ_1 and μ_2 (or μ_1 and σ_1). Use the mini-batch gradient descent, (much faster than EM algorithm)





SG-MCMC: 1000 times faster!



Published in ICML 2017, https://arxiv.org/pdf/1706.04632.pdf

Extension

Now we can efficiently estimate the gradient of the likelihood.

1. It is possible to estimate the Hessian efficiently as well. So we can calculate the observed fisher information (Hessian of MLE).
2. Second order method to find stationary points is possible.
3. Extend to finite state Kalman filter.