1 코드 주석

1.1 makeYX.m

사실 수학적인 모델만 파악되면 코드가 하는 건 그다지 어렵지 않다. 왜냐하면 베이지언 모델은 사전분포와 가능도 함수만 구할 수 있으면 MCMC를 짜서 돌리면 사후분포로부터 샘플을 얻을 수 있기 때문이다. 물론 빠르게 target distribution로 수렴하는 MCMC를 고안하는 것은 여전히 연구대상이다.
먼저 `makeYX.m`의 코드를 뜯어보면 다음의 모델을 고려했다는 것을 알 수 있다.

$$
\begin{aligned}
y_t' &= y_{t-1}'\Phi_1' + \cdots + y_{t-p}'\Phi_p + e_t' && (1)\\
&= \underbrace{\begin{bmatrix} y_{t-1}' & y_{t-2}' & \cdots & y_{t-p}' \end{bmatrix}}_{x_t'} \underbrace{\begin{bmatrix} \Phi_1' \\ \Phi_2' \\ \vdots \\ \Phi_p' \end{bmatrix}}_{\beta} + e_t' && (2)\\
&= x_t'\beta + e_t', \quad e_t \sim \mathcal{N}_k\left(\mathbf{0}, \Omega\right) && (3)
\end{aligned}
$$

- $\Phi_t: \ k \times k$
- $y_t, e_t: \ k \times 1$
- $\beta: \ pk \times k$
- $x_t': \ 1 \times pk$

그리고 모든 가능한 $t = (p+1), \ldots, T$에 해당하는 방정식들을 한번에 다음과 같이 나타낼 수 있다.

$$
\underbrace{\begin{bmatrix} y_{p+1}' \\ \vdots \\ y_T' \end{bmatrix}}_{Y_0} = \underbrace{\begin{bmatrix} X_{p+1}' \\ \vdots \\ x_t' \end{bmatrix}}_{YL} \beta + \begin{bmatrix} e_{p+1}' \\ \vdots \\ e_T' \end{bmatrix} \tag{4}
$$

코드에서도 이와 마찬가지로 `Y0`와 `YL`을 만들어낸다.
위 모델을 열벡터로 다시 바꾸면 $y_t = \beta' x_t + e_t$인데 여기서 $\beta' x_t$는 어차피 $k \times 1$ 벡터이므로 $\mathrm{vec}(\beta' x_t)$로 써도 무방하다. 그리고 다음 관계를 이용하려 한다.

$$
\mathrm{vec}(A_1 A_2 A_3) = \left(A_1' \otimes A_3\right) \mathrm{vec}\left(A_2\right) \tag{5}
$$

그러므로 우리는 $\text{vec}\,(I_k \beta' x_t)$에 대해서

$$\text{vec}\left(I_k \beta' x_t\right) = \left(I_k \otimes x_t\right) \text{vec}\left(\beta'\right) \tag{6}$$

로 나타낼 수 있다. 코드에서는 모든 t에 대해 $I_k \otimes x_t$를 해서 3차원 array에 저장한 게 `YLm`이다.

1.2 `Gen_Phi.m`, `Gen_Omega.m`

데이터셋이 만들어졌으니 이제 해야 하는 건 모수를 추정하는 일이다. 사실 업데이트하는 방법은 책에 나와있기는 하지만 가능도함수를 계산하는 것이 빠져있다. 원래대로 하자면

$$L\left(\{\Phi_t\}_{t=1}^{p}, \Omega \mid D\right) = \prod_{t=1}^{p} p\left(y_t\right) \prod_{t=p+1}^{T} p\left(y_t\,;\, \beta' x_t, \Omega\right) \tag{7}$$

이지만 1부터 p까지의 개별적인 밀도함수는 알 수 없으므로 penalized likelihood를 쓴다.

$$\begin{aligned} L\left(\{\Phi_t\}_{t=1}^{p}, \Omega \mid D\right) &= \prod_{t=p+1}^{T} p\left(y_t\,;\, \beta' x_t, \Omega\right) && (8) \\ &= |\Omega|^{-(T-p-1)/2} \exp\left[-\frac{1}{2} \sum_{t=p+1}^{T} \left(y_t - \beta' x_t\right)' \Omega^{-1} \left(y_t - \beta' x_t\right)\right] && (9) \end{aligned}$$

이제 이걸 바탕으로 추정을 하면 되는데 마지막으로 첨언을 하자면 코드에서 `ehat*ehat'` 이런 식으로 벡터의 텐서곱을 하는 부분이 있는데 이것은 저 가능도 함수에서

$$\left(y_t - \beta' x_t\right)' \Omega^{-1} \left(y_t - \beta' x_t\right) = \text{tr}\left(\Omega \left(y_t - \beta' x_t\right) \left(y_t - \beta' x_t\right)'\right) \tag{10}$$

이기 때문에 나온 부분이다. 나머지는 이전에 Gibbs sampler 구하듯이 똑같이 하면 된다.