UNIVERSITY OF TEXAS ARLINGTON

Bayesian Hierarchical Dynamic Factor Models

Anthony M. Thomas, Jr.
anthony.thomas3@mavs.uta.edu

Department of Mathematics
The University of Texas at Arlington

September 8, 2020

Contents

Section 1

Bayesian Inference

Bayesian Inference can be described by two parts:

1. Build a model based on data $\boldsymbol{X}$ and parameters $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2\}$
   - Likelihood: $p(\boldsymbol{X}|\boldsymbol{\Theta})$
   - Prior: $p(\boldsymbol{\Theta})$
2. Compute the posterior
   - Posterior:

$$p(\boldsymbol{\Theta}|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\boldsymbol{\Theta})p(\boldsymbol{\Theta})}{p(\boldsymbol{X})}$$

   - Report summaries, e.g. posterior expectations

$$\mathbb{E}[h(\boldsymbol{\Theta})|\boldsymbol{X}]$$

     or marginal posterior expectations

$$\mathbb{E}[h(\boldsymbol{\Theta}_i)|\boldsymbol{X}]$$

Factor Analysis

- **Factor Analysis** (FA) is a method that assumes that the covariance structure of a set of cross-sectional observations can be described in terms of a linear combination of latent variables called factors
- A sample of P observations are related to a set of factors through the equation

$$\boldsymbol{X}_i = \boldsymbol{\Lambda}\boldsymbol{F}_i + \boldsymbol{e}_i, \qquad i = 1, \ldots, P \tag{1}$$

where

  - $\boldsymbol{X}_i = (X_{1i}, \ldots, X_{Ni})^\top$ denotes a vector of observations for variable i
  - $\boldsymbol{F}_i = (F_{1i}, \ldots, F_{Ki})^\top$ denotes a vector of factors for variable i
  - $\boldsymbol{e}_i = (e_{1i}, \ldots, e_{Ni})^\top$ denotes a vector of measurement errors and idiosyncratic (unique) factors for variable i
  - $\boldsymbol{\Lambda} = [\lambda_{nk}]_{N \times K}$ denotes a matrix of factor loadings

- The following assumptions are typically made in FA:
  1. $\text{rank}(\boldsymbol{\Lambda}) = K$
  2. $\mathbb{E}[\boldsymbol{X}_i] = \mathbb{E}[\boldsymbol{e}_i] = \mathbf{0}_N$ and $\mathbb{E}[\boldsymbol{F}_i] = \mathbf{0}_K \quad \forall i$
  3. $\text{Var}(\boldsymbol{F}_i) = \mathbf{I}_K$ and $\text{Var}(\boldsymbol{e}_i) = \boldsymbol{\Sigma}$ where $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_N^2) \quad \forall i$
  4. $\text{Cov}(\boldsymbol{F}_i, \boldsymbol{e}_i) = \mathbf{0}_{K \times N} \quad \forall i$
- Under these assumptions it follows that

$$\text{Var}(\boldsymbol{X}_i) = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Sigma} \quad \forall i$$

- Typical uses of FA:
  1. Dimension reduction: explain the covariation between N variables using $K < N$ factors
  2. Data interpretation: find factors that explain the covariation
  3. Theory testing: test whether a hypothesized factor structure fits observed data

Time Series Factor Analysis

- In [1] FA is extended to time series data as **time series factor analysis** (TSFA)
- A sample of T time series observations are related to the factors through the equation

$$\boldsymbol{X}_t = \boldsymbol{\Lambda}\boldsymbol{F}_t + \boldsymbol{e}_t \qquad t = 1, \ldots, T \tag{2}$$

where

- $\boldsymbol{X}_t = (X_{1t}, \ldots, X_{Nt})^\top$ denotes a vector of observations at time t
- $\boldsymbol{F}_t = (F_{1t}, \ldots, F_{K_F t})^\top$ denotes a vector of factors at time t
- $\boldsymbol{e}_t$ is a vector of measurement errors and idiosyncratic factors at time t
- $\boldsymbol{\Lambda} = [\lambda_{nk}]_{N \times K}$ denotes a matrix of factor loadings

Dynamic Factor Analysis

- In **dynamic factor analysis** (DFA) the factors are assumed to not only affect the observations contemporaneously, but affect them through their lags as well:

$$X_{nt} = \lambda^n(L)\boldsymbol{F}_t + \boldsymbol{e}_t \quad n = 1, \ldots, N \tag{3}$$

where

$$\lambda^n(L) = \lambda_0^n + \lambda_1^n L + \cdots + \lambda_q^n L^q$$

$$L^s \boldsymbol{F}_t = \boldsymbol{F}_{t-s} \quad \forall s \geq 0$$

is a distributed lag polynomial of factor loadings in the lag operator L for the nth series

Dynamic Factor Analysis

- In DFA the factors are modeled as a time series process
- The time series process is commonly taken to be a vector autoregressive process, i.e.

$$\boldsymbol{\Psi}(L)\boldsymbol{F}_t = \boldsymbol{\varepsilon}_t \tag{4}$$

where

$$\boldsymbol{\Psi}(L) = \mathbf{I}_K - \boldsymbol{\Psi}_1 L - \cdots - \boldsymbol{\Psi}_p L^p$$

is a matrix polynomial of autocorrelation coefficients in the lag operator L

Section 2

Hierarchical Dynamic Factor Analysis

- Suppose you organize a large panel of data into B blocks (e.g. Production, Employment, Demand, etc.) and each block b has N_b series
- $N = \sum_{b=1}^{B} N_b$
- A block may be divided into subblocks (e.g. Demand: Retail Sales, Auto Sales)

Hierarchical Dynamic Factor Model

- In [2], the authors generalize the dynamic factor model by positing that for each t, the nth series in a given block b, denoted by X_{bnt}, has three sources of variation:
  1. idiosyncratic
  2. block-specific
  3. common

Hierarchical Dynamic Factor Model

- A three-level representation of the data for $b = 1, \ldots, B$ and $n = 1, \ldots, N_b$ is given as

$$X_{bnt} = \boldsymbol{\lambda}^{n}_{G.b}(L)\boldsymbol{G}_{bt} + e_{Xbnt} \tag{5}$$
$$\boldsymbol{G}_{bt} = \boldsymbol{\Lambda}_{F.b}(L)\boldsymbol{F}_t + \boldsymbol{e}_{Gbt} \tag{6}$$
$$\boldsymbol{\Psi}_F(L)\boldsymbol{F}_t = \boldsymbol{\epsilon}_{Ft}, \tag{7}$$

  where
  - $\boldsymbol{\lambda}^{n}_{G.b}(L)$ denotes a distributed lag polynomial of block-level factor loadings
  - $\boldsymbol{\Lambda}_{F.b}(L)$ denotes a distributed lag matrix polynomial of common factor loadings
  - $\boldsymbol{G}_{bt} = (G_{b1t}, \ldots, G_{bK_{Gb}t})^{\top}$ denotes the block-level factors
  - $\boldsymbol{F}_t = (F_{1t}, \ldots, F_{K_F t})^{\top}$ denotes the common factors

Hierarchical Dynamic Factor Model

- For some blocks, it may be appropriate to break up the data into subblocks, which adds another source of variation
- Let Z_{bsnt} be the nth series in subblock s of block b
- A four-level representation of the subblock data is given as

$$Z_{bsnt} = \boldsymbol{\lambda}^{n}_{H.bs}(L)\boldsymbol{H}_{bst} + e_{Zbsnt} \tag{8}$$
$$\boldsymbol{H}_{bst} = \boldsymbol{\Lambda}_{G.bs}(L)\boldsymbol{G}_{bt} + \boldsymbol{e}_{Hbst} \tag{9}$$
$$\boldsymbol{G}_{bt} = \boldsymbol{\Lambda}_{F.b}(L)\boldsymbol{F}_{t} + \boldsymbol{e}_{Gbt} \tag{10}$$
$$\boldsymbol{\Psi}_{F}(L)\boldsymbol{F}_{t} = \boldsymbol{\epsilon}_{Ft} \tag{11}$$

where

  - $\boldsymbol{\lambda}^{n}_{H.bs}(L)$ denotes a distributed lag polynomial of subblock-level factor loadings
  - $\boldsymbol{\Lambda}_{G.bs}(L)$ denotes a distributed lag matrix polynomial of block-level factor loadings
  - $\boldsymbol{H}_{bst} = (H_{bs1t}, \ldots, H_{bsK_{Hbst}})^{\top}$ denotes the subblock-level factors

Hierarchical Dynamic Factor Model

- The idiosyncratic components, the subblock-specific, block-specific, and common factors are assumed to be stationary, Gaussian autoregressive processes of orders q_{Zbsn}, q_{Xbn}, q_{Hbsi}, q_{Gbj}, and q_{Fk}, respectively, i.e.

$$
\begin{aligned}
\psi_{Z.bsn}(L)e_{Zbsnt} &= \epsilon_{Zbsnt}, & \epsilon_{Zbsnt} &\sim \mathcal{N}\left(0, \sigma^2_{Zbsn}\right) & n &= 1, \ldots, N_{bs} \\
\psi_{X.bn}(L)e_{Xbnt} &= \epsilon_{Xbnt}, & \epsilon_{Xbnt} &\sim \mathcal{N}\left(0, \sigma^2_{Xbn}\right) & n &= 1, \ldots, N_{b} \\
\psi_{H.bsi}(L)e_{Hbsit} &= \epsilon_{Hbsit}, & \epsilon_{Hbsi} &\sim \mathcal{N}\left(0, \sigma^2_{Hbsi}\right) & i &= 1, \ldots, K_{Hbs} \\
\psi_{G.bj}(L)e_{Gbjt} &= \epsilon_{Gbjt}, & \epsilon_{Gbjt} &\sim \mathcal{N}\left(0, \sigma^2_{Gbj}\right) & j &= 1, \ldots, K_{Gb} \\
\psi_{F.k}(L)F_{kt} &= \epsilon_{Fkt}, & \epsilon_{Fkt} &\sim \mathcal{N}\left(0, \sigma^2_{Fk}\right) & k &= 1, \ldots, K_{F}
\end{aligned}
$$

Hierarchical Dynamic Factor Model

- Not all series need to belong to blocks and subblocks
- In general, the data used in a four-level model are a mixture of Z_{bsnt}, X_{bnt}, and X_{nt}

State Space Representation

- Observed data (vector/matrix form):

$$\begin{aligned} \boldsymbol{Z}_{bst} &= \boldsymbol{\Lambda}_{H.bs}(L)\boldsymbol{H}_{bst} + \boldsymbol{e}_{Zbst} \\ \boldsymbol{\Psi}_{Z.bs}(L)\boldsymbol{e}_{Zbst} &= \boldsymbol{\epsilon}_{Zbst} \end{aligned}$$

implies that the measurement equation is

$$\begin{aligned} \boldsymbol{\Psi}_{Z.bs}(L)\boldsymbol{Z}_{bst} &= \boldsymbol{\Psi}_{Z.bs}(L)\boldsymbol{\Lambda}_{H.bs}(L)\boldsymbol{H}_{bst} + \boldsymbol{\epsilon}_{Zbst} \\ \tilde{\boldsymbol{Z}}_{bst} &= \tilde{\boldsymbol{\Lambda}}_{H.bs}\vec{\boldsymbol{H}}_{bst} + \boldsymbol{\epsilon}_{Zbst}, \quad \boldsymbol{\epsilon}_{Zbst} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{Zbs}) \end{aligned} \tag{12}$$

with transition equation

$$\vec{\boldsymbol{H}}_{bst} = \vec{\boldsymbol{\alpha}}_{G.bst} + \tilde{\boldsymbol{\Psi}}_{Hbs}\vec{\boldsymbol{H}}_{bst-1} + \vec{\boldsymbol{\epsilon}}_{Hbst}, \quad \vec{\boldsymbol{\epsilon}}_{Hbst} \sim \mathcal{N}\big(\boldsymbol{0}, \boldsymbol{1}\boldsymbol{1}^{\top} \otimes \boldsymbol{\Sigma}_{Hbs}\big) \tag{13}$$

where $\boldsymbol{\alpha}_{G.bst} = \boldsymbol{\Psi}_{H.bs}(L)\boldsymbol{\Lambda}_{G.bs}(L)\boldsymbol{G}_{bt}$

State Space Representation

- Subblock-level factors (vector/matrix form):

$$\begin{aligned} \boldsymbol{H}_{bt} &= \boldsymbol{\Lambda}_{G.b}(L)\boldsymbol{G}_{bt} + \boldsymbol{e}_{Hbt} \\ \boldsymbol{\Psi}_{H.b}(L)\boldsymbol{e}_{Hbt} &= \boldsymbol{\epsilon}_{Hbt} \end{aligned}$$

implies that the (pseudo) measurement equation is

$$\begin{aligned} \boldsymbol{\Psi}_{H.b}(L)\boldsymbol{H}_{bt} &= \boldsymbol{\Psi}_{H.b}(L)\boldsymbol{\Lambda}_{G.b}(L)\boldsymbol{G}_{bt} + \boldsymbol{\epsilon}_{Hbt} \\ \tilde{\boldsymbol{H}}_{bt} &= \tilde{\boldsymbol{\Lambda}}_{G.b}\vec{\boldsymbol{G}}_{bt} + \boldsymbol{\epsilon}_{Hbt}, \quad \boldsymbol{\epsilon}_{Hbt} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{Hb}) \end{aligned} \tag{14}$$

with transition equation

$$\vec{\boldsymbol{G}}_{bt} = \boldsymbol{\alpha}_{F.bt} + \tilde{\boldsymbol{\Psi}}_{G.b}\vec{\boldsymbol{G}}_{bt-1} + \vec{\boldsymbol{\epsilon}}_{Gbt}, \quad \vec{\boldsymbol{\epsilon}}_{Gbt} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{1}\boldsymbol{1}^{\top} \otimes \boldsymbol{\Sigma}_{Gb}\right) \tag{15}$$

where $\boldsymbol{\alpha}_{F.bt} = \boldsymbol{\Psi}_{G.b}(L)\boldsymbol{\Lambda}_{F.b}(L)\boldsymbol{F}_t$

State Space Representation

- Block-level factors (vector/matrix form):

$$\begin{aligned} \boldsymbol{G}_t &= \boldsymbol{\Lambda}_F(L)\boldsymbol{F}_t + \boldsymbol{e}_{Gt} \\ \boldsymbol{\Psi}_G(L)\boldsymbol{e}_{Gt} &= \boldsymbol{\epsilon}_{Gt} \end{aligned}$$

implies that the (pseudo) measurement equation is

$$\begin{aligned} \boldsymbol{\Psi}_G(L)\boldsymbol{G}_t &= \boldsymbol{\Psi}_G(L)\boldsymbol{\Lambda}_F(L)\boldsymbol{F}_t + \boldsymbol{\epsilon}_{Gt} \\ \tilde{\boldsymbol{G}}_t &= \tilde{\boldsymbol{\Lambda}}_F\vec{\boldsymbol{F}}_t + \boldsymbol{\epsilon}_{Gt}, \quad \boldsymbol{\epsilon}_{Gt} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_F) \end{aligned} \tag{16}$$

with transition equation

$$\vec{\boldsymbol{F}}_t = \tilde{\boldsymbol{\Psi}}_F\vec{\boldsymbol{F}}_{t-1} + \vec{\boldsymbol{\epsilon}}_{Ft}, \quad \vec{\boldsymbol{\epsilon}}_{Gbt} \sim \mathcal{N}\big(\boldsymbol{0}, \boldsymbol{1}\boldsymbol{1}^\top \otimes \boldsymbol{\Sigma}_{Gb}\big) \tag{17}$$

Data and Model Structure

- The authors organized a dataset consisting of $N = 445$ series giving $T = 227$ observations into 5 blocks

Table 1: Block Structure

Block	Subblock	Source	N	K_{Hbs}
Production	CU	Fed	25	1
	IP	Fed	38	1
	DG	Census	60	2
Employment	ES	BLS	82	2
	HS	BLS	92	1
Consumption	WT	Census	54	1
	RS	Census	30	1
Housing		Census	29	
Manufacturing surveys		ISM, Fed	35	

Data and Model Structure

- The distributed lag (matrix) polynomials of factor loadings are assumed to be constant (order 0), i.e.

$$\begin{aligned}\lambda^{n}_{H.bs}(L) &= \lambda^{n}_{H.bs0}\\ \mathbf{\Lambda}_{G.bs}(L) &= \mathbf{\Lambda}_{G.bs0}\\ \mathbf{\Lambda}_{F.b}(L) &= \mathbf{\Lambda}_{F.b0}\end{aligned}$$

- The (matrix) polynomials of autocorrelation coefficients are assumed to be of order 1, i.e.

$$\begin{aligned}\psi_{Z.bsn}(L) &= 1 - \psi_{Z.bsn1}L\\ \psi_{X.bn}(L) &= 1 - \psi_{X.bn1}L\\ \psi_{H.bsi}(L) &= 1 - \psi_{H.bsi1}L\\ \psi_{G.bj}(L) &= 1 - \psi_{G.bj1}L\\ \psi_{F.k}(L) &= 1 - \psi_{F.k1}L\end{aligned}$$

Data and Model Structure

- They estimate one common factor, i.e. $K_F = 1$, one common factor per block, i.e. $K_{Gb} = 1 \quad \forall b$, and one or two factors per subblock, i.e. $K_{Hbs} = 1, 2$
- For the cases where $K_{Hbs} = 2$, the factor loading matrices are assumed to be lower triangular with 1's along the diagonal, i.e.

$$\mathbf{\Lambda}_{H.bs} = \begin{bmatrix} 1 & 0 \\ \lambda_{H.bs_{2,1}} & 1 \\ \lambda_{H.bs_{3,1}} & \lambda_{H.bs_{3,2}} \\ \vdots & \vdots \\ \lambda_{H.bs_{N_{bs},1}} & \lambda_{H.bs_{N_{bs},2}} \end{bmatrix}$$

Parameter Priors

- All free factor loadings and autocorrelation coefficients are assigned independent standard Gaussian priors, i.e.

$$\lambda_{(\cdot)} \sim \mathcal{N}(0, 1)$$
$$\psi_{(\cdot)} \sim \mathcal{N}(0, 1)$$

- All variance parameters are assigned independent scaled-inverse chi squared distributions with 4 degrees of freedom and a scale of 0.01, i.e.

$$\sigma^2_{(\cdot)} \sim \text{scale-inv-}\chi^2\left(4, 0.01^2\right)$$

Gibbs Sampler

- The authors implement a Gibbs sampling algorithm based on a degenerate linear Gaussian state-space representation of the model to obtain samples from the posterior and compute posterior means
- 50,000 draws are used as a burn-in, i.e. discarded
- 50,000 more draws are obtained while storing every 50th draw to obtain a posterior sample size of 1,000
- The main idea of the algorithm is presented next

Gibbs Sampler

- Let $\mathbf{\Lambda} = (\mathbf{\Lambda}_H, \mathbf{\Lambda}_G, \mathbf{\Lambda}_F)$, $\mathbf{\Psi} = (\mathbf{\Psi}_F, \mathbf{\Psi}_G, \mathbf{\Psi}_H, \mathbf{\Psi}_Z)$, and $\mathbf{\Sigma} = (\mathbf{\Sigma}_F, \mathbf{\Sigma}_G, \mathbf{\Sigma}_H, \mathbf{\Sigma}_Z)$

1. Organize the data into blocks and subblocks to get Z_{bst} for $b = 1, \ldots, B, s = 1, \ldots, B_S$. Get initial values for $\{H_{bst}\}$, $\{G_{bt}\}$, and $\{F_t\}$ using principal components. Use these to produce initial values for $\mathbf{\Lambda}, \mathbf{\Psi}, \mathbf{\Sigma}$
2. Conditional on $\mathbf{\Lambda}, \mathbf{\Psi}, \mathbf{\Sigma}, \{G_{bt}\}, \{F_t\}$ and the data, draw $\{H_{bst}\} \quad \forall b, s$
3. Conditional on $\mathbf{\Lambda}, \mathbf{\Psi}, \mathbf{\Sigma}, \{H_{bst}\}, \{F_t\}$ and the data, draw $\{G_{bt}\} \quad \forall b$
4. Conditional on $\mathbf{\Lambda}, \mathbf{\Psi}, \mathbf{\Sigma}, \{H_{bst}\}, \{G_{bt}\}$ and the data, draw $\{F_t\}$
5. Conditional on $\{H_{bst}\}$, $\{G_{bt}\}, \{F_t\}$ and the data, draw $\mathbf{\Lambda}, \mathbf{\Psi}, \mathbf{\Sigma}$
6. Return to 2

Section 3

Why Variational Bayesian Inference?

- Consider a model with data $\boldsymbol{X}$ and latent variables $\boldsymbol{Z}$ (model parameters can be included)
- The goal is to compute the joint posterior of the latent variables given the data

$$p(\boldsymbol{Z}|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\boldsymbol{Z})p(\boldsymbol{Z})}{p(\boldsymbol{X})} \tag{18}$$

  - Likelihood: $p(\boldsymbol{X}|\boldsymbol{Z})$
  - Prior: $p(\boldsymbol{Z})$
  - Evidence: $p(\boldsymbol{X})$

$$p(\boldsymbol{X}) = \int p(\boldsymbol{X}, \boldsymbol{Z})\, d\boldsymbol{Z} \tag{19}$$

Why Variational Bayesian Inference?

- For complex models (18) and (19) either have no closed-form or require high-dimensional integration which causes the "inference problem"
- As a result, the joint posterior has to be approximated
- Markov chain Monte Carlo (MCMC) methods have been the gold standard to solve this problem
  1. Construct an ergodic Markov chain on $\boldsymbol{Z}$ whose stationary distribution is the joint posterior $p(\boldsymbol{Z}|\boldsymbol{X})$
  2. Sample from the chain to collect samples from the stationary distribution
  3. Approximate the posterior with an empirical estimate constructed from a subset of the collected samples
  4. Use the subset of collected samples to estimate expectations of interest

Why Variational Bayesian Inference?

- MCMC uses sampling to solve the inference problem
- MCMC methods eventually produce accurate results, but are usually computationally intensive for interesting models
- Variational Bayes (VB) uses optimization to solve the inference problem
- VB typical obtains results much faster

Variational Bayesian Inference

- The main idea of the VB framework is to
  1. Posit a family of "nice" approximate densities $\mathcal{Q}$
  2. Find a member of that family that is "closest" to the exact posterior

$$q^\star(\boldsymbol{Z}) = \underset{q(\boldsymbol{Z})\in\mathcal{Q}}{\arg\min}\, \mathrm{KL}(q(\boldsymbol{Z}) \,||\, p(\boldsymbol{Z}|\boldsymbol{X})) \tag{20}$$

where

$$\mathrm{KL}(q(\boldsymbol{Z}) \,||\, p(\boldsymbol{Z}|\boldsymbol{X})) = \mathbb{E}_q[\log q(\boldsymbol{Z})] - \mathbb{E}_q[\log p(\boldsymbol{Z}|\boldsymbol{X})] \tag{21}$$

Variational Bayesian Inference

- It follows that

$$\mathrm{KL}(q(\boldsymbol{Z}) \,||\, p(\boldsymbol{Z}|\boldsymbol{X})) = \mathbb{E}_q[\log q(\boldsymbol{Z})] - \mathbb{E}_q[\log p(\boldsymbol{X}, \boldsymbol{Z})] + \log p(\boldsymbol{X})$$

  which reveals that the objective in (20) depends on the evidence, thus it cannot be computed directly

Evidence Lower Bound

- The reason why $\mathrm{KL}(q||p)$ is a desirable measure of "closeness" is because it leads to a lower bound on $\log p(\boldsymbol{X})$ called the **evidence lower bound** (ELBO)

$$
\begin{aligned}
\log p(\boldsymbol{X}) &= \log \int q(\boldsymbol{Z}) \frac{p(\boldsymbol{X}, \boldsymbol{Z})}{q(\boldsymbol{Z})} \, d\boldsymbol{Z} \\
&= \log \left(\mathbb{E}_q \left[\frac{p(\boldsymbol{X}, \boldsymbol{Z})}{q(\boldsymbol{Z})} \right] \right) \\
&\geq \mathbb{E}_q \left[\log \frac{p(\boldsymbol{X}, \boldsymbol{Z})}{q(\boldsymbol{Z})} \right], \qquad \text{by Jensen's Inequality} \\
&= \mathbb{E}_q[\log p(\boldsymbol{X}, \boldsymbol{Z})] - \mathbb{E}_q[\log q(\boldsymbol{Z})] \\
&= \mathrm{ELBO}(q)
\end{aligned}
$$

Evidence Lower Bound

- It follows that

$$\mathrm{KL}(q(\boldsymbol{Z}) \,||\, p(\boldsymbol{Z}|\boldsymbol{X})) = \log p(\boldsymbol{X}) - \mathrm{ELBO}(q)$$

- Since $\log p(\boldsymbol{X})$ is constant wrt $q(\boldsymbol{Z})$ we can redefine the objective function as

$$\arg\min_{q(\boldsymbol{Z})\in\mathcal{Q}} \mathrm{KL}(q(\boldsymbol{Z}) \,||\, p(\boldsymbol{Z}|\boldsymbol{X})) = \arg\max_{q(\boldsymbol{Z})\in\mathcal{Q}} \mathrm{ELBO}(q)$$

Variational Bayesian Inference

- The VB framework is now
  1. Posit a family of "nice" approximate densities $\mathcal{Q}$
  2. Find a member of that family that is "closest" to the exact posterior, i.e.

$$q^\star(\boldsymbol{Z}) = \underset{q(\boldsymbol{Z})\in\mathcal{Q}}{\arg\max}\, \mathrm{ELBO}(q) \tag{22}$$

- Solving this optimization problem is still difficult in general
- Using the mean-field assumption can make it easier

Mean-Field Assumption

- The mean-field assumption says to:
  1. Partition the latent variables into M groups, say $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_M$
  2. Assume that the distributions in $\mathcal{Q}$ factorize across the groups, i.e.

$$\mathcal{Q} = \left\{ q : q(\boldsymbol{Z}) = \prod_{m=1}^{M} q_m(\boldsymbol{Z}_m) \right\}$$

- Learning the optimal q now reduces to learning the optimal $q_1, \ldots, q_M$
- Straightforward to optimize via coordinate ascent
- This is **NOT** a modeling assumption

Mean-Field Assumption

- Interestingly, under the mean-field assumption, the optimization problem for a single q_m has the solution:

$$q_m(\boldsymbol{Z}_m) = \frac{\exp\{\mathbb{E}_{-m}[\log p(\boldsymbol{X}, \boldsymbol{Z})]\}}{\int \exp\{\mathbb{E}_{-m}[\log p(\boldsymbol{X}, \boldsymbol{Z})]\}\, d\boldsymbol{Z}_m} \quad \forall m \tag{23}$$

- This establishes what is called the **coordinate ascent variational inference** (CAVI) algorithm

Coordinate Ascent Variational Inference

Algorithm 1: Coordinate ascent variational inference

Input: Model $p(\boldsymbol{X}, \boldsymbol{Z})$, Data $\boldsymbol{X}$,
Output: Variational density $q(\boldsymbol{Z}) = \prod_{m=1}^{M} q_m(\boldsymbol{Z}_m)$
Initialize: Variational densities $q_m(\boldsymbol{Z}_m)$
while *the ELBO has not converged* **do**
  for $m \in \{1, 2, \ldots, M\}$ **do**
    Set $q_m(\boldsymbol{Z}_m) \propto \exp\{\mathbb{E}_{-m}[\log p(\boldsymbol{X}, \boldsymbol{Z})]\}$
  end
  Compute ELBO(q)
end
return $q(\boldsymbol{Z})$

Conjugate-Exponential Models

- Conjugate-exponential models satisfy two conditions:
  1. The complete-data likelihood is in the exponential family, i.e.

$$p(\boldsymbol{X}_i, \boldsymbol{Z}_i|\boldsymbol{\theta}) = g(\boldsymbol{\theta}) f(\boldsymbol{X}_i, \boldsymbol{Z}_i) \exp\Big\{\phi(\boldsymbol{\theta})^\top u(\boldsymbol{X}_i, \boldsymbol{Z}_i)\Big\} \tag{24}$$

     where $\phi(\boldsymbol{\theta})$ is the vector of natural parameters and u is a vector of sufficient statistics
  2. The parameter prior is conjugate to the complete-data likelihood, i.e.

$$p(\boldsymbol{\theta}|\eta, \boldsymbol{\nu}) = h(\eta, \boldsymbol{\nu}) g(\boldsymbol{\theta})^\eta \exp\Big\{\phi(\boldsymbol{\theta})^\top \boldsymbol{\nu}\Big\} \tag{25}$$

     where η and $\boldsymbol{\nu}$ are hyperparameters of the prior
- The author in [3] generalizes the expectation-maximization (EM) to a VB-EM algorithm for conjugate-exponential models (HMMs, MFA, and SSMs)

Variational Bayesian EM

- Given an i.i.d. dataset $\boldsymbol{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$ the ELBO(q) can be maximized iteratively

1. The VBE step:

$$q(\boldsymbol{Z}_i) \propto f(\boldsymbol{X}_i, \boldsymbol{Z}_i) \exp\Big\{\bar{\phi}^\top u(\boldsymbol{X}_i, \boldsymbol{Z}_i)\Big\} \quad \forall i \tag{26}$$

with

$$\bar{\phi} = \mathbb{E}_{\boldsymbol{\theta}}[\phi(\boldsymbol{\theta})]$$

2. The VBM step:

$$q(\boldsymbol{\theta}) = h(\tilde{\eta}, \tilde{\boldsymbol{\nu}}) g(\boldsymbol{\theta})^{\tilde{\eta}} \exp\Big[\phi(\boldsymbol{\theta})^\top \tilde{\boldsymbol{\nu}}\Big] \tag{27}$$

with

$$\tilde{\eta} = \eta + N$$

$$\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} + \sum_{i=1}^{N} \bar{u}(\boldsymbol{X}_i)$$

$$\bar{u}(\boldsymbol{X}_i) = \mathbb{E}_{\boldsymbol{Z}_i}[u(\boldsymbol{X}_i, \boldsymbol{Z}_i)]$$

Section 4

Future Work

- Ultimate goal is to develop a VB framework to handle the three- and four-level hierarchical dynamic factor model in [2]
- Developing a VB framework according to CAVI is straightforward, but very tedious
  - Still an open problem
- Developing a VB framework according to the VBEM using the degenerate state space representation is not as straightforward
  - The state transition covariance matrix is singular
  - The state transition equations have time-varying intercepts
  - Still an open problem

Section 5

Selected References I


Paul D Gilbert, Erik Meijer, et al. “Time series factor analysis with an application to measuring money”. In: *University of Groningen, Research School SOM Research Report 05F10* (2005).


Emanuel Moench, Serena Ng, and Simon Potter. “Dynamic Hierarchical Factor Models”. en. In: *Review of Economics and Statistics* 95.5 (Dec. 2013), pp. 1811–1817. ISSN: 0034-6535, 1530-9142. DOI: 10.1162/REST_a_00359.


Matthew James Beal. “Variational Algorithms for Approximate Bayesian Inference”. en. In: (), p. 281.