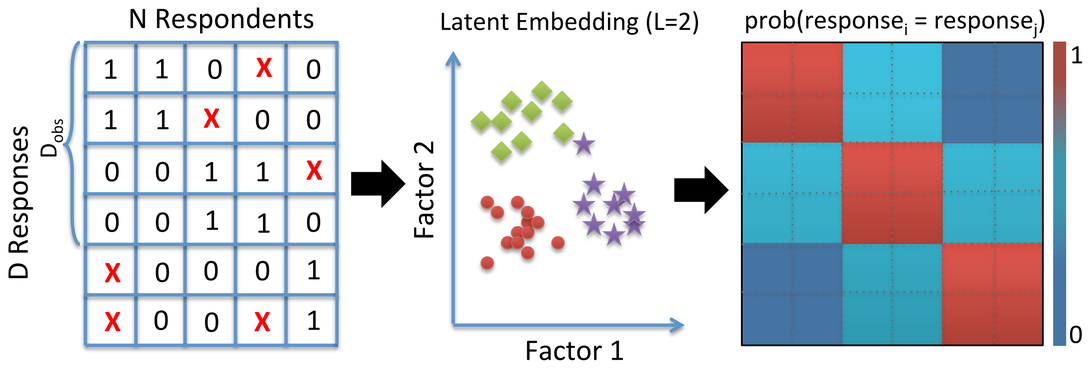
Large-Scale Approximate Bayesian Learning for Non-Conjugate Latent Gaussian Models

Mohammad Emtiyaz Khan, **Shakir Mohamed** and Kevin P. Murphy

University of British Columbia

Introduction

Motivation: Analysis of high-dimensional data is essential in applications in information retrieval, econometrics, social sciences, and medical diagnostics. Such analysis can be carried out using latent Gaussian models, such as factor analysis and PCA.



Contributions:

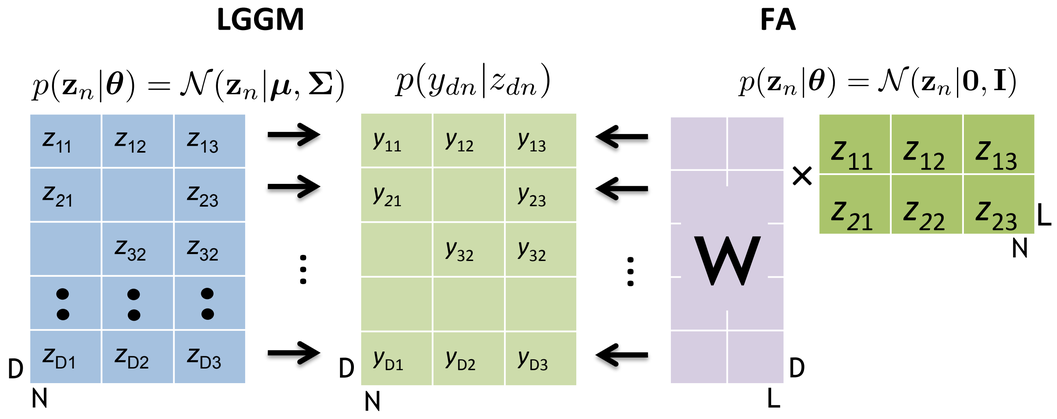
1. We derive a variational EM algorithm where the lower bounds are obtained using **concave optimization**.
2. For sparse data where $D_{obs} \ll D < N$, we **accelerate the E and M steps** resulting in a faster algorithm.

Non-conjugate Factor Analysis (FA) does not scale well to large data. For data with N observations, D-dimensional features and L latent factors, the computational cost is $O(NL^3 + NDL^2)$ and the memory cost is $O(NL^2)$. For large data, L is usually close to D, bringing these costs for computation and memory closer to $\mathbf{O(ND^3)}$ and $\mathbf{O(ND^2)}$, respectively.

- We use **latent Gaussian Graphical models** as an alternative to factor analysis, for which we derive **fast inference and learning** algorithms.
- When the data has many missing values, we obtain a reduction in:
  - Computational cost from $O(ND^3)$ to $O(ND^3_{obs})$
  - Memory cost from $O(ND^2)$ to $O(D^2)$

Latent Gaussian Graphical Models

Latent Gaussian Graphical Models (LGGMs) use a correlated latent Gaussian vector of length equal to the data dimension D, unlike models such as FA that use a low-dimensional vector of length $L \ll D$. For the nth data vector, the generative process is: (1) Sample latent Gaussian vectors $\mathbf{z}_n|\boldsymbol{\theta} \in \mathbb{R}^D$. (2) Draw data $\mathbf{y}_n$ from an appropriate distribution given $\mathbf{z}_n$.



Concave Lower Bounds

Variational lower bound: Computation of the marginal likelihood is intractable since the likelihood is not conjugate to the Gaussian prior. Using Jensen's inequality, we can obtain a lower bound to the marginal likelihood. We assume $\mu = 0$.

$$\mathcal{L}(\theta) = \sum_{n=1}^{N} \log \int \prod_{d \in D_{obs}} p(y_{dn}|\mathbf{z}, \theta) \mathcal{N}(\mathbf{z}|\mathbf{0}, \boldsymbol{\Sigma}) d\mathbf{z} = \sum_{n=1}^{N} \log \int \frac{\prod_{d \in D_{obs}} p(y_{dn}|\mathbf{z}, \theta) \mathcal{N}(\mathbf{z}|\mathbf{0}, \boldsymbol{\Sigma})}{\mathcal{N}(\mathbf{z}|\mathbf{m}_n, \mathbf{V}_n)} \mathcal{N}(\mathbf{z}|\mathbf{m}_n, \mathbf{V}_n) d\mathbf{z}$$

$$\geq \sum_{n=1}^{N} \max_{\mathbf{m}_n, \mathbf{V}_n} -KL\left[\mathcal{N}(\mathbf{m}_n, \mathbf{V}_n) || \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})\right] + \sum_{d \in D_{obs}} \int \log p(y_{dn}|\mathbf{z}, \theta) \mathcal{N}(\mathbf{z}|\mathbf{m}_n, \mathbf{V}_n) d\mathbf{z}$$

The expectation $\mathbb{E}[\log p(y_{dn}|\mathbf{z})]$ is not available for most non-Gaussian distributions. We obtain a tractable bound using a local variational bound, such that $\mathbb{E}[\log p(y_{dn}|\mathbf{z})] \geq f_b(y_{dn}, \mathbf{m}_{dn}, \mathbf{V}_{dd,n})$.

$$\geq \sum_{n=1}^{N} \max_{\mathbf{m}_n, \mathbf{V}_n} \frac{1}{2}\left[\log|\mathbf{V}_n\boldsymbol{\Sigma}^{-1}| - \mathrm{tr}(\mathbf{V}_n\boldsymbol{\Sigma}^{-1}) - \mathbf{m}_n^T\boldsymbol{\Sigma}^{-1}\mathbf{m}_n + D\right] + \sum_{d \in D_{obs}} f_b(y_{dn}, \mathbf{m}_{dn}, \mathbf{V}_{dd,n}) \quad (1)$$

Bound Optimization:

- Our variational bound is strictly concave when f_b is **jointly concave** with respect to $\mathbf{m}_n, \mathbf{V}_n$.
- Given $\mathbf{V}_n$, optimization w.r.t. $\mathbf{m}_n$ is a **non-linear least-squares** function.
- Given $\mathbf{m}_n$, optimization w.r.t. $\mathbf{V}_n$ is a form of **covariance selection** or graphical Lasso.

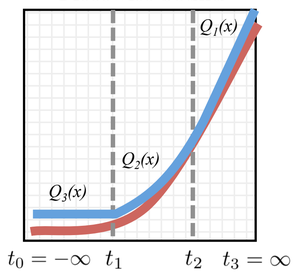
Local Variational Bounds

Type	Distribution	$p(y\|\eta)$	$\mathbb{E}[\log p(y\|\eta)]$
Count	Poisson	$p(y=k\|\eta) = \frac{e^{-e^\eta} e^{k\eta}}{k!}$	$ym - \exp(m + v/2) - \log y!$
Binary	Bernoulli logit	$p(y=1\|\eta) = \sigma(\eta)$	$ym - \mathbb{E}[\mathrm{llp}(\eta)]$
Categorical	Multinomial logit	$p(y=k\|\boldsymbol{\eta}) = e^{\eta_k - \mathrm{lse}(\eta)}$	$\mathbf{y}^T\mathbf{m} - \mathbb{E}[\mathrm{lse}(\boldsymbol{\eta})]$
Ordinal	Cumulative logit	$p(y \leq k\|\eta) = \sigma(\phi_k - \eta)$	$m - \mathbb{E}[\mathrm{llp}(-\phi_y + \eta)] - \mathbb{E}[\mathrm{llp}(-\phi_{y-1} + \eta)]$

The distributions above can be expressed in terms of $llp(x) = \log(1 + \exp(x))$, the logistic log-partition function (LLP), or the log-sum-exp (lse) function. Many convex bounds exist for these functions, ensuring concavity of the entire objective function.

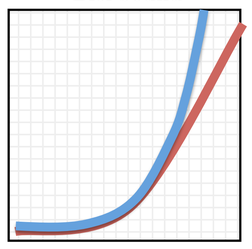
Bound for $log(1 + e^x)$

Piecewise bound



Bound for log-sum-exp

Blei bound



Large-Scale Variational Inference

The number of variational parameters, i.e. $\mathbf{m}_{1:N}$ and $\mathbf{V}_{1:N}$, is $O(ND^2)$, making optimization slow and inefficient. We **reparameterize** the objective function to reduce the number of parameters and the computational cost.

For sparse data, we only work with the observed entries. The set of observed entries will be indexed by O. We let $[\mathbf{a}_O]$ be a vector of length D, with entries O equal to $\mathbf{a}$ and the remaining entries being zero. Similarly, $[\mathbf{A}_{OO}]$ is a $D \times D$ matrix with zeros except the submatrix $\mathbf{A}_{OO}$. For simplicity, we drop the subscript n.

Using the derivative of the lower bound w.r.t. $\mathbf{m}$ and $\mathbf{V}$, we can reparameterize the objective function in terms of new parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$:

$$\text{Since } -\boldsymbol{\Sigma}^{-1}\mathbf{m} + \left[\frac{\partial f_b}{\partial \mathbf{m}}\right] = 0 \text{ allows us to reparameterize } \mathbf{m} = \boldsymbol{\Sigma}_{:,O}\boldsymbol{\alpha}_O$$

$$\text{Similarly, } \mathbf{V}^{-1} - \boldsymbol{\Sigma}^{-1} + \mathrm{diag}\left(\left[\frac{\partial f_b}{\partial \mathbf{V}}\right]\right) = 0 \text{ gives us } \mathbf{V} = (\boldsymbol{\Sigma}^{-1} + \mathrm{diag}([\boldsymbol{\lambda}_O]))^{-1} = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}_{:,O}\underbrace{(\mathrm{diag}(\boldsymbol{\lambda}_O)^{-1} + \boldsymbol{\Sigma}_{OO})^{-1}}_{\mathbf{B}}\boldsymbol{\Sigma}_{O,:}$$

Substituting this into equation 1, the objective functions for the nth data point is:

$$\underline{\mathcal{L}}_R(\boldsymbol{\alpha}_O, \boldsymbol{\lambda}_O) = \frac{1}{2}\left(-\log(|\mathbf{B}\,\mathrm{diag}(\boldsymbol{\lambda}_O)|) + \mathrm{trace}(\mathbf{B}^{-1}\boldsymbol{\Sigma}_{OO}) - \boldsymbol{\alpha}_O^T\boldsymbol{\Sigma}_{OO}\boldsymbol{\alpha}_O\right) + \sum_{d \in D_{obs}} f_b(y_d, m_d, V_{dd}) \quad (2)$$

We optimize w.r.t. $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$, bringing down the number of parameters to $O(ND_{obs})$. Computation of $\mathbf{V}$ requires inversion of $\mathbf{B}$, making the computation $O(ND^3_{obs})$.

Large-Scale Variational Learning

Using the reparameterization, the update for $\boldsymbol{\Sigma}$ can be written in terms of $\boldsymbol{\alpha}_{On}$ and $\mathbf{B}_n$ as:

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{V}_n + \mathbf{m}_n\mathbf{m}_n^T = \boldsymbol{\Sigma} + \frac{1}{N}\boldsymbol{\Sigma}\underbrace{\left(\sum_{n=1}^{N}[\boldsymbol{\alpha}_{On}\boldsymbol{\alpha}_{On}^T] - [\mathbf{B}_n^{-1}]\right)}_{\mathbf{S}}\boldsymbol{\Sigma}$$

Thus, we never form the matrix $\mathbf{V}_n$ and only work with $\mathbf{B}_n$ by maintaining a $D \times D$ matrix $\mathbf{S}$ and filling parts of it with $\mathbf{B}_n^{-1}$. This reduces the memory cost to $O(D^2)$

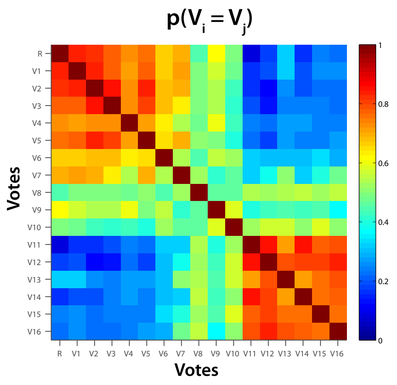
Algorithm: EM for LGGM learning with missing entries

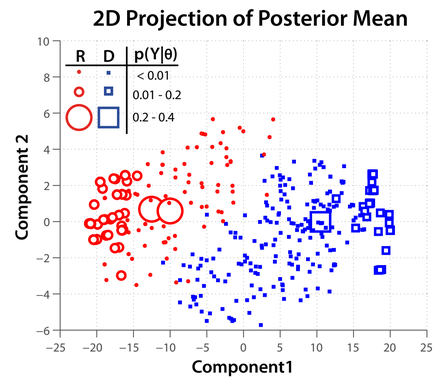
- Initialize $\boldsymbol{\Sigma}$.
- Iterate the following until convergence.
  1. Form $\mathbf{S}$ of size $D \times D$ with all entries set to 0.
  2. For $n = 1, \ldots, N$
     - Get the observed data $\mathbf{y}_{On}$ and corresponding subblock $\boldsymbol{\Sigma}_{OO}$.
     - Compute $\boldsymbol{\alpha}_{On}$ and $\boldsymbol{\lambda}_{On}$ by maximizing equation 2.
     - $\mathbf{S}_{OO} \leftarrow \mathbf{S}_{OO} + [\boldsymbol{\alpha}_{On}\boldsymbol{\alpha}_{On}^T] - [\mathbf{B}_n^{-1}]$.
  3. Compute $\boldsymbol{\Sigma} \leftarrow \boldsymbol{\Sigma} + \boldsymbol{\Sigma}\mathbf{S}\boldsymbol{\Sigma}/N$.

Results

Illustrating LGGM on voting dataset ($D = 16, N = 435$). We plot the probability of two variables (votes) taking the same value. We also plot a 2-D projection of the posterior mean of the latent variables with size proportional to the log-likelihood.

	Description
V1	budget-resolution
V2	aid-nicaraguan-contras
V3	mx-missile
V4	duty-free-exports
V5	anti-satellite-test-ban
V6	handicapped-infants
V7	export-admin-south-africa
V8	immigration
V9	synfuels-corp-cutback
V10	water-project-cost-sharing
V11	physician-fee-freeze
V12	El-salvador-aid
V13	religious-groups-in-schools
V14	education-spending
V15	superfund-right-to-sue
V16	crime





Large Synthetic Data ($D = 100, N = 240$) We generate a large binary synthetic dataset consisting of 12 patterns. Each pattern is replicated 20 times and 5% of the entries are flipped. We artificially create missing values in this dataset, such that only 35% of the dataset is observed. We impute the missing values using LGGM and FA.