Locally (Conditionally) Conjugate Models

Piyush Rai

Probabilistic Machine Learning (CS772A)

September 12, 2017

Moving Beyond Simple Models..

- Looked at models with one "main" parameter and maybe a few fixed hyperparameters

Moving Beyond Simple Models..

- Looked at models with one "main" parameter and maybe a few fixed hyperparameters
  - E.g., probabilistic linear regression with weight vector $\boldsymbol{w}$ and fixed hyperparams β, λ

Moving Beyond Simple Models..

- Looked at models with one "main" parameter and maybe a few fixed hyperparameters
  - E.g., probabilistic linear regression with weight vector $\boldsymbol{w}$ and fixed hyperparams β, λ
  - Simple to compute posterior if the prior on main parameter is conjugate to the likelihood

Moving Beyond Simple Models..

- Looked at models with one "main" parameter and maybe a few fixed hyperparameters
  - E.g., probabilistic linear regression with weight vector $\boldsymbol{w}$ and fixed hyperparams β, λ
  - Simple to compute posterior if the prior on main parameter is conjugate to the likelihood
- Also looked at models with multiple unknowns, but inference usually becomes tricky, e.g.,

Moving Beyond Simple Models..

- Looked at models with one "main" parameter and maybe a few fixed hyperparameters
  - E.g., probabilistic linear regression with weight vector $\boldsymbol{w}$ and fixed hyperparams β, λ
  - Simple to compute posterior if the prior on main parameter is conjugate to the likelihood
- Also looked at models with multiple unknowns, but inference usually becomes tricky, e.g.,
  - E.g., when hyperparameter needed to be inferred, have to use MLE-II or EM (HW2)

Moving Beyond Simple Models..

- Looked at models with one "main" parameter and maybe a few fixed hyperparameters
  - E.g., probabilistic linear regression with weight vector $\boldsymbol{w}$ and fixed hyperparams β, λ
  - Simple to compute posterior if the prior on main parameter is conjugate to the likelihood
- Also looked at models with multiple unknowns, but inference usually becomes tricky, e.g.,
  - E.g., when hyperparameter needed to be inferred, have to use MLE-II or EM (HW2)
  - In LVMs with parameters and latent vars, have to use Expectation Maximization

Moving Beyond Simple Models..

- Looked at models with one "main" parameter and maybe a few fixed hyperparameters
  - E.g., probabilistic linear regression with weight vector $\boldsymbol{w}$ and fixed hyperparams β, λ
  - Simple to compute posterior if the prior on main parameter is conjugate to the likelihood
- Also looked at models with multiple unknowns, but inference usually becomes tricky, e.g.,
  - E.g., when hyperparameter needed to be inferred, have to use MLE-II or EM (HW2)
  - In LVMs with parameters and latent vars, have to use Expectation Maximization
- Usually, models with many unknowns do not have closed form posteriors in general

Moving Beyond Simple Models..

- Looked at models with one "main" parameter and maybe a few fixed hyperparameters
  - E.g., probabilistic linear regression with weight vector $\boldsymbol{w}$ and fixed hyperparams β, λ
  - Simple to compute posterior if the prior on main parameter is conjugate to the likelihood
- Also looked at models with multiple unknowns, but inference usually becomes tricky, e.g.,
  - E.g., when hyperparameter needed to be inferred, have to use MLE-II or EM (HW2)
  - In LVMs with parameters and latent vars, have to use Expectation Maximization
- Usually, models with many unknowns do not have closed form posteriors in general
- Today, will look at a general scheme for doing <u>approximate</u> posterior inference in such models

Moving Beyond Simple Models..

- Looked at models with one "main" parameter and maybe a few fixed hyperparameters
  - E.g., probabilistic linear regression with weight vector $\boldsymbol{w}$ and fixed hyperparams β, λ
  - Simple to compute posterior if the prior on main parameter is conjugate to the likelihood
- Also looked at models with multiple unknowns, but inference usually becomes tricky, e.g.,
  - E.g., when hyperparameter needed to be inferred, have to use MLE-II or EM (HW2)
  - In LVMs with parameters and latent vars, have to use Expectation Maximization
- Usually, models with many unknowns do not have closed form posteriors in general
- Today, will look at a general scheme for doing approximate posterior inference in such models
  - Based on the idea of local or conditional conjugacy (yes, conjugacy helps even in complex problems!)

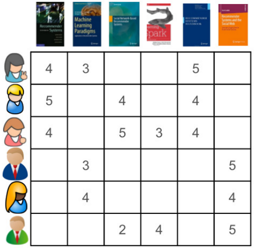
Moving Beyond Simple Models..

- Looked at models with one "main" parameter and maybe a few fixed hyperparameters
  - E.g., probabilistic linear regression with weight vector $\boldsymbol{w}$ and fixed hyperparams β, λ
  - Simple to compute posterior if the prior on main parameter is conjugate to the likelihood
- Also looked at models with multiple unknowns, but inference usually becomes tricky, e.g.,
  - E.g., when hyperparameter needed to be inferred, have to use MLE-II or EM (HW2)
  - In LVMs with parameters and latent vars, have to use Expectation Maximization
- Usually, models with many unknowns do not have closed form posteriors in general
- Today, will look at a general scheme for doing approximate posterior inference in such models
  - Based on the idea of local or conditional conjugacy (yes, conjugacy helps even in complex problems!)
  - Leads to "update one parameter keeping others fixed" style updates (similar in spirit to EM)

Moving Beyond Simple Models..

- Looked at models with one "main" parameter and maybe a few fixed hyperparameters
  - E.g., probabilistic linear regression with weight vector $\boldsymbol{w}$ and fixed hyperparams β, λ
  - Simple to compute posterior if the prior on main parameter is conjugate to the likelihood
- Also looked at models with multiple unknowns, but inference usually becomes tricky, e.g.,
  - E.g., when hyperparameter needed to be inferred, have to use MLE-II or EM (HW2)
  - In LVMs with parameters and latent vars, have to use Expectation Maximization
- Usually, models with many unknowns do not have closed form posteriors in general
- Today, will look at a general scheme for doing approximate posterior inference in such models
  - Based on the idea of local or conditional conjugacy (yes, conjugacy helps even in complex problems!)
  - Leads to "update one parameter keeping others fixed" style updates (similar in spirit to EM)
  - Local conjugacy makes each individual update have a simple form!

An Example Problem: Matrix Completion



4	3			5	
5		4		4	
4		5	3	4	
	3				5
	4				4
		2	4		5

- Given: Data $\mathcal{D} = \{r_{ij}\}$ of "interactions" (e.g., ratings) of users $i = 1, \ldots, N$ on $j = 1, \ldots, M$ items

An Example Problem: Matrix Completion

4	3			5	
5		4		4	
4		5	3	4	
	3				5
	4				4
		2	4		5

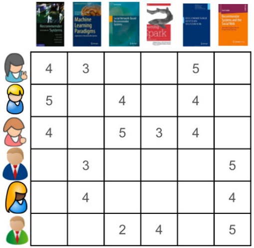
- Given: Data $\mathcal{D} = \{r_{ij}\}$ of "interactions" (e.g., ratings) of users $i = 1, \ldots, N$ on $j = 1, \ldots, M$ items
  - Note: "users" and "items" could mean other things too (depends on the data)

An Example Problem: Matrix Completion

4	3			5	
5		4		4	
4		5	3	4	
	3				5
	4				4
		2	4		5

- Given: Data $\mathcal{D} = \{r_{ij}\}$ of "interactions" (e.g., ratings) of users $i = 1, \ldots, N$ on $j = 1, \ldots, M$ items
  - Note: "users" and "items" could mean other things too (depends on the data)
- $(i, j) \in \Omega$ denotes an observed user-item pair. Ω is the set of all such pairs

An Example Problem: Matrix Completion



4	3			5	
5		4		4	
4		5	3	4	
	3				5
	4				4
		2	4		5

- Given: Data $\mathcal{D} = \{r_{ij}\}$ of "interactions" (e.g., ratings) of users $i = 1, \ldots, N$ on $j = 1, \ldots, M$ items
  - Note: "users" and "items" could mean other things too (depends on the data)
- $(i, j) \in \Omega$ denotes an observed user-item pair. Ω is the set of all such pairs
- Only a small number of user-item ratings observed, i.e., $|\Omega| \ll NM$

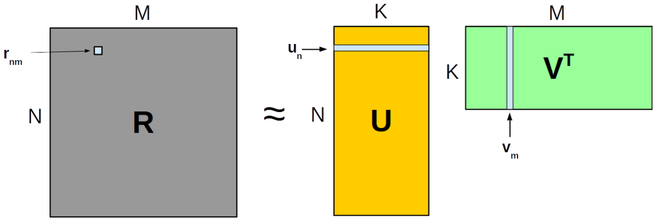
An Example Problem: Matrix Completion

4	3			5	
5		4		4	
4		5	3	4	
	3				5
	4				4
		2	4		5

- Given: Data $\mathcal{D} = \{r_{ij}\}$ of "interactions" (e.g., ratings) of users $i = 1, \ldots, N$ on $j = 1, \ldots, M$ items
  - Note: "users" and "items" could mean other things too (depends on the data)
- $(i, j) \in \Omega$ denotes an observed user-item pair. Ω is the set of all such pairs
- Only a small number of user-item ratings observed, i.e., $|\Omega| \ll NM$
- We would like to predict the unobserved values $r_{ij} \notin \mathcal{D}$

Matrix Completion via Matrix Factorization

- Suppose the "ratings" matrix **R** can be approximately factorized as a product of two matrices

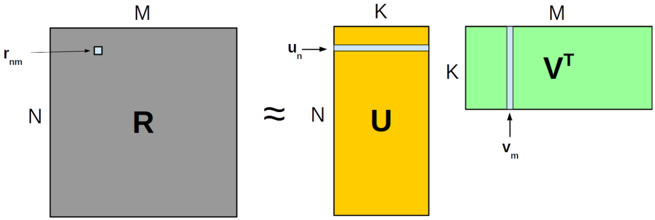


- Equivalent to writing each entry r_{nm} as

$$r_{nm} \approx \boldsymbol{u}_n^\top \boldsymbol{v}_m = \sum_{k=1}^{K} u_{nk} v_{mk}$$

Matrix Completion via Matrix Factorization

- Suppose the "ratings" matrix **R** can be approximately factorized as a product of two matrices



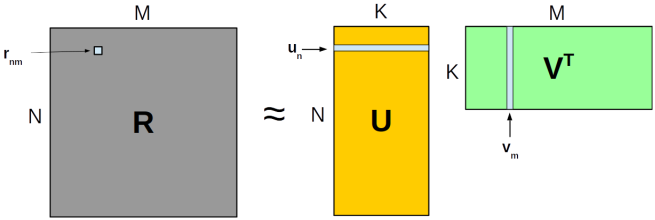
- Equivalent to writing each entry r_{nm} as

$$r_{nm} \approx \boldsymbol{u}_n^\top \boldsymbol{v}_m = \sum_{k=1}^{K} u_{nk} v_{mk}$$

- Note: Assume $K \ll N, M \Rightarrow$ **R** is approximately a low-rank matrix

Matrix Completion via Matrix Factorization

- Suppose the "ratings" matrix $\mathbf{R}$ can be approximately factorized as a product of two matrices



- Equivalent to writing each entry r_{nm} as

$$r_{nm} \approx \boldsymbol{u}_n^\top \boldsymbol{v}_m = \sum_{k=1}^{K} u_{nk} v_{mk}$$

- Note: Assume $K \ll N, M \Rightarrow \mathbf{R}$ is approximately a low-rank matrix

- Given $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$, any missing element r_{nm} in $\mathbf{R}$ can be predicted as $r_{nm} \approx \boldsymbol{u}_n^\top \boldsymbol{v}_m$

Probabilistic Matrix Factorization

- Can assume **R** to be a low-rank matrix plus some noise **E**

$$\mathbf{R} = \mathbf{U}\mathbf{V}^\top + \mathbf{E}$$

Probabilistic Matrix Factorization

- Can assume **R** to be a low-rank matrix plus some noise **E**

$$\mathbf{R} = \mathbf{U}\mathbf{V}^\top + \mathbf{E}$$

- $\mathbf{U} = [\boldsymbol{u}_1 \ldots \boldsymbol{u}_N]^\top$ is $N \times K$ and consists of the latent factors of the N users
    - $\boldsymbol{u}_i \in \mathbb{R}^K$ denotes the latent factors (or learned features) of user i

Probabilistic Matrix Factorization

- Can assume $\mathbf{R}$ to be a low-rank matrix plus some noise $\mathbf{E}$

$$\mathbf{R} = \mathbf{U}\mathbf{V}^\top + \mathbf{E}$$

- $\mathbf{U} = [\boldsymbol{u}_1 \ldots \boldsymbol{u}_N]^\top$ is $N \times K$ and consists of the latent factors of the N users
  - $\boldsymbol{u}_i \in \mathbb{R}^K$ denotes the latent factors (or learned features) of user i
- $\mathbf{V} = [\boldsymbol{v}_1 \ldots \boldsymbol{v}_M]^\top$ is $M \times K$ and consists of the latent factors of the M items
  - $\boldsymbol{v}_j \in \mathbb{R}^K$ denotes the latent factors (or learned features) of item j

Probabilistic Matrix Factorization

- Can assume $\mathbf{R}$ to be a low-rank matrix plus some noise $\mathbf{E}$

$$\mathbf{R} = \mathbf{U}\mathbf{V}^\top + \mathbf{E}$$

- $\mathbf{U} = [\boldsymbol{u}_1 \dots \boldsymbol{u}_N]^\top$ is $N \times K$ and consists of the latent factors of the N users
  - $\boldsymbol{u}_i \in \mathbb{R}^K$ denotes the latent factors (or learned features) of user i
- $\mathbf{V} = [\boldsymbol{v}_1 \dots \boldsymbol{v}_M]^\top$ is $M \times K$ and consists of the latent factors of the M items
  - $\boldsymbol{v}_j \in \mathbb{R}^K$ denotes the latent factors (or learned features) of item j
- $\mathbf{E} = \{\epsilon_{ij}\}$ consists of the "noise" in $\mathbf{R}$ (not captured by the low-rank assumption)

Probabilistic Matrix Factorization

- Can assume **R** to be a low-rank matrix plus some noise **E**

$$\mathbf{R} = \mathbf{U}\mathbf{V}^\top + \mathbf{E}$$

- $\mathbf{U} = [\boldsymbol{u}_1 \dots \boldsymbol{u}_N]^\top$ is $N \times K$ and consists of the latent factors of the N users
  - $\boldsymbol{u}_i \in \mathbb{R}^K$ denotes the latent factors (or learned features) of user i
- $\mathbf{V} = [\boldsymbol{v}_1 \dots \boldsymbol{v}_M]^\top$ is $M \times K$ and consists of the latent factors of the M items
  - $\boldsymbol{v}_j \in \mathbb{R}^K$ denotes the latent factors (or learned features) of item j
- $\mathbf{E} = \{\epsilon_{ij}\}$ consists of the "noise" in **R** (not captured by the low-rank assumption)
- We can write each element of matrix **R** as

$$r_{ij} = \boldsymbol{u}_i^\top \boldsymbol{v}_j + \epsilon_{ij} \qquad (i = 1, \dots, N, \quad j = 1, \dots, M)$$

A Bayesian Model for Matrix Factorization

- The low-rank matrix factorization model assumes

$$r_{ij} = \boldsymbol{u}_i^\top \boldsymbol{v}_j + \epsilon_{ij}$$

A Bayesian Model for Matrix Factorization

- The low-rank matrix factorization model assumes

$$r_{ij} = \boldsymbol{u}_i^\top \boldsymbol{v}_j + \epsilon_{ij}$$

- Let's assume the noise to be Gaussian $\epsilon_{ij} \sim \mathcal{N}(\epsilon_{ij}|0, \beta^{-1})$

A Bayesian Model for Matrix Factorization

- The low-rank matrix factorization model assumes

$$r_{ij} = \boldsymbol{u}_i^\top \boldsymbol{v}_j + \epsilon_{ij}$$

- Let's assume the noise to be Gaussian $\epsilon_{ij} \sim \mathcal{N}(\epsilon_{ij}|0, \beta^{-1})$
- This results in the following Gaussian likelihood for each observation

A Bayesian Model for Matrix Factorization

- The low-rank matrix factorization model assumes

$$r_{ij} = \boldsymbol{u}_i^\top \boldsymbol{v}_j + \epsilon_{ij}$$

- Let's assume the noise to be Gaussian $\epsilon_{ij} \sim \mathcal{N}(\epsilon_{ij}|0, \beta^{-1})$
- This results in the following Gaussian likelihood for each observation

$$p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j) =$$

A Bayesian Model for Matrix Factorization

- The low-rank matrix factorization model assumes

$$r_{ij} = \boldsymbol{u}_i^\top \boldsymbol{v}_j + \epsilon_{ij}$$

- Let's assume the noise to be Gaussian $\epsilon_{ij} \sim \mathcal{N}(\epsilon_{ij}|0, \beta^{-1})$
- This results in the following Gaussian likelihood for each observation

$$p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j) = \mathcal{N}(r_{ij}|\boldsymbol{u}_i^\top \boldsymbol{v}_j, \beta^{-1})$$

A Bayesian Model for Matrix Factorization

- The low-rank matrix factorization model assumes

$$r_{ij} = \boldsymbol{u}_i^\top \boldsymbol{v}_j + \epsilon_{ij}$$

- Let's assume the noise to be Gaussian $\epsilon_{ij} \sim \mathcal{N}(\epsilon_{ij}|0, \beta^{-1})$
- This results in the following Gaussian likelihood for each observation

$$p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j) = \mathcal{N}(r_{ij}|\boldsymbol{u}_i^\top \boldsymbol{v}_j, \beta^{-1})$$

- Assume Gaussian priors on the user and item latent factors

$$p(\boldsymbol{u}_i) = \mathcal{N}(\boldsymbol{u}_i|\mathbf{0}, \lambda_u^{-1}\mathbf{I}_K) \quad \text{and} \quad p(\boldsymbol{v}_j) = \mathcal{N}(\boldsymbol{v}_j|\mathbf{0}, \lambda_v^{-1}\mathbf{I}_K)$$

A Bayesian Model for Matrix Factorization

- The low-rank matrix factorization model assumes

$$r_{ij} = \boldsymbol{u}_i^\top \boldsymbol{v}_j + \epsilon_{ij}$$

- Let's assume the noise to be Gaussian $\epsilon_{ij} \sim \mathcal{N}(\epsilon_{ij}|0, \beta^{-1})$
- This results in the following Gaussian likelihood for each observation

$$p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j) = \mathcal{N}(r_{ij}|\boldsymbol{u}_i^\top \boldsymbol{v}_j, \beta^{-1})$$

- Assume Gaussian priors on the user and item latent factors

$$p(\boldsymbol{u}_i) = \mathcal{N}(\boldsymbol{u}_i|\mathbf{0}, \lambda_u^{-1}\mathbf{I}_K) \quad \text{and} \quad p(\boldsymbol{v}_j) = \mathcal{N}(\boldsymbol{v}_j|\mathbf{0}, \lambda_v^{-1}\mathbf{I}_K)$$

- The goal is to infer latent factors $\mathbf{U} = \{\boldsymbol{u}_i\}_{i=1}^N$ and $\mathbf{V} = \{\boldsymbol{v}_j\}_{j=1}^M$, given observed ratings from $\mathbf{R}$

A Bayesian Model for Matrix Factorization

- The low-rank matrix factorization model assumes

$$r_{ij} = \boldsymbol{u}_i^\top \boldsymbol{v}_j + \epsilon_{ij}$$

- Let's assume the noise to be Gaussian $\epsilon_{ij} \sim \mathcal{N}(\epsilon_{ij}|0, \beta^{-1})$
- This results in the following Gaussian likelihood for each observation

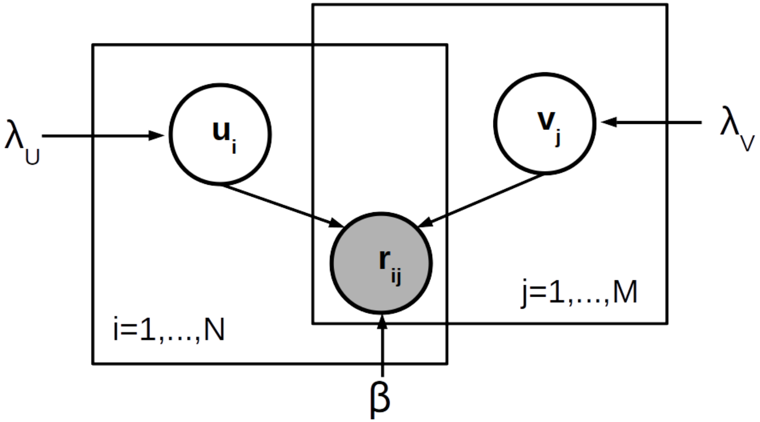
$$p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j) = \mathcal{N}(r_{ij}|\boldsymbol{u}_i^\top \boldsymbol{v}_j, \beta^{-1})$$

- Assume Gaussian priors on the user and item latent factors

$$p(\boldsymbol{u}_i) = \mathcal{N}(\boldsymbol{u}_i|\mathbf{0}, \lambda_u^{-1}\mathbf{I}_K) \quad \text{and} \quad p(\boldsymbol{v}_j) = \mathcal{N}(\boldsymbol{v}_j|\mathbf{0}, \lambda_v^{-1}\mathbf{I}_K)$$

- The goal is to infer latent factors $\mathbf{U} = \{\boldsymbol{u}_i\}_{i=1}^N$ and $\mathbf{V} = \{\boldsymbol{v}_j\}_{j=1}^M$, given observed ratings from $\mathbf{R}$
- For simplicity, we will assume the hyperparams $\beta, \lambda_u, \lambda_v$ to be fixed and not to be learned

Matrix Completion via Matrix Factorization



The Posterior

- Our target posterior distribution for this model will be

$$p(\mathbf{U}, \mathbf{V}|\mathbf{R}) = \frac{p(\mathbf{R}|\mathbf{U}, \mathbf{V})p(\mathbf{U}, \mathbf{V})}{\int \int p(\mathbf{R}|\mathbf{U}, \mathbf{V})p(\mathbf{U}, \mathbf{V})d\mathbf{U}d\mathbf{V}}$$

The Posterior

- Our target posterior distribution for this model will be

$$p(\mathbf{U},\mathbf{V}|\mathbf{R}) = \frac{p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})}{\int\int p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})d\mathbf{U}d\mathbf{V}} = \frac{\prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)}{\int\ldots\int \prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j) d\boldsymbol{u}_1\ldots d\boldsymbol{u}_N d\boldsymbol{v}_1\ldots d\boldsymbol{v}_M}$$

The Posterior

- Our target posterior distribution for this model will be

$$p(\mathbf{U}, \mathbf{V}|\mathbf{R}) = \frac{p(\mathbf{R}|\mathbf{U}, \mathbf{V})p(\mathbf{U}, \mathbf{V})}{\int \int p(\mathbf{R}|\mathbf{U}, \mathbf{V})p(\mathbf{U}, \mathbf{V})d\mathbf{U}d\mathbf{V}} = \frac{\prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)}{\int \ldots \int \prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j) d\boldsymbol{u}_1 \ldots d\boldsymbol{u}_N d\boldsymbol{v}_1 \ldots d\boldsymbol{v}_M}$$

- The denominator (and hence the posterior) is intractable!

The Posterior

- Our target posterior distribution for this model will be

$$p(\mathbf{U},\mathbf{V}|\mathbf{R}) = \frac{p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})}{\int\int p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})d\mathbf{U}d\mathbf{V}} = \frac{\prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)}{\int\dots\int\prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)d\boldsymbol{u}_1\dots d\boldsymbol{u}_N d\boldsymbol{v}_1\dots d\boldsymbol{v}_M}$$

- The denominator (and hence the posterior) is intractable! **Question: Why?**

The Posterior

- Our target posterior distribution for this model will be

$$p(\mathbf{U},\mathbf{V}|\mathbf{R}) = \frac{p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})}{\int\int p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})d\mathbf{U}d\mathbf{V}} = \frac{\prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)}{\int\ldots\int\prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)d\boldsymbol{u}_1\ldots d\boldsymbol{u}_N d\boldsymbol{v}_1\ldots d\boldsymbol{v}_M}$$

- The denominator (and hence the posterior) is intractable! **Question: Why?**
- Therefore, the posterior must be approximated somehow

The Posterior

- Our target posterior distribution for this model will be

$$p(\mathbf{U},\mathbf{V}|\mathbf{R}) = \frac{p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})}{\int\int p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})d\mathbf{U}d\mathbf{V}} = \frac{\prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)}{\int\dots\int \prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j) d\boldsymbol{u}_1\dots d\boldsymbol{u}_N d\boldsymbol{v}_1\dots d\boldsymbol{v}_M}$$

- The denominator (and hence the posterior) is intractable! **Question: Why?**
- Therefore, the posterior must be approximated somehow
- One way to approximate is to compute **Conditional Posterior** (CP) over each unknown, e.g.,

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V},\mathbf{U}_{-i}) \quad \text{and} \quad p(\boldsymbol{v}_j|\mathbf{R},\mathbf{U},\mathbf{V}_{-j})$$

The Posterior

- Our target posterior distribution for this model will be

$$p(\mathbf{U},\mathbf{V}|\mathbf{R}) = \frac{p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})}{\int\int p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})d\mathbf{U}d\mathbf{V}} = \frac{\prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)}{\int\ldots\int\prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)d\boldsymbol{u}_1\ldots d\boldsymbol{u}_N d\boldsymbol{v}_1\ldots d\boldsymbol{v}_M}$$

- The denominator (and hence the posterior) is intractable! **Question: Why?**
- Therefore, the posterior must be approximated somehow
- One way to approximate is to compute **Conditional Posterior** (CP) over each unknown, e.g.,

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V},\mathbf{U}_{-i}) \quad \text{and} \quad p(\boldsymbol{v}_j|\mathbf{R},\mathbf{U},\mathbf{V}_{-j})$$

.. in an alternating fashion until convergence

The Posterior

- Our target posterior distribution for this model will be

$$p(\mathbf{U}, \mathbf{V}|\mathbf{R}) = \frac{p(\mathbf{R}|\mathbf{U}, \mathbf{V})p(\mathbf{U}, \mathbf{V})}{\int\int p(\mathbf{R}|\mathbf{U}, \mathbf{V})p(\mathbf{U}, \mathbf{V})d\mathbf{U}d\mathbf{V}} = \frac{\prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)}{\int\ldots\int \prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j) d\boldsymbol{u}_1\ldots d\boldsymbol{u}_N d\boldsymbol{v}_1\ldots d\boldsymbol{v}_M}$$

- The denominator (and hence the posterior) is intractable! **Question: Why?**
- Therefore, the posterior must be approximated somehow
- One way to approximate is to compute **Conditional Posterior** (CP) over each unknown, e.g.,

$$p(\boldsymbol{u}_i|\mathbf{R}, \mathbf{V}, \mathbf{U}_{-i}) \quad \text{and} \quad p(\boldsymbol{v}_j|\mathbf{R}, \mathbf{U}, \mathbf{V}_{-j})$$

  .. in an alternating fashion until convergence

- $\mathbf{U}_{-i}$ denotes all of $\mathbf{U}$ except $\boldsymbol{u}_i$. $\mathbf{V}, \mathbf{U}_{-i}$ is the set of all other unknowns (fixed to a “current” value)

The Posterior

- Our target posterior distribution for this model will be

$$p(\mathbf{U},\mathbf{V}|\mathbf{R}) = \frac{p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})}{\int\int p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})d\mathbf{U}d\mathbf{V}} = \frac{\prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)}{\int\ldots\int \prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)d\boldsymbol{u}_1\ldots d\boldsymbol{u}_N d\boldsymbol{v}_1\ldots d\boldsymbol{v}_M}$$

- The denominator (and hence the posterior) is intractable! **Question: Why?**
- Therefore, the posterior must be approximated somehow
- One way to approximate is to compute **Conditional Posterior** (CP) over each unknown, e.g.,

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V},\mathbf{U}_{-i}) \quad \text{and} \quad p(\boldsymbol{v}_j|\mathbf{R},\mathbf{U},\mathbf{V}_{-j})$$

.. in an alternating fashion until convergence

- $\mathbf{U}_{-i}$ denotes all of $\mathbf{U}$ except $\boldsymbol{u}_i$. $\mathbf{V},\mathbf{U}_{-i}$ is the set of all other unknowns (fixed to a "current" value)
- $\mathbf{V}_{-j}$ denotes all of $\mathbf{V}$ except $\boldsymbol{v}_j$. $\mathbf{U},\mathbf{V}_{-j}$ is the set of all other unknowns (fixed to a "current" value)

The Posterior

- Our target posterior distribution for this model will be

$$p(\mathbf{U},\mathbf{V}|\mathbf{R}) = \frac{p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})}{\int\int p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})d\mathbf{U}d\mathbf{V}} = \frac{\prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)}{\int\ldots\int \prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j) d\boldsymbol{u}_1\ldots d\boldsymbol{u}_N d\boldsymbol{v}_1\ldots d\boldsymbol{v}_M}$$

- The denominator (and hence the posterior) is intractable! **Question: Why?**
- Therefore, the posterior must be approximated somehow
- One way to approximate is to compute **Conditional Posterior** (CP) over each unknown, e.g.,

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V},\mathbf{U}_{-i}) \quad \text{and} \quad p(\boldsymbol{v}_j|\mathbf{R},\mathbf{U},\mathbf{V}_{-j})$$

.. in an alternating fashion until convergence

- $\mathbf{U}_{-i}$ denotes all of $\mathbf{U}$ except $\boldsymbol{u}_i$. $\mathbf{V},\mathbf{U}_{-i}$ is the set of all other unknowns (fixed to a "current" value)
- $\mathbf{V}_{-j}$ denotes all of $\mathbf{V}$ except $\boldsymbol{v}_j$. $\mathbf{U},\mathbf{V}_{-j}$ is the set of all other unknowns (fixed to a "current" value)
- **Can we compute these CPs easily?**

The Posterior

- Our target posterior distribution for this model will be

$$p(\mathbf{U},\mathbf{V}|\mathbf{R}) = \frac{p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})}{\int\int p(\mathbf{R}|\mathbf{U},\mathbf{V})p(\mathbf{U},\mathbf{V})d\mathbf{U}d\mathbf{V}} = \frac{\prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)}{\int\dots\int \prod_{(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)\prod_i p(\boldsymbol{u}_i)\prod_j p(\boldsymbol{v}_j)d\boldsymbol{u}_1\dots d\boldsymbol{u}_N d\boldsymbol{v}_1\dots d\boldsymbol{v}_M}$$

- The denominator (and hence the posterior) is intractable! **Question: Why?**
- Therefore, the posterior must be approximated somehow
- One way to approximate is to compute **Conditional Posterior** (CP) over each unknown, e.g.,

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V},\mathbf{U}_{-i}) \quad \text{and} \quad p(\boldsymbol{v}_j|\mathbf{R},\mathbf{U},\mathbf{V}_{-j})$$

  .. in an alternating fashion until convergence

- $\mathbf{U}_{-i}$ denotes all of $\mathbf{U}$ except $\boldsymbol{u}_i$. $\mathbf{V},\mathbf{U}_{-i}$ is the set of all other unknowns (fixed to a "current" value)
- $\mathbf{V}_{-j}$ denotes all of $\mathbf{V}$ except $\boldsymbol{v}_j$. $\mathbf{U},\mathbf{V}_{-j}$ is the set of all other unknowns (fixed to a "current" value)
- **Can we compute these CPs easily?** Yes, if the model has "local conjugacy"

Local Conjugacy

- Conditional Posteriors (CP) are easy to compute if the model admits local conjugacy

Local Conjugacy

- Conditional Posteriors (CP) are easy to compute if the model admits local conjugacy
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$

Local Conjugacy

- Conditional Posteriors (CP) are easy to compute if the model admits local conjugacy
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose the overall posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (as is often the case!)

Local Conjugacy

- Conditional Posteriors (CP) are easy to compute if the model admits local conjugacy
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose the overall posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (as is often the case!)
- However suppose, given a "fixed" Θ_{-k}, we can compute the following CP tractably

$$p(\theta_k|\mathbf{X}, \Theta_{-k}) = \frac{p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)}{\int p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)d\theta_k}$$

Local Conjugacy

- Conditional Posteriors (CP) are easy to compute if the model admits local conjugacy
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose the overall posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (as is often the case!)
- However suppose, given a "fixed" Θ_{-k}, we can compute the following CP tractably

$$p(\theta_k|\mathbf{X}, \Theta_{-k}) = \frac{p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)}{\int p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)d\theta_k} \propto p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)$$

Local Conjugacy

- Conditional Posteriors (CP) are easy to compute if the model admits local conjugacy
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose the overall posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (as is often the case!)
- However suppose, given a "fixed" Θ_{-k}, we can compute the following CP tractably

$$p(\theta_k|\mathbf{X}, \Theta_{-k}) = \frac{p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)}{\int p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)d\theta_k} \propto p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)$$

.. which would be possible if $p(\mathbf{X}|\theta_k, \Theta_{-k})$ and $p(\theta_k)$ are conjugate to each other

Local Conjugacy

- Conditional Posteriors (CP) are easy to compute if the model admits local conjugacy
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose the overall posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (as is often the case!)
- However suppose, given a "fixed" Θ_{-k}, we can compute the following CP tractably

$$p(\theta_k|\mathbf{X}, \Theta_{-k}) = \frac{p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)}{\int p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)d\theta_k} \propto p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)$$

  .. which would be possible if $p(\mathbf{X}|\theta_k, \Theta_{-k})$ and $p(\theta_k)$ are conjugate to each other
- Such models are called "locally conjugate" models

Local Conjugacy

- Conditional Posteriors (CP) are easy to compute if the model admits local conjugacy
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose the overall posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (as is often the case!)
- However suppose, given a "fixed" Θ_{-k}, we can compute the following CP tractably

$$p(\theta_k|\mathbf{X}, \Theta_{-k}) = \frac{p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)}{\int p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)d\theta_k} \propto p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)$$

  .. which would be possible if $p(\mathbf{X}|\theta_k, \Theta_{-k})$ and $p(\theta_k)$ are conjugate to each other
- Such models are called "locally conjugate" models
- Important: In the above context, when considering the likelihood $p(\mathbf{X}|\theta_k, \Theta_{-k})$

Local Conjugacy

- Conditional Posteriors (CP) are easy to compute if the model admits local conjugacy
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose the overall posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (as is often the case!)
- However suppose, given a "fixed" Θ_{-k}, we can compute the following CP tractably

$$p(\theta_k|\mathbf{X}, \Theta_{-k}) = \frac{p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)}{\int p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)d\theta_k} \propto p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)$$

.. which would be possible if $p(\mathbf{X}|\theta_k, \Theta_{-k})$ and $p(\theta_k)$ are conjugate to each other

- Such models are called "locally conjugate" models
- Important: In the above context, when considering the likelihood $p(\mathbf{X}|\theta_k, \Theta_{-k})$
  - $\mathbf{X}$ actually refers to only that part of data $\mathbf{X}$ that depends on θ_k

Local Conjugacy

- Conditional Posteriors (CP) are easy to compute if the model admits local conjugacy
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose the overall posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (as is often the case!)
- However suppose, given a "fixed" Θ_{-k}, we can compute the following CP tractably

$$p(\theta_k|\mathbf{X}, \Theta_{-k}) = \frac{p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)}{\int p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)d\theta_k} \propto p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)$$

.. which would be possible if $p(\mathbf{X}|\theta_k, \Theta_{-k})$ and $p(\theta_k)$ are conjugate to each other

- Such models are called "locally conjugate" models
- Important: In the above context, when considering the likelihood $p(\mathbf{X}|\theta_k, \Theta_{-k})$
  - $\mathbf{X}$ actually refers to only that part of data $\mathbf{X}$ that depends on θ_k
  - Θ_{-k} refers to only those unknowns that "interact" with θ_k in generating that part of data

Local Conjugacy

- Conditional Posteriors (CP) are easy to compute if the model admits local conjugacy
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose the overall posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (as is often the case!)
- However suppose, given a "fixed" Θ_{-k}, we can compute the following CP tractably

$$p(\theta_k|\mathbf{X}, \Theta_{-k}) = \frac{p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)}{\int p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)d\theta_k} \propto p(\mathbf{X}|\theta_k, \Theta_{-k})p(\theta_k)$$

.. which would be possible if $p(\mathbf{X}|\theta_k, \Theta_{-k})$ and $p(\theta_k)$ are conjugate to each other

- Such models are called "locally conjugate" models
- Important: In the above context, when considering the likelihood $p(\mathbf{X}|\theta_k, \Theta_{-k})$
  - $\mathbf{X}$ actually refers to only that part of data $\mathbf{X}$ that depends on θ_k
  - Θ_{-k} refers to only those unknowns that "interact" with θ_k in generating that part of data
  - This will be more clear from the example of the Bayesian matrix factorization model

Representation of Posterior

- With the conditional posterior based approximation, the target posterior

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

.. is represented by several conditional posteriors $p(\theta_k|\mathbf{X}, \Theta_{-k}), \ k = 1, \ldots, K$

Representation of Posterior

- With the conditional posterior based approximation, the target posterior

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

.. is represented by several conditional posteriors $p(\theta_k|\mathbf{X}, \Theta_{-k})$, $k = 1, \ldots, K$

- Each of the conditional posterior is a distribution over one unknown θ_k, given all other unknowns

Representation of Posterior

- With the conditional posterior based approximation, the target posterior

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

.. is represented by several conditional posteriors $p(\theta_k|\mathbf{X}, \Theta_{-k})$, $k = 1, \ldots, K$

- Each of the conditional posterior is a distribution over one unknown θ_k, given all other unknowns
- One way to get the overall representation of the posterior can be can be using sampling based inference algorithms like Gibbs sampling or MCMC, or Variational Inference (more on this later)

Bayesian Matrix Factorization

- The BMF model with Gaussian likelihood and Gaussian prior has local conjugacy

Bayesian Matrix Factorization

- The BMF model with Gaussian likelihood and Gaussian prior has local conjugacy
- To see this, note that the conditional posterior for user latent factor $\boldsymbol{u}_i$ is

Bayesian Matrix Factorization

- The BMF model with Gaussian likelihood and Gaussian prior has local conjugacy
- To see this, note that the conditional posterior for user latent factor $\boldsymbol{u}_i$ is

$$p(\boldsymbol{u}_i|\mathbf{R}, \mathbf{V}, \mathbf{U}_{-i}) \propto \prod_{j:(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j) p(\boldsymbol{u}_i)$$

Bayesian Matrix Factorization

- The BMF model with Gaussian likelihood and Gaussian prior has local conjugacy
- To see this, note that the conditional posterior for user latent factor $\boldsymbol{u}_i$ is

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V},\mathbf{U}_{-i}) \propto \prod_{j:(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)p(\boldsymbol{u}_i)$$

- Note: the posterior of $\boldsymbol{u}_i$ doesn't actually depend on $\mathbf{U}_{-i}$ and rows of $\mathbf{R}$ except row i
- After substituting the likelihood and prior (both Gaussians), the conditional posterior of $\boldsymbol{u}_i$ is

Bayesian Matrix Factorization

- The BMF model with Gaussian likelihood and Gaussian prior has local conjugacy
- To see this, note that the conditional posterior for user latent factor $\boldsymbol{u}_i$ is

$$p(\boldsymbol{u}_i|\mathbf{R}, \mathbf{V}, \mathbf{U}_{-i}) \propto \prod_{j:(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j)p(\boldsymbol{u}_i)$$

- Note: the posterior of $\boldsymbol{u}_i$ doesn't actually depend on $\mathbf{U}_{-i}$ and rows of $\mathbf{R}$ except row i
- After substituting the likelihood and prior (both Gaussians), the conditional posterior of $\boldsymbol{u}_i$ is

$$p(\boldsymbol{u}_i|\mathbf{R}, \mathbf{V}) \propto \prod_{j:(i,j)\in\Omega} \mathcal{N}(r_{ij}|\boldsymbol{u}_i^\top \boldsymbol{v}_j, \beta^{-1})\mathcal{N}(\boldsymbol{u}_i|\mathbf{0}, \lambda_u^{-1}\mathbf{I}_K)$$

Bayesian Matrix Factorization

- The BMF model with Gaussian likelihood and Gaussian prior has local conjugacy
- To see this, note that the conditional posterior for user latent factor $\boldsymbol{u}_i$ is

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V},\mathbf{U}_{-i}) \propto \prod_{j:(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)p(\boldsymbol{u}_i)$$

- Note: the posterior of $\boldsymbol{u}_i$ doesn't actually depend on $\mathbf{U}_{-i}$ and rows of $\mathbf{R}$ except row i
- After substituting the likelihood and prior (both Gaussians), the conditional posterior of $\boldsymbol{u}_i$ is

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V}) \propto \prod_{j:(i,j)\in\Omega} \mathcal{N}(r_{ij}|\boldsymbol{u}_i^\top\boldsymbol{v}_j,\beta^{-1})\mathcal{N}(\boldsymbol{u}_i|\mathbf{0},\lambda_u^{-1}\mathbf{I}_K)$$

- Since $\mathbf{V}$ fixed (remember we are computing conditional posteriors alternating fashion), the likelihood and prior are conjugate. This is just like Bayesian linear regression

Bayesian Matrix Factorization

- The BMF model with Gaussian likelihood and Gaussian prior has local conjugacy
- To see this, note that the conditional posterior for user latent factor $\boldsymbol{u}_i$ is

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V},\mathbf{U}_{-i}) \propto \prod_{j:(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)p(\boldsymbol{u}_i)$$

- Note: the posterior of $\boldsymbol{u}_i$ doesn't actually depend on $\mathbf{U}_{-i}$ and rows of $\mathbf{R}$ except row i
- After substituting the likelihood and prior (both Gaussians), the conditional posterior of $\boldsymbol{u}_i$ is

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V}) \propto \prod_{j:(i,j)\in\Omega} \mathcal{N}(r_{ij}|\boldsymbol{u}_i^\top\boldsymbol{v}_j,\beta^{-1})\mathcal{N}(\boldsymbol{u}_i|\mathbf{0},\lambda_u^{-1}\mathbf{I}_K)$$

- Since $\mathbf{V}$ fixed (remember we are computing conditional posteriors alternating fashion), the likelihood and prior are conjugate. This is just like Bayesian linear regression
  - Linear regression analogy: $\{\boldsymbol{v}_j\}_{j:(i,j)\in\Omega}$: inputs, $\{r_{ij}\}_{j:(i,j)\in\Omega}$: responses, $\boldsymbol{u}_i$: unknown weight vector

Bayesian Matrix Factorization

- The BMF model with Gaussian likelihood and Gaussian prior has local conjugacy
- To see this, note that the conditional posterior for user latent factor $\boldsymbol{u}_i$ is

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V},\mathbf{U}_{-i}) \propto \prod_{j:(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)p(\boldsymbol{u}_i)$$

- Note: the posterior of $\boldsymbol{u}_i$ doesn't actually depend on $\mathbf{U}_{-i}$ and rows of $\mathbf{R}$ except row i
- After substituting the likelihood and prior (both Gaussians), the conditional posterior of $\boldsymbol{u}_i$ is

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V}) \propto \prod_{j:(i,j)\in\Omega} \mathcal{N}(r_{ij}|\boldsymbol{u}_i^\top\boldsymbol{v}_j,\beta^{-1})\mathcal{N}(\boldsymbol{u}_i|\mathbf{0},\lambda_u^{-1}\mathbf{I}_K)$$

- Since $\mathbf{V}$ fixed (remember we are computing conditional posteriors alternating fashion), the likelihood and prior are conjugate. This is just like Bayesian linear regression
  - Linear regression analogy: $\{\boldsymbol{v}_j\}_{j:(i,j)\in\Omega}$: inputs, $\{r_{ij}\}_{j:(i,j)\in\Omega}$: responses, $\boldsymbol{u}_i$: unknown weight vector
- Likewise, the conditional posterior of $\boldsymbol{v}_j$ will be

Bayesian Matrix Factorization

- The BMF model with Gaussian likelihood and Gaussian prior has local conjugacy
- To see this, note that the conditional posterior for user latent factor $\boldsymbol{u}_i$ is

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V},\mathbf{U}_{-i}) \propto \prod_{j:(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)p(\boldsymbol{u}_i)$$

- Note: the posterior of $\boldsymbol{u}_i$ doesn't actually depend on $\mathbf{U}_{-i}$ and rows of $\mathbf{R}$ except row i
- After substituting the likelihood and prior (both Gaussians), the conditional posterior of $\boldsymbol{u}_i$ is

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V}) \propto \prod_{j:(i,j)\in\Omega} \mathcal{N}(r_{ij}|\boldsymbol{u}_i^\top\boldsymbol{v}_j,\beta^{-1})\mathcal{N}(\boldsymbol{u}_i|\mathbf{0},\lambda_u^{-1}\mathbf{I}_K)$$

- Since $\mathbf{V}$ fixed (remember we are computing conditional posteriors alternating fashion), the likelihood and prior are conjugate. This is just like Bayesian linear regression
  - Linear regression analogy: $\{\boldsymbol{v}_j\}_{j:(i,j)\in\Omega}$: inputs, $\{r_{ij}\}_{j:(i,j)\in\Omega}$: responses, $\boldsymbol{u}_i$: unknown weight vector
- Likewise, the conditional posterior of $\boldsymbol{v}_j$ will be

$$p(\boldsymbol{v}_j|\mathbf{R},\mathbf{U}) \propto \prod_{i:(i,j)\in\Omega} \mathcal{N}(r_{ij}|\boldsymbol{u}_i^\top\boldsymbol{v}_j,\beta^{-1})\mathcal{N}(\boldsymbol{v}_j|\mathbf{0},\lambda_v^{-1}\mathbf{I}_K)$$

Bayesian Matrix Factorization

- The BMF model with Gaussian likelihood and Gaussian prior has local conjugacy
- To see this, note that the conditional posterior for user latent factor $\boldsymbol{u}_i$ is

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V},\mathbf{U}_{-i}) \propto \prod_{j:(i,j)\in\Omega} p(r_{ij}|\boldsymbol{u}_i,\boldsymbol{v}_j)p(\boldsymbol{u}_i)$$

- Note: the posterior of $\boldsymbol{u}_i$ doesn't actually depend on $\mathbf{U}_{-i}$ and rows of $\mathbf{R}$ except row i
- After substituting the likelihood and prior (both Gaussians), the conditional posterior of $\boldsymbol{u}_i$ is

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V}) \propto \prod_{j:(i,j)\in\Omega} \mathcal{N}(r_{ij}|\boldsymbol{u}_i^\top\boldsymbol{v}_j,\beta^{-1})\mathcal{N}(\boldsymbol{u}_i|\mathbf{0},\lambda_u^{-1}\mathbf{I}_K)$$

- Since $\mathbf{V}$ fixed (remember we are computing conditional posteriors alternating fashion), the likelihood and prior are conjugate. This is just like Bayesian linear regression
  - Linear regression analogy: $\{\boldsymbol{v}_j\}_{j:(i,j)\in\Omega}$: inputs, $\{r_{ij}\}_{j:(i,j)\in\Omega}$: responses, $\boldsymbol{u}_i$: unknown weight vector
- Likewise, the conditional posterior of $\boldsymbol{v}_j$ will be

$$p(\boldsymbol{v}_j|\mathbf{R},\mathbf{U}) \propto \prod_{i:(i,j)\in\Omega} \mathcal{N}(r_{ij}|\boldsymbol{u}_i^\top\boldsymbol{v}_j,\beta^{-1})\mathcal{N}(\boldsymbol{v}_j|\mathbf{0},\lambda_v^{-1}\mathbf{I}_K)$$

.. like Bayesian lin. reg. with $\{\boldsymbol{u}_i\}_{i:(i,j)\in\Omega}$: inputs, $\{r_{ij}\}_{i:(i,j)\in\Omega}$: responses, $\boldsymbol{v}_j$: unknown weight vec

Bayesian Matrix Factorization

- The conditional posteriors will have forms similar to solution of Bayesian linear regression

Bayesian Matrix Factorization

- The conditional posteriors will have forms similar to solution of Bayesian linear regression
- For each $\boldsymbol{u}_i$, its conditional posterior, given $\mathbf{V}$ and ratings

$$p(\boldsymbol{u}_i|\mathbf{R}, \mathbf{V}) = \mathcal{N}(\boldsymbol{u}_i|\boldsymbol{\mu}_{u_i}, \boldsymbol{\Sigma}_{u_i})$$

Bayesian Matrix Factorization

- The conditional posteriors will have forms similar to solution of Bayesian linear regression
- For each $\boldsymbol{u}_i$, its conditional posterior, given $\mathbf{V}$ and ratings

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V}) = \mathcal{N}(\boldsymbol{u}_i|\boldsymbol{\mu}_{u_i},\boldsymbol{\Sigma}_{u_i})$$

where $\boldsymbol{\Sigma}_{u_i} = (\lambda_u \mathbf{I} + \beta \sum_{j:(i,j)\in\Omega} \boldsymbol{v}_j \boldsymbol{v}_j^\top)^{-1}$ and $\boldsymbol{\mu}_{u_i} = \boldsymbol{\Sigma}_{u_i}(\beta \sum_{j:(i,j)\in\Omega} r_{ij}\boldsymbol{v}_j)$

Bayesian Matrix Factorization

- The conditional posteriors will have forms similar to solution of Bayesian linear regression
- For each $\boldsymbol{u}_i$, its conditional posterior, given $\mathbf{V}$ and ratings

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V}) = \mathcal{N}(\boldsymbol{u}_i|\boldsymbol{\mu}_{u_i},\boldsymbol{\Sigma}_{u_i})$$

where $\boldsymbol{\Sigma}_{u_i} = (\lambda_u \mathbf{I} + \beta \sum_{j:(i,j)\in\Omega} \boldsymbol{v}_j \boldsymbol{v}_j^\top)^{-1}$and $\boldsymbol{\mu}_{u_i} = \boldsymbol{\Sigma}_{u_i}(\beta \sum_{j:(i,j)\in\Omega} r_{ij}\boldsymbol{v}_j)$

- For each $\boldsymbol{v}_j$, its conditional posterior, given $\mathbf{U}$ and ratings

$$p(\boldsymbol{v}_j|\mathbf{R},\mathbf{U}) = \mathcal{N}(\boldsymbol{v}_j|\boldsymbol{\mu}_{v_j},\boldsymbol{\Sigma}_{v_j})$$

where $\boldsymbol{\Sigma}_{v_j} = (\lambda_v \mathbf{I} + \beta \sum_{i:(i,j)\in\Omega} \boldsymbol{u}_i \boldsymbol{u}_i^\top)^{-1}$ and $\boldsymbol{\mu}_{v_j} = \boldsymbol{\Sigma}_{v_j}(\beta \sum_{i:(i,j)\in\Omega} r_{ij}\boldsymbol{u}_i)$

Bayesian Matrix Factorization

- The conditional posteriors will have forms similar to solution of Bayesian linear regression
- For each $\boldsymbol{u}_i$, its conditional posterior, given $\mathbf{V}$ and ratings

$$p(\boldsymbol{u}_i|\mathbf{R}, \mathbf{V}) = \mathcal{N}(\boldsymbol{u}_i|\boldsymbol{\mu}_{u_i}, \boldsymbol{\Sigma}_{u_i})$$

  where $\boldsymbol{\Sigma}_{u_i} = (\lambda_u \mathbf{I} + \beta \sum_{j:(i,j)\in\Omega} \boldsymbol{v}_j \boldsymbol{v}_j^\top)^{-1}$ and $\boldsymbol{\mu}_{u_i} = \boldsymbol{\Sigma}_{u_i}(\beta \sum_{j:(i,j)\in\Omega} r_{ij} \boldsymbol{v}_j)$
- For each $\boldsymbol{v}_j$, its conditional posterior, given $\mathbf{U}$ and ratings

$$p(\boldsymbol{v}_j|\mathbf{R}, \mathbf{U}) = \mathcal{N}(\boldsymbol{v}_j|\boldsymbol{\mu}_{v_j}, \boldsymbol{\Sigma}_{v_j})$$

  where $\boldsymbol{\Sigma}_{v_j} = (\lambda_v \mathbf{I} + \beta \sum_{i:(i,j)\in\Omega} \boldsymbol{u}_i \boldsymbol{u}_i^\top)^{-1}$ and $\boldsymbol{\mu}_{v_j} = \boldsymbol{\Sigma}_{v_j}(\beta \sum_{i:(i,j)\in\Omega} r_{ij} \boldsymbol{u}_i)$
- These conditional posteriors can be updated in an alternating fashion until convergence

Bayesian Matrix Factorization

- The conditional posteriors will have forms similar to solution of Bayesian linear regression
- For each $\boldsymbol{u}_i$, its conditional posterior, given $\mathbf{V}$ and ratings

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V}) = \mathcal{N}(\boldsymbol{u}_i|\boldsymbol{\mu}_{u_i},\boldsymbol{\Sigma}_{u_i})$$

  where $\boldsymbol{\Sigma}_{u_i} = (\lambda_u \mathbf{I} + \beta \sum_{j:(i,j)\in\Omega} \boldsymbol{v}_j \boldsymbol{v}_j^\top)^{-1}$and $\boldsymbol{\mu}_{u_i} = \boldsymbol{\Sigma}_{u_i}(\beta \sum_{j:(i,j)\in\Omega} r_{ij}\boldsymbol{v}_j)$

- For each $\boldsymbol{v}_j$, its conditional posterior, given $\mathbf{U}$ and ratings

$$p(\boldsymbol{v}_j|\mathbf{R},\mathbf{U}) = \mathcal{N}(\boldsymbol{v}_j|\boldsymbol{\mu}_{v_j},\boldsymbol{\Sigma}_{v_j})$$

  where $\boldsymbol{\Sigma}_{v_j} = (\lambda_v \mathbf{I} + \beta \sum_{i:(i,j)\in\Omega} \boldsymbol{u}_i \boldsymbol{u}_i^\top)^{-1}$ and $\boldsymbol{\mu}_{v_j} = \boldsymbol{\Sigma}_{v_j}(\beta \sum_{i:(i,j)\in\Omega} r_{ij}\boldsymbol{u}_i)$

- These conditional posteriors can be updated in an alternating fashion until convergence
  - This can be be implemented using a Gibbs sampler (more on this when we discuss MCMC methods)

Bayesian Matrix Factorization

- The conditional posteriors will have forms similar to solution of Bayesian linear regression
- For each $\boldsymbol{u}_i$, its conditional posterior, given $\mathbf{V}$ and ratings

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V}) = \mathcal{N}(\boldsymbol{u}_i|\boldsymbol{\mu}_{u_i},\boldsymbol{\Sigma}_{u_i})$$

  where $\boldsymbol{\Sigma}_{u_i} = (\lambda_u \mathbf{I} + \beta \sum_{j:(i,j)\in\Omega} \boldsymbol{v}_j \boldsymbol{v}_j^\top)^{-1}$and $\boldsymbol{\mu}_{u_i} = \boldsymbol{\Sigma}_{u_i}(\beta \sum_{j:(i,j)\in\Omega} r_{ij}\boldsymbol{v}_j)$

- For each $\boldsymbol{v}_j$, its conditional posterior, given $\mathbf{U}$ and ratings

$$p(\boldsymbol{v}_j|\mathbf{R},\mathbf{U}) = \mathcal{N}(\boldsymbol{v}_j|\boldsymbol{\mu}_{v_j},\boldsymbol{\Sigma}_{v_j})$$

  where $\boldsymbol{\Sigma}_{v_j} = (\lambda_v \mathbf{I} + \beta \sum_{i:(i,j)\in\Omega} \boldsymbol{u}_i \boldsymbol{u}_i^\top)^{-1}$ and $\boldsymbol{\mu}_{v_j} = \boldsymbol{\Sigma}_{v_j}(\beta \sum_{i:(i,j)\in\Omega} r_{ij}\boldsymbol{u}_i)$

- These conditional posteriors can be updated in an alternating fashion until convergence
  - This can be be implemented using a Gibbs sampler (more on this when we discuss MCMC methods)
  - Note: Hyperparameters can also be inferred by computing their conditional posteriors (also see "Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo" by Salakhutdinov and Mnih (2008))

Bayesian Matrix Factorization

- The conditional posteriors will have forms similar to solution of Bayesian linear regression
- For each $\boldsymbol{u}_i$, its conditional posterior, given $\mathbf{V}$ and ratings

$$p(\boldsymbol{u}_i|\mathbf{R},\mathbf{V}) = \mathcal{N}(\boldsymbol{u}_i|\boldsymbol{\mu}_{u_i},\boldsymbol{\Sigma}_{u_i})$$

  where $\boldsymbol{\Sigma}_{u_i} = (\lambda_u \mathbf{I} + \beta \sum_{j:(i,j)\in\Omega} \boldsymbol{v}_j \boldsymbol{v}_j^\top)^{-1}$and $\boldsymbol{\mu}_{u_i} = \boldsymbol{\Sigma}_{u_i}(\beta \sum_{j:(i,j)\in\Omega} r_{ij}\boldsymbol{v}_j)$

- For each $\boldsymbol{v}_j$, its conditional posterior, given $\mathbf{U}$ and ratings

$$p(\boldsymbol{v}_j|\mathbf{R},\mathbf{U}) = \mathcal{N}(\boldsymbol{v}_j|\boldsymbol{\mu}_{v_j},\boldsymbol{\Sigma}_{v_j})$$

  where $\boldsymbol{\Sigma}_{v_j} = (\lambda_v \mathbf{I} + \beta \sum_{i:(i,j)\in\Omega} \boldsymbol{u}_i \boldsymbol{u}_i^\top)^{-1}$ and $\boldsymbol{\mu}_{v_j} = \boldsymbol{\Sigma}_{v_j}(\beta \sum_{i:(i,j)\in\Omega} r_{ij}\boldsymbol{u}_i)$

- These conditional posteriors can be updated in an alternating fashion until convergence
  - This can be be implemented using a Gibbs sampler (more on this when we discuss MCMC methods)
  - Note: Hyperparameters can also be inferred by computing their conditional posteriors (also see "Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo" by Salakhutdinov and Mnih (2008))
  - Can extend Gaussian BMF easily to other exp. family distr. while maintaining local conjugacy

Summary and Some Comments

- Inference in complex probabilistic models can often be performed rather easily if the models have the local conjugacy property

Summary and Some Comments

- Inference in complex probabilistic models can often be performed rather easily if the models have the local conjugacy property
- It therefore helps to choose the likelihood model and priors on each param as exp. family distr.

Summary and Some Comments

- Inference in complex probabilistic models can often be performed rather easily if the models have the local conjugacy property
- It therefore helps to choose the likelihood model and priors on each param as exp. family distr.
  - Even if we can't get a globally conjugacy model, we can still get a model with local conjugacy

Summary and Some Comments

- Inference in complex probabilistic models can often be performed rather easily if the models have the local conjugacy property
- It therefore helps to choose the likelihood model and priors on each param as exp. family distr.
  - Even if we can't get a globally conjugacy model, we can still get a model with local conjugacy
- Local conjugacy allows computing conditional posteriors that are needed in inference algos like EM, Gibbs sampling, MCMC, variational inference, etc.

Summary and Some Comments

- Inference in complex probabilistic models can often be performed rather easily if the models have the local conjugacy property
- It therefore helps to choose the likelihood model and priors on each param as exp. family distr.
  - Even if we can't get a globally conjugacy model, we can still get a model with local conjugacy
- Local conjugacy allows computing conditional posteriors that are needed in inference algos like EM, Gibbs sampling, MCMC, variational inference, etc.
- The idea is similar in spirit to alternating optimization techniques (and methods such as EM)

Summary and Some Comments

- Inference in complex probabilistic models can often be performed rather easily if the models have the local conjugacy property
- It therefore helps to choose the likelihood model and priors on each param as exp. family distr.
  - Even if we can't get a globally conjugacy model, we can still get a model with local conjugacy
- Local conjugacy allows computing conditional posteriors that are needed in inference algos like EM, Gibbs sampling, MCMC, variational inference, etc.
- The idea is similar in spirit to alternating optimization techniques (and methods such as EM)
  - Local conjugacy makes each "sub-problem" simple!

Summary and Some Comments

- Inference in complex probabilistic models can often be performed rather easily if the models have the local conjugacy property
- It therefore helps to choose the likelihood model and priors on each param as exp. family distr.
  - Even if we can't get a globally conjugacy model, we can still get a model with local conjugacy
- Local conjugacy allows computing conditional posteriors that are needed in inference algos like EM, Gibbs sampling, MCMC, variational inference, etc.
- The idea is similar in spirit to alternating optimization techniques (and methods such as EM)
  - Local conjugacy makes each "sub-problem" simple!
- Exercise: Use the idea of infer (μ, Σ) given N observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ from $\mathcal{N}(\mu, \Sigma)$

Summary and Some Comments

- Inference in complex probabilistic models can often be performed rather easily if the models have the local conjugacy property
- It therefore helps to choose the likelihood model and priors on each param as exp. family distr.
  - Even if we can't get a globally conjugacy model, we can still get a model with local conjugacy
- Local conjugacy allows computing conditional posteriors that are needed in inference algos like EM, Gibbs sampling, MCMC, variational inference, etc.
- The idea is similar in spirit to alternating optimization techniques (and methods such as EM)
  - Local conjugacy makes each "sub-problem" simple!
- Exercise: Use the idea of infer (μ, Σ) given N observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ from $\mathcal{N}(\mu, \Sigma)$
  - What choice of priors would make this model locally conjugate?

Summary and Some Comments

- Inference in complex probabilistic models can often be performed rather easily if the models have the local conjugacy property
- It therefore helps to choose the likelihood model and priors on each param as exp. family distr.
  - Even if we can't get a globally conjugacy model, we can still get a model with local conjugacy
- Local conjugacy allows computing conditional posteriors that are needed in inference algos like EM, Gibbs sampling, MCMC, variational inference, etc.
- The idea is similar in spirit to alternating optimization techniques (and methods such as EM)
  - Local conjugacy makes each "sub-problem" simple!
- Exercise: Use the idea of infer (μ, Σ) given N observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ from $\mathcal{N}(\mu, \Sigma)$
  - What choice of priors would make this model locally conjugate?
  - What would be the CPs in this problem?