Approximate Bayesian inference in high-dimensional applications

Barbara E Engelhardt

Department of Computer Science
Center for Statistics and Machine Learning
Princeton University

December 9, 2016

Motivation and question

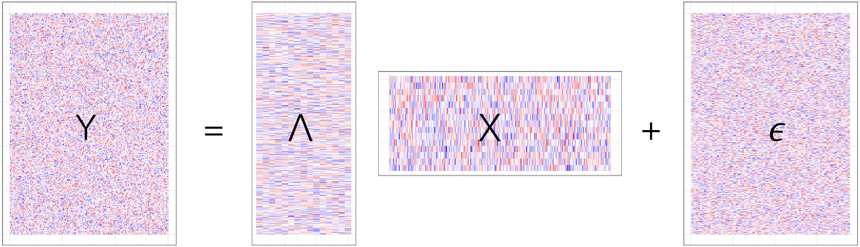
Variational inference is not robust for complex hierarchical models fitted to high-dimensional data

- how can we combine results among different VI estimates?
- what can we say about the estimates from these aggregations?

What problem are we trying to solve?

- Main goal here is parameter inference, not prediction
- Two local optima with the same evidence lower bound (elbo) are not equivalent, because they highlight different signals in the data

Factor analysis: linear map of high dimensional data



Matrix Y is observations of p features over n samples (this is the transpose of classical FA, for data reasons)

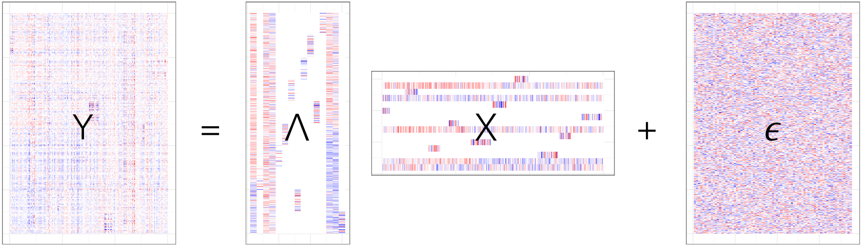
- *Factor analysis*: project matrix Y onto a linear subspace Λ (*loadings*) using weights X (*factors*), assuming Gaussian noise ϵ:

$$Y_{j,i} \sim \mathcal{N}\left(\sum_{k=1}^{K} \Lambda_{j,k} X_{k,i}, \psi_j^{-1}\right)$$

Bayesian biclustering for genomic data

Contributions to variation in gene expression levels are

- sparse & dense in genes: small sets of genes may be affected by covariates
- sparse & dense in samples: genotype, cell type, sex, smoking status



We build a model for *biclustering*, creating non-disjoint clusters in both genes and samples

Bayesian biclustering model

We put a *three parameter beta* prior on the factors and loadings:

$$
\begin{aligned}
\varrho &\sim \mathcal{TPB}(e, f, \nu), \\
\zeta_k &\sim \mathcal{TPB}\left(c, d, \frac{1}{\varrho} - 1\right) \\
\varphi_{i,k} &\sim \mathcal{TPB}\left(a, b, \frac{1}{\zeta_k} - 1\right), \\
\Lambda_{i,k} &\sim \mathcal{N}\left(0, \frac{1}{\varphi_{i,k}} - 1\right),
\end{aligned}
$$

and similarly for factors $\mathbf{X}$ to induce sparsity.

Bayesian biclustering model: Regularization

Regularization on **X** (structurally identical to regularization for Λ), can be written as *[Armagan, Dunson, Clyde 2011]*:

$$
\begin{aligned}
\varphi &\sim \mathcal{G}a(f_X, \xi), \\
\chi &\sim \mathcal{G}a(e_X, \varphi), \\
\kappa_k &\sim \mathcal{G}a(d_X, \chi), \\
\omega_k &\sim \mathcal{G}a(c_X, \kappa_k) \\
\rho_{k,i} &\sim \mathcal{G}a(b_X, \omega_k), \\
\sigma_{k,i} &\sim \pi \mathcal{G}a(a_X, \rho_{k,i}) + (1 - \pi)\delta(\omega_k) \\
x_{k,i} &\sim \mathcal{N}(0, \sigma_{k,i}),
\end{aligned}
$$

Bayesian biclustering model

To allow both sparse and dense factors and loadings, we use a two-component mixture:

$$\varphi_{i,k} \sim \pi \mathcal{TPB}\left(a, b, \frac{1}{\zeta_k} - 1\right) + (1-\pi)\delta(\zeta_k),$$

where the indicator variable z_k has a beta Bernoulli distribution:

$$\begin{aligned} \pi|\alpha, \beta &\sim Be(\alpha, \beta) \\ z_k|\pi &\sim \text{Bern}(\pi), k = \{1, \ldots, K\}. \end{aligned}$$

Recovering gene networks from factor models

Marginalizing over **X**, FA becomes regularized covariance estimation:

$$\begin{aligned}\mathbf{Y}_i &\sim \mathcal{N}_p(0, \Omega) \text{ for } i = 1, \ldots, n \\ \Omega &= \Lambda \Sigma \Lambda^T + \Psi,\end{aligned}$$

where Σ is the $K \times K$ covariance matrix for **X**.

- If we invert Ω, we recover the precision matrix for the genes
- (Normalized) precision matrix represents partial correlation of every gene pair: $cor(x_j, x_{j'} | x_{\neg j, j'})$
- Thresholding the precision matrix (FDR), we recover a Gaussian Markov random field across genes

,

Context-specific gene co-expression networks

We can subset the components in the biclustering model to recover interesting types of co-expression networks:

$$\begin{aligned} A &\subseteq \{1, \ldots, K\} \\ \Omega_A &= \Lambda_A \Sigma_{A,A} \Lambda_A^T + \Psi. \end{aligned}$$

If we invert Ω_A, we recover the precision matrix for the genes that load onto the components in A.

We choose to subset A as follows:

- *Ubiquitous networks*: factor is dense across samples
- *Differential networks*: factor modes across two sample subtypes differ
- *Context-specific networks*: factor is non-zero only for sample subtype

Variational expectation maximization

The variational approximation of $p(\mathbf{\Lambda}, \mathbf{X}, \mathbf{z}, \mathbf{o}, \mathbf{\Theta}|\mathbf{Y})$ is written as:

$$q(\mathbf{\Lambda}, \mathbf{X}, \mathbf{z}, \mathbf{o}, \mathbf{\Theta}) = p(\mathbf{\Lambda}|\mathbf{z}, \mathbf{\Theta}_{\mathbf{\Lambda}})p(\mathbf{X}|\mathbf{o}, \mathbf{\Theta}_{\mathbf{X}})p(\mathbf{z}|\mathbf{\Theta}_{\mathbf{\Lambda}})p(\mathbf{o}|\mathbf{\Theta}_{\mathbf{X}})p(\mathbf{\Theta}_{\Lambda})p(\mathbf{\Theta}_{X})$$

where $\mathbf{\Theta}_{\Lambda}$ and $\mathbf{\Theta}_{X}$ denote the parameters of $\mathbf{\Lambda}$ and $\mathbf{X}$, respectively. Then,

$$\begin{aligned}
p(\mathbf{\Lambda}, \mathbf{z}, \mathbf{\Theta}_{\mathbf{\Lambda}}) &= p(\mathbf{\Lambda}|\mathbf{z}, \mathbf{\Theta}_{\mathbf{\Lambda}})p(\mathbf{z}|\mathbf{\Theta}_{\mathbf{\Lambda}})p(\mathbf{\Theta}_{\Lambda}) \\
&= \left[\prod_{j=1}^{p}\prod_{k=1}^{K}\mathcal{N}(\Lambda_{j,k}|\theta_{j,k})\mathcal{G}a(\theta_{j,k}|a, \delta_{j,k})\mathcal{G}a(\delta_{j,k}|b, \phi_k)\right]^{\mathbb{1}_{z_k=1}} \\
&\quad \times \left[\prod_{j=1}^{p}\prod_{k=1}^{K}\mathcal{N}(\Lambda_{j,k}|\phi_k)\right]^{\mathbb{1}_{z_k=0}} \left[\prod_{k=1}^{K}\mathcal{B}ern(z_k|\pi)\right]\mathcal{B}eta(\pi|\alpha, \beta) \\
&\quad \times \left[\prod_{k=1}^{K}\mathcal{G}a(\phi_k|c, \tau_k)\mathcal{G}a(\tau_k|d, \eta)\right]\mathcal{G}a(\eta|e, \gamma)\mathcal{G}a(\gamma|f, \nu).
\end{aligned}$$

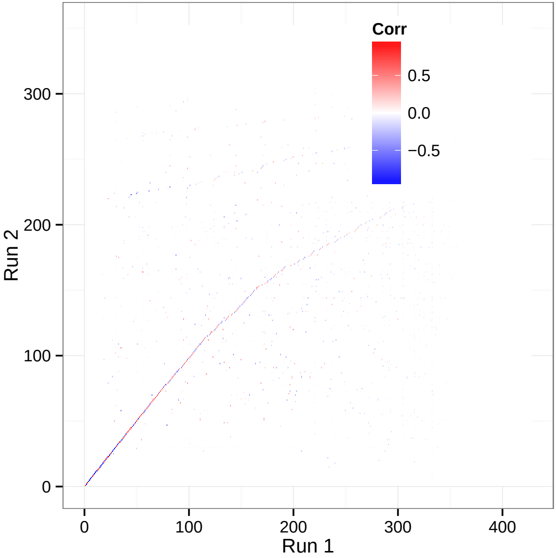
Variational expectation maximization

- random initialization
  - generate parameters from variational approximation
  - specifically, generate $\mathbf{\Lambda} \sim \mathcal{N}(0, I)$
- iterate until convergence
  - E-step
    - compute the expected value of $z_{1:K}$
    - compute the expected value of $\mathbf{X}$
    - compute the expected value of $\mathbf{X}\psi_{j,j}^{-1}\mathbf{X}^T$
  - variational M-step: coordinate ascent variational inference
    - $\hat{\mathbf{\Theta}}_\Lambda = arg \min_{q(\Theta_\Lambda)} KL(q(\mathbf{\Theta}_\Lambda)||p(\mathbf{\Theta}_\Lambda|\mathbf{Y}))$
    - convergence defined by evidence lower bound:

$$elbo(q) = E[\log p(Y, \Theta)] + E[\log q(\Theta)]$$

    - specifically, update Λ in a greedy way

VEM results not robust to random initializations



Variational EM: first try to robustify results

- We run variational EM 1,000 times with random restarts.
- We build a network from the results from each run
- We let each network "vote" on the network edges: edge is in the network if number of models that it appears in is $\geq r$

Related ideas in combining across approximate marginals

- Bagging (bootstrap aggregation) *[Breiman 1996]*
- Firefly Monte Carlo *[Maclaurin & Adams 2014]*
- Median posterior *[Minsker, Srivastava, Lin, Dunson 2014]*
- Structured stochastic variational inference *[Hoffman & Blei 2015]*
- Intersection of sparse factors across tensor decomposition runs *[Hore et al. 2016]*

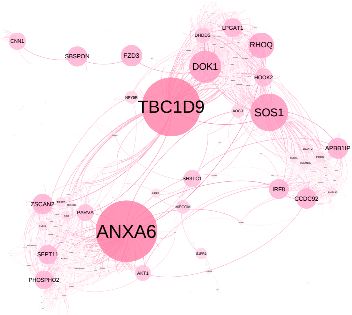
Tissue-specific networks

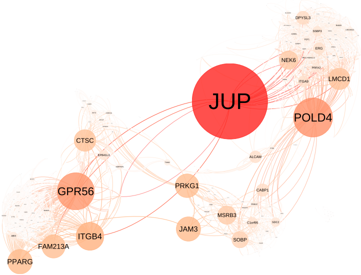
Adipose-specific network

- *RHOQ* involved in glucose uptake
- *ANXA6* reduces cholesterol
- *DOK1* mediates diet-induced obesity

Artery-specific network

- *JUP-81* atherosclerotic plaques
- *PPAR gamma* lipid metabolism and atherogenesis
- *ETS* arterial specification





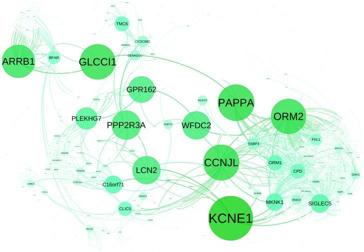
Tissue-specific networks

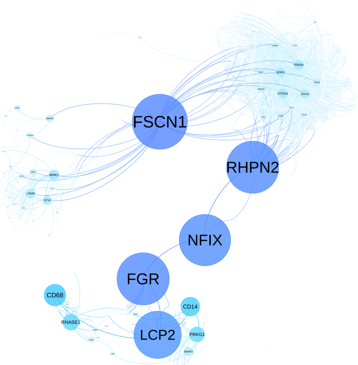
Lung-specific network

- *KCNE1* lung lobectomy responsive
- *PAPPA* lung cancer growth
- *ARRB1* nicotine-induced growth of lung tumors

Skin-specific network

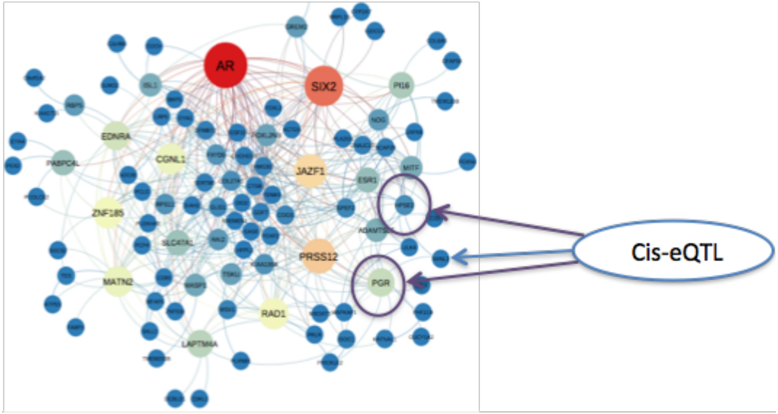
- *RHPN2* cancer initiatiator
- *CD68* skin tumors growth





Validation of network edges

Given a gene of interest A, its associated genetic variant Q, and a gene B that is a neighbor of A in the tissue-specific network, we tested for association between Q and B in out of sample data.



Validated edges

Adipose network validation

- 85 trans-eQTLs ($FDR \leq 0.10$)
- trans-eQTL for *TK2*, deficiency causes abnormal adipose tissues

Artery network validation

- two trans-eQTLs ($FDR \leq 0.10$)
- trans-eQTLs for *PLVAP* and *CYYR1*, unique to artery samples

Lung network validation

- nine trans-eQTLs ($FDR \leq 0.15$)
- trans-eQTL for *DENND1C*, which is unique to lung samples

Skin network validation

- eight trans-eQTLs ($FDR \leq 0.25$)
- trans-eQTLs for *CDH3*, related to juvenile macular dystrophy

Summary

We developed Bayesian biclustering models and fitted these models to gene expression data using variational EM

- to identify sources of gene co-variation;
- to recover gene co-expression networks.

Ongoing work

- developing and formalizing methods to robustify results;
- use stochastic variational inference for additional stochasticity across runs;
- methods to combine across posterior estimates with different (non-Bernoulli) marginals

Acknowledgements

Princeton University:

- Derek Aguiar
- Li-Fang Cheng
- Greg Darnell
- **Bianca Dumitrascu**
- **Ariel Gewirtz**

Duke University:

- **Chuan Gao**
- **Shiwen Zhao**

- David Dunson
- Sayan Mukherjee

Collaborators:

- Ryan P Adams (Harvard)
- Casey Brown (UPenn)
- Patrick Flaherty (UMass Amherst)

Data sets:

- Cholesterol and Pharmacogenetics (CHORI)
- Genotype-Tissue Expression (GTEx)

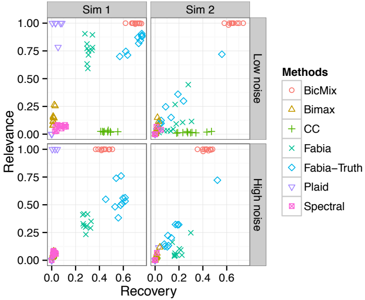
Funding:

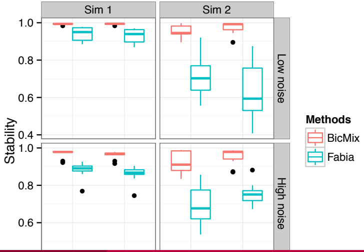
- NIH NHGRI R00 HG006265
- NIH GTEx R01 MH101822

Bayesian biclustering results on simulated data

- *Sim1*: Only sparse components
- *Sim2*: Sparse and dense components

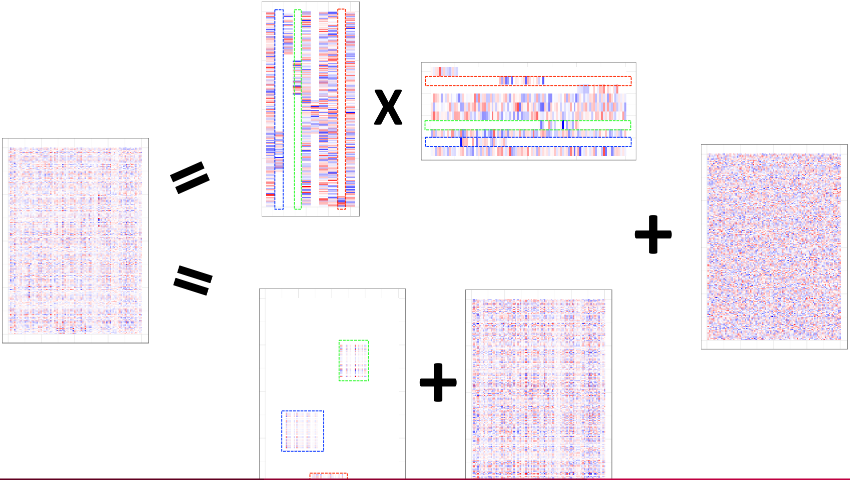
- *BicMix*: Our biclustering method
- *Bimax*: hierarchical clustering
- *CC*: hierarchical clustering
- *Fabia*: latent factor model
- *Plaid*: sparse matrix factorization
- *Spectral*: orthogonal matrix factorization





Biclustering model

Model for biclustering encodes subsets of samples, genes for which covariation is observed

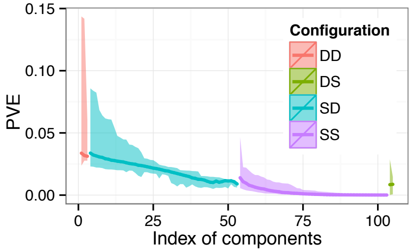


Bayesian biclustering results on GTEx data

- Genotype-Tissue Expression (GTEx) study
- Hundreds of individuals, RNA-seq on > 30 tissues per individual
- Whole-genome sequences for all individuals
- Here: data subset with four tissues, ~ 200 individuals

- BicMix identified 9,854 unique sparse components across 200 runs

- DD = Dense loading, dense factor (population structure)
- SD = Sparse loading, dense factor (age, BMI, batch)
- DS = Dense loading, sparse factor (bad sample)
- SS = Sparse loading, sparse factor (eQTLs, cell type, sex)



Median component-wise PVE for three DD, 50 SD, 50 SS, and two DS components