Bayesian Factor Models

November 29, 2016

Overview

- High-dimensional data ubiuitous in modern applications
- Sample size n smaller than the number of variables p - ‘small n large p problem
- Classical statistical methods break down in such settings
- Exploiting structure is crucial
- Low rank matrix/tensor factorizations for estimating joint dependence among high dimensional continuous/categorical variables

Motivating application - high-dim regression

- $y_i \in \mathbb{R}$ & $x_i = (x_{i1}, \dots, x_{ip})' \in \mathcal{X} \subset \mathbb{R}^p$, $i = 1, \dots, n$
- n = sample size, p=number of predictors & $p \gg n$
- $y_i = x_i^{\mathrm{T}}\beta + \epsilon_i, \quad \epsilon_i \sim \mathrm{N}(0, \sigma^2)$
- In big data problems, dimensionality reduction is crucial
- *sparsity* in β: L_1 & other regularization methods

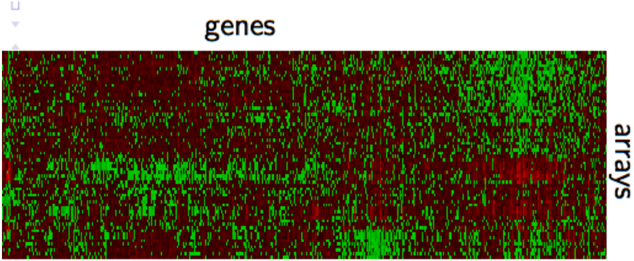


Figure: millions of genes potentially affecting a biomarker

Motivating application - large covariance estimation

- Inference on dependence in $y_i = (y_{i1}, \dots, y_{ip})' \in \mathbb{R}^p$, $i = 1, \dots, n$: estimate $\Omega = \mathrm{cov}(y_i)$
- Regularization approaches for large covariance estimation
- banding/tapering (BL 08, WP 10), thresholding (BL 08, RLZ 09, CL 11), banding/penalizing Cholesky factor (WP 03, RLZ 10), regularized PCA (JL 09, HT 06) and many others
- Many regularization approaches but what about uncertainty?
- Bayesian methods enter naturally

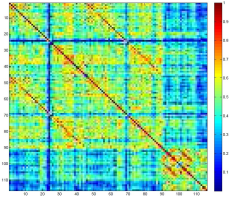


Figure: exploiting structure in estimating covariance

Background on factor models

- Massive dimensional vector of candidate predictors encountered in many application areas.
- Factor models provide a convenient framework for dimension reduction in large p, small n applications (West, 2003; Lucas et al., 2006; Carvalho et al., 2008).
- Explain dependence among high dimensional observations through fewer number of underlying factors.

Factor modeling

- Highly successful approach for dimensionality reduction
- Relate high-dimensional y_i to low-dimensional η_i through

$$y_i = \Lambda\eta_i + \epsilon_i, \quad \epsilon_i \sim \mathrm{N}_p(0, \Omega)$$

- $\Lambda = p \times k$ tall skinny factor loadings matrix
- $\eta_i \sim N_k(0, \mathrm{I}_k)$ are latent factors
- Marginalizing out η_i,

$$y_i \sim N_p(0, \Sigma), \quad \Sigma = \Lambda\Lambda^T + \Omega$$

Sparse factor modeling

- West, 2003 & many others
- Reduce dimensionality in two ways
- The number of latent factors $k \ll p$
- In addition, the loadings matrix Λ relating η_i to y_i has lots of zeros
- This structure is well motivated in many biomedical applications
- For example, $y_i =$ gene expression, $\eta_i =$ pathway expression, & small proportion of genes are in any given pathway
- Connection to sparse PCA (Zou, Hastie & Tibshirani, 2006)

Bayesian factor models - recent developments

- Variable selection-type mixture prior on loadings (Lucas et al., 2006; Carvalho et al., 2008).
- Recent work on latent feature models using the Indian buffet process (Griffiths & Ghahramani, 2006; Thibaux & Jordan, 2007).
- Weighted versions have found applications in factor analysis (Knowles & Ghahramani, 2007; Meeds et al., 2007; Rai & Daumé, 2009).
- Parameter expansion to induce heavy-tailed default prior on the loadings (Ghosh & Dunson, 2009).

Some notations

- Θ_Λ to denote the collection of matrices Λ with p rows and infinitely many columns such that $\Lambda\Lambda^{\mathrm{T}}$ is a $p \times p$ matrix with all entries finite.

$$\Theta_\Lambda = \left\{ \Lambda = (\lambda_{jh}),\ j = 1, \ldots, p,\ \ h = 1 \ldots, \infty,\ \max_{1 \le j \le p} \sum_{h=1}^{\infty} \lambda_{jh}^2 < \infty \right\}$$

- Denote Θ_Σ to be the set of $p \times p$ diagonal matrices with non-negative entries and Θ to be all $p \times p$ positive semi-definite matrices.
- Define $g : \Theta_\Lambda \times \Theta_\Sigma \to \Theta$ where $g(\Lambda, \Sigma) = \Lambda\Lambda^{\mathrm{T}} + \Sigma$.
- Choose independent priors supported on $\Theta_\Lambda \times \Theta_\Sigma$, which induce a prior on $\Omega \in \Theta$ through g.

The MGPS prior (Bhattacharya & Dunson, 2011 (Biometrika)

- Proposed multiplicative gamma process shrinkage (MGPS) prior on the loadings is given by

$$\lambda_{jh} \mid \phi_{jh}, \tau_h \sim N(0, \phi_{jh}^{-1}\tau_h^{-1}), \; \phi_{jh} \sim \mathcal{G}(\nu/2, \nu/2),$$

$$\tau_h = \prod_{l=1}^{h} \delta_l, \; \delta_1 \sim \mathcal{G}(a_1, 1), \; \delta_l \sim \mathcal{G}(a_2, 1), \; l \geq 2,$$

- τ_h is a global shrinkage parameter for the hth column, stochastically increasing under the restriction $a_2 > 1$.
- ϕ_{jh}'s are local shrinkage parameters for the elements in the hth column, avoid over-shrinking the non-zero loadings in later columns.

Choice of the truncation level

- Truncate the loadings matrix to have $k^* << p$ columns. Posterior samples from approximated conditional posterior.
- How to chose an appropriate level of truncation?
- Redundant factors – correspond to columns of loadings whose all elements are less than ϵ in magnitude.
- Effective factors – all non-redundant factors.

A possible approach

- Start with a conservative guess $\tilde{k}$ of k^*.
- At the tth iteration of the Gibbs sampler, define $m^{(t)}$ to be the number of redundant columns in $\Lambda_{\tilde{k}}$, whose all elements are less than ϵ in magnitude($\epsilon = 10^{-4}$ used as a default)
- Usual shrinkage priors on the loadings exhibit the phenomenon of factor splitting.
- Our approach avoids this problem by shrinking increasingly in later columns.
- Define $k^{*(t)} = \tilde{k} - m^{(t)}$ to be the effective number of factors at iteration t.

Adaptive Gibbs sampler

- Adapt the number of factors as the sampler progresses – avoids specifying over-conservative initial guess.
- Designed to satisfy the diminishing adaptation condition of Roberts & Rosenthal (2007). Discard redundant columns if $m^{(t)} > 0$, otherwise add a new column with additional parameters drawn from the prior.
- Let $\tilde{k}^{(t)}$ be the truncation level at the tth iteration and $k^{*(t)} = \tilde{k}^{(t)} - m^{(t)}$ the effective number of factors.
- Estimate k^* by the mode or median of the samples $\{k^{*(t)}\}_{t=B+1}^{N}$.

Covariance matrix estimation

- Set $\Sigma^{(t)} = \Lambda^{(t)}_{\tilde{k}^{(t)}} \Lambda^{(t)\prime}_{\tilde{k}^{(t)}} + \Omega^{(t)}$.
- $\{\Sigma^{(t)}\}_{t=B+1}^{N}$ represent draws from the approximated marginal posterior distribution of Σ given $y_i, i = 1, \ldots, n$.

Regression Coefficient Estimation

- Recall, after marginalizing out latent factors, $y_i \sim N_p(0, \Omega)$ with $\Sigma = \Lambda\Lambda^{\mathrm{T}} + \Omega$.
- $E(z_i \mid x_i) = x_i^{\mathrm{T}}\beta$, with $\beta = \Sigma_{xx}^{-1}\Sigma_{zx}$, true regression coefficients of z on x.
- Set $\beta^{(t)} = \{\Sigma_{xx}^{(t)}\}^{-1}\Sigma_{zx}^{(t)}$, where $\Sigma_{xx}^{(t)} = \Lambda_x^{(t)}\Lambda_x^{(t)\,\mathrm{T}} + \Omega_{xx}^{(t)}$ denote posterior samples at the tth iteration.
- Computation involves inverting $\tilde{k}^{(t)} \times \tilde{k}^{(t)}$ matrices at tth iteration.
- Let $\hat{\beta}$ denote the posterior mean of β. The proposed formulation retains the non-zero elements of β while heavily shrinks the rest toward zero.

Covariance matrix estimation

true (p, k)		$(100, 5)$			$(500, 10)$			$(1000, 15)$	
method	MGPS	Band	MAP	MGPS	Band	MAP	MGPS	Band	MAP
mse									
mean	0·2	1·3	0·2	0·1	0·4	0·1	0·1	0·3	0·1
min	0·1	0·9	0·1	0·02	0·4	0·05	0·02	0·2	0·05
max	0·3	1·6	0·3	0·2	0·5	0·3	0·4	0·5	0·3
aab									
mean	1·9	3·1	1·0	0·6	0·6	0·3	0·4	0·5	0·3
min	1·3	2·5	0·6	0·4	0·6	0·2	0·2	0·4	0·2
max	2·5	4·9	1·5	0·9	0·9	0·5	0·6	0·5	0·5
mab									
mean	50·9	111	44·8	95·4	117·8	97·7	115	115	108
min	38·8	99·8	24·7	50·2	105	64·4	52·6	111	74·7
max	74·1	131	105	152	131	162	242	240	221

Simulation study performance in covariance matrix estimation. The average, best and worst case performance across 50 simulation replicates in terms of mean square error $(\times 10^2)$, average absolute bias $(\times 10^2)$ and maximum absolute bias $(\times 10^2)$ are tabulated for the different methods. MGPS: posterior mean under proposed prior; Band: Banding algorithm of Bickel and Levina, 2008; MAP: approximate MAP estimate of covariance matrix

Time & Memory Constraints

- Theoretical aspects such as convergence rates of the estimators well studied in Bayesian factor models. (PBPD 14)
- Computation of the covariance estimate $\hat{\Sigma} = \hat{\Lambda}\hat{\Lambda}^{\mathrm{T}} + \hat{\Omega}$ can be challenging for high to ultra-high p
- In standard implementations,
  - the $\Lambda_{p \times k}$ needs to be stored
  - the low-rank matrix must be inverted, which requires costs $\mathcal{O}(k^3)$ and $\mathcal{O}(k^2)$ in time and memory respectively per MCMC iteration
  - posterior mean and variance involve matrix multiplications that involve more than $\mathcal{O}(p)$ computations

A Divide-and-Conquer Approach To Covariance Matrix Estimation In The Bayesian Framework

Existing Literature

- Computer Science
  - Distributed Training Strategies for the Structured Perceptron (MHM 10)
  - Divide-and-Conquer matrix factorization (MJT 11)
  - A Divide-and-Conquer procedure for Sparse Inverse Covariance Matrix Estimation (HDRB 12)
- Statistics
  - Bootstrapping big data (KTSJ 12)
  - Divide-and-Conquer kernel ridge regression (ZDW 13)
  - Robust and Scalable Bayes Via A Median of Subset Posterior Measures (MSLD 14)
  - Computational limits of Divide-and-Conquer method (SC 15)

The Divide-and-Conquer Framework

- (D step)- Randomly partition y into g p_g-dimensional subvectors, $\{y^{(1)}, \ldots, y^{(g)}\}$ where $y_i^{(m)} \in \mathbb{R}^{p_g}$, $m = 1, \ldots, g$
- (F step) - Fit a factor model to g parallel subvectors using MCMC to obtain posterior quantities of interest. All posterior quantities are retained in factored form.
- (C step) - The parallel MCMCs generate a final covariance matrix estimate $\hat{\Sigma}$ by combining $[\Lambda^{(1)}, \ldots, \Lambda^{(g)}]$ using the correlation structure induced through the latent factors.

C step: Combine estimates from subgroups

- Parallel MCMCs generate g estimates of the low rank matrix $[\Lambda^{(1)}, \dots, \Lambda^{(m)}, \dots, \Lambda^{(g)}]$ and the sparse matrix $[\Omega^{(1)}, \dots, \Omega^{(m)}, \dots, \Omega^{(g)}]$
- From (**??**), an estimate of the covariance matrix for the $\Sigma^{(m)}$ is given by

$$\hat{\Sigma}^{(m)} = \hat{\Lambda}^{(m)\mathrm{T}} \hat{\Lambda}^{(m)} + \hat{\Omega}^{(m)}, \quad m = 1, \dots g$$

- An estimate of the originial covariance matrix $\hat{\Sigma}$ is given by

$$\hat{\Sigma} = \begin{bmatrix} \hat{\Sigma}^{(1)} & 0 & \dots & 0 \\ 0 & \hat{\Sigma}^{(2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{\Sigma}^{(g)} \end{bmatrix}$$

- Independence across sub-estimates ignores the inherent dependence structure in the observed $y_i s$

Hierarchical Modeling of the Latent Factor Structure

- Conside the hierarchical model on the latent factors $\eta_i^{(m)} \in \mathbb{R}^{k_g}$,

$$\eta_i^{(m)} \mid X_i, Z_i^{(m)} = \sqrt{\rho}\, X_i + \sqrt{1-\rho}\, Z_i^{(m)}, \quad i = 1, \ldots, n, \quad m = 1, \ldots, g \tag{1}$$

  where
  - $X_i \sim \mathcal{N}_{k_g}(0, \mathrm{I})$ shared across all the latent sub-factors
  - $Z_i^{(m)} \sim \mathcal{N}_{k_g}(0, \mathrm{I})$ is idiosyncratic to the sub-factor m
  - ρ is the correlation that will be induced between the sub-estimates obtained from the respective sub-groups
  - $\eta_i^{(m)} \sim \mathcal{N}_{k_g}(0, \mathrm{I})$ since
  $$\mathbb{E}\left(\eta_i^{(m)} \mid X_i, Z_i^{(m)}\right) = \mathbf{0}, \quad \mathbb{V}\left(\eta_i^{(m)} \mid X_i, Z_i^{(m)}\right) = \mathrm{I}$$
  - $\mathrm{Cov}(\eta_i^{(m)}, \eta_i^{(m')}) = \rho \mathrm{I}$

Posterior Computations: Parallel MCMCs I

1. Sample $X_i, i = 1, \ldots, n$ from conditionally independent Gaussian posteriors

$$X_i \mid \text{rest} \sim \mathcal{N}_{k_g}\left(\mu_{X_i}, \Sigma_{X_i}\right)$$

2. Sample $Z_i^{(m)} \mid \text{rest}, i = 1, \ldots, n, \; m = 1, \ldots, g$ from conditionally independent Gaussian posteriors

$$Z_i^{(m)} \mid \text{rest} \sim \mathcal{N}_{k_g}\left(\mu_{Z_i^{(m)}}, \Sigma_{Z_i^{(m)}}\right)$$

3. Update $\eta_i^{(m)} \mid \text{rest}$

$$\eta_i^{(m)} \mid \text{rest} = X_i \mid \text{rest} + Z_i^{(m)} \mid \text{rest}$$

C step

The estimate for the original covariance matrix Σ is given by

$$\hat{\Sigma} = \hat{\mathbf{D}}\hat{\mathbf{E}}\hat{\mathbf{D}}^{\mathrm{T}} + \hat{\Omega}$$

where

- $\hat{\mathbf{D}} = \operatorname{diag}\left(\hat{\Lambda}^{(1)}, \ldots, \hat{\Lambda}^{(m)}\right)$
- $\hat{\mathbf{E}} = I_{k_g}\ I(i = j) + \hat{\rho} I_{k_g}\ I(i \neq j) \in \mathbb{R}^{k_g \times k_g}$ consists of k_g^2 block matrices
- For $g = 2$ groups, an estimate of the covariance matrix $\hat{\Sigma}$ is given by

$$\begin{bmatrix} \hat{\Lambda}^{(1)}\hat{\Lambda}^{(1)\mathrm{T}} + \hat{\Omega}^{(1)} & \rho\hat{\Lambda}^{(1)}\hat{\Lambda}^{(2)\mathrm{T}} \\ \rho\hat{\Lambda}^{(1)}\hat{\Lambda}^{(2)\mathrm{T}} & \hat{\Lambda}^{(2)}\hat{\Lambda}^{(2)\mathrm{T}} + \hat{\Omega}^{(2)} \end{bmatrix}$$

Computational Gains: Running time of DnC

Key facts:

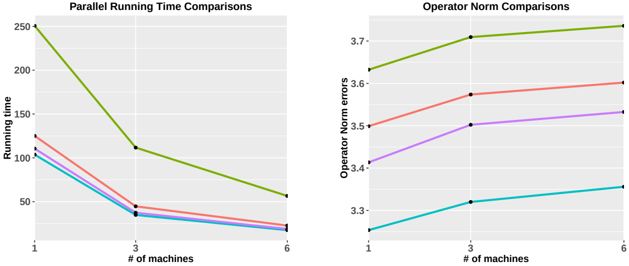
- If $B_1 \in \mathbb{R}^{m_1 \times m_2}$, $B_2 \in \mathbb{R}^{m_2 \times m_3}$ then $B_1 B_2$ requires $\mathcal{O}(m_1 m_2 m_3)$ floating point operations.
- If $B \in \mathbb{R}^{m \times m}$, then $Bx = y$ can be solved in $\mathcal{O}(m^3)$ operations.
- If D is diagonal, then $u \sim \mathcal{N}_p(0, D)$ can be carried out in $\mathcal{O}(p)$ floating point operations.
- Given $k_g < k$ and $p_g \ll p$

$$\mathcal{O}(k^3 + npk + nk^2 + pk^2) \to \mathcal{O}(k_g^3 + np_g k_g + nk_g^2 + p_g k_g^2)$$

Simulation Settings

- Explore the decrease in statistical accuracy and speed-up of DNC in a variety of experimental simulation settings
- Comparison with
  - Full factor model $(g = 1)$ using the MGPS prior
  - Factor model with 3 groups using the MGPS prior
  - Factor model with 6 groups using the MGPS prior
- Sample sizes: $n = 100, 200$
- Size of the dimension: $p = 252, 504, 1008, 2016$
- True number of factors: $k = 6, 12, 24, 36$
- The true covariance is generated from a factor model with idiosyncratic error $\sigma^2 \mathrm{I}_p$ where
  1. $\sigma^2 = 0.5$
  2. $\sigma^2 \sim \mathcal{U}(1, 5)$

Simulation Results I

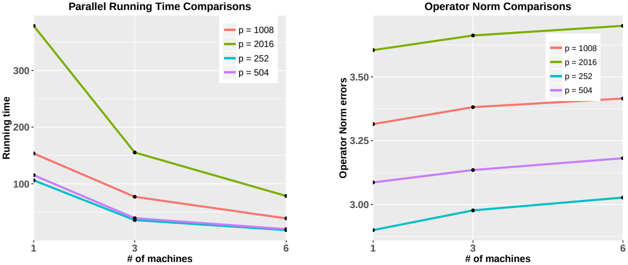


(a) Parallel running time per replicate

(b) Operator norm error in log scale

Figure : Simulation Model 1: Operator norm and parallel running time per replicate (in minutes) comparisons over 20 replicates for $n = 100$.

Simulation Results II



(a) Parallel running time per replicate

(b) Operator norm error in log scale

Figure : Simulation Model 2: Operator norm and parallel running time per replicate (in minutes) comparisons over 20 replicates for $n = 200$.

Simulation Results III

Table : Comparative performance in covariance matrix estimation in a simulation study where $p \asymp 10^4$. Average, best and worst performance reported in terms of operator norm errors with standard errors in parantheses.

p	**10000**			**20000**		
k	**100**			**200**		
g	**1**	**10**	**20**	**1**	**10**	**20**
Error	Fail	46.81 (0.11)	47.28 (0.09)	Fail	49.35 (0.16)	51.39 (0.11)
maxError	Fail	47.30	47.37	Fail	49.31	50.11
minError	Fail	46.62	47.06	Fail	49.65	52.39
Time	Fail	1626	998	Fail	2234	1276

Theoretical properties

Lemma

Suppose $\mathrm{rank}(\Lambda^{(m)}) = k_g, m = 1, \ldots, g$ and $\mathrm{rank}(\Lambda) = k$, then $A = \Lambda\Lambda^{\mathrm{T}}$ and $A^* = DED^{\mathrm{T}}$ have the same rank.

Remark. The approximation DED^{T} preserves the rank aposteriori.