Inference in Multiparameter Models, Conditional Posterior, Local Conjugacy

Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

Feb 4, 2019

Today's Plan

- Foray into models with several parameters
- Goal will be to infer the posterior over all of them (not posterior for some, MLE-II for others)
- Idea of conditional/local posteriors in such problems
- Local conjugacy (which helps in computing conditional posteriors)
- Gibbs sampling (an algorithm that infer the joint posterior via conditional posteriors)
- An example: Bayesian matrix factorization (model with many parameters)
- Note: Conditional/local posterior, local conjugacy, etc are important ideas (will appear in many inference algorithms that we will see later)

Moving Beyond Simple Models..

- So far we've usually seen models with one "main" parameter and maybe a few hyperparams

Moving Beyond Simple Models..

- So far we've usually seen models with one "main" parameter and maybe a few hyperparams, e.g.,
  - Given data assumed to be from a Gaussian, infer the mean assuming variance known (or vice-versa)

Moving Beyond Simple Models..

- So far we've usually seen models with one "main" parameter and maybe a few hyperparams, e.g.,
  - Given data assumed to be from a Gaussian, infer the mean assuming variance known (or vice-versa)
  - Bayesian linear regression with weight vector $\boldsymbol{w}$ and noise/prior precision hyperparams β, λ

Moving Beyond Simple Models..

- So far we've usually seen models with one "main" parameter and maybe a few hyperparams, e.g.,
  - Given data assumed to be from a Gaussian, infer the mean assuming variance known (or vice-versa)
  - Bayesian linear regression with weight vector $\boldsymbol{w}$ and noise/prior precision hyperparams β, λ
  - GP regression with one function to be learned

Moving Beyond Simple Models..

- So far we've usually seen models with one "main" parameter and maybe a few hyperparams, e.g.,
  - Given data assumed to be from a Gaussian, infer the mean assuming variance known (or vice-versa)
  - Bayesian linear regression with weight vector $\boldsymbol{w}$ and noise/prior precision hyperparams β, λ
  - GP regression with one function to be learned
- Easy posterior inference if the likelihood and prior are conjugate to each other

Moving Beyond Simple Models..

- So far we've usually seen models with one "main" parameter and maybe a few hyperparams, e.g.,
  - Given data assumed to be from a Gaussian, infer the mean assuming variance known (or vice-versa)
  - Bayesian linear regression with weight vector $\boldsymbol{w}$ and noise/prior precision hyperparams β, λ
  - GP regression with one function to be learned
- Easy posterior inference if the likelihood and prior are conjugate to each other
  - Otherwise have to approx. the posterior (e.g., Laplace approx. - recall Bayesian logistic regression)

Moving Beyond Simple Models..

- So far we've usually seen models with one "main" parameter and maybe a few hyperparams, e.g.,
  - Given data assumed to be from a Gaussian, infer the mean assuming variance known (or vice-versa)
  - Bayesian linear regression with weight vector $\boldsymbol{w}$ and noise/prior precision hyperparams β, λ
  - GP regression with one function to be learned
- Easy posterior inference if the likelihood and prior are conjugate to each other
  - Otherwise have to approx. the posterior (e.g., Laplace approx. - recall Bayesian logistic regression)
- Hyperparams, if desired, can be also estimated via MLE-II

Moving Beyond Simple Models..

- So far we've usually seen models with one "main" parameter and maybe a few hyperparams, e.g.,
  - Given data assumed to be from a Gaussian, infer the mean assuming variance known (or vice-versa)
  - Bayesian linear regression with weight vector $\boldsymbol{w}$ and noise/prior precision hyperparams β, λ
  - GP regression with one function to be learned
- Easy posterior inference if the likelihood and prior are conjugate to each other
  - Otherwise have to approx. the posterior (e.g., Laplace approx. - recall Bayesian logistic regression)
- Hyperparams, if desired, can be also estimated via MLE-II
  - Note however that MLE-II would only give a point estimate of hyperparams

Moving Beyond Simple Models..

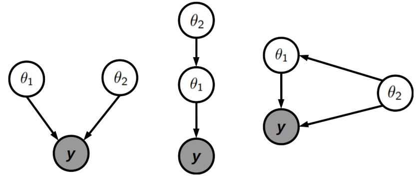
- So far we've usually seen models with one "main" parameter and maybe a few hyperparams, e.g.,
  - Given data assumed to be from a Gaussian, infer the mean assuming variance known (or vice-versa)
  - Bayesian linear regression with weight vector $\boldsymbol{w}$ and noise/prior precision hyperparams β, λ
  - GP regression with one function to be learned
- Easy posterior inference if the likelihood and prior are conjugate to each other
  - Otherwise have to approx. the posterior (e.g., Laplace approx. - recall Bayesian logistic regression)
- Hyperparams, if desired, can be also estimated via MLE-II
  - Note however that MLE-II would only give a point estimate of hyperparams
- What if we have a model with lots of parames/hyperparams and want posteriors over all of those?

Moving Beyond Simple Models..

- So far we've usually seen models with one "main" parameter and maybe a few hyperparams, e.g.,
  - Given data assumed to be from a Gaussian, infer the mean assuming variance known (or vice-versa)
  - Bayesian linear regression with weight vector $\boldsymbol{w}$ and noise/prior precision hyperparams β, λ
  - GP regression with one function to be learned
- Easy posterior inference if the likelihood and prior are conjugate to each other
  - Otherwise have to approx. the posterior (e.g., Laplace approx. - recall Bayesian logistic regression)
- Hyperparams, if desired, can be also estimated via MLE-II
  - Note however that MLE-II would only give a point estimate of hyperparams
- What if we have a model with lots of parames/hyperparams and want posteriors over all of those?
  - Intractable in general but today we will look at a way of doing approx. inference in such models

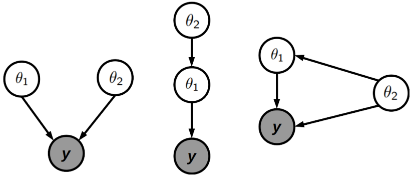
Multiparameter Models

- Multiparameter models consist of two or more unknowns, say θ_1 and θ_2
- Given data $\boldsymbol{y}$, some examples for the simple two parameter case



Multiparameter Models

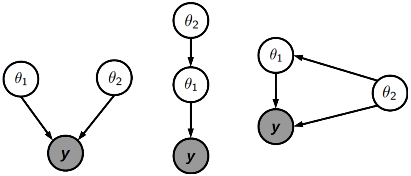
- Multiparameter models consist of two or more unknowns, say θ_1 and θ_2
- Given data $\boldsymbol{y}$, some examples for the simple two parameter case



- Assume the likelihood model to be of the form $p(\boldsymbol{y}|\theta_1, \theta_2)$ (e.g., case 1 and 3 above)

Multiparameter Models

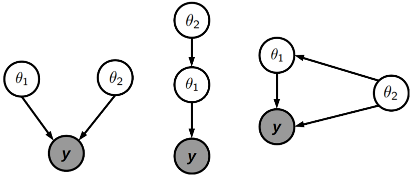
- Multiparameter models consist of two or more unknowns, say θ_1 and θ_2
- Given data $\boldsymbol{y}$, some examples for the simple two parameter case



- Assume the likelihood model to be of the form $p(\boldsymbol{y}|\theta_1, \theta_2)$ (e.g., case 1 and 3 above)
- Assume a joint prior distribution $p(\theta_1, \theta_2)$

Multiparameter Models

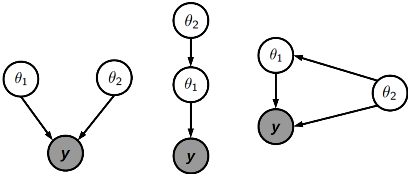
- Multiparameter models consist of two or more unknowns, say θ_1 and θ_2
- Given data $\boldsymbol{y}$, some examples for the simple two parameter case



- Assume the likelihood model to be of the form $p(\boldsymbol{y}|\theta_1, \theta_2)$ (e.g., case 1 and 3 above)
- Assume a joint prior distribution $p(\theta_1, \theta_2)$
- The joint posterior $p(\theta_1, \theta_2|\boldsymbol{y}) \propto p(\boldsymbol{y}|\theta_1, \theta_2)p(\theta_1, \theta_2)$

Multiparameter Models

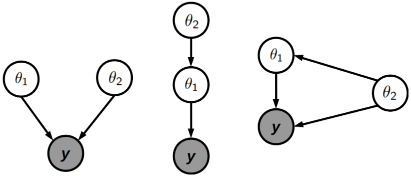
- Multiparameter models consist of two or more unknowns, say θ_1 and θ_2
- Given data $\boldsymbol{y}$, some examples for the simple two parameter case



- Assume the likelihood model to be of the form $p(\boldsymbol{y}|\theta_1, \theta_2)$ (e.g., case 1 and 3 above)
- Assume a joint prior distribution $p(\theta_1, \theta_2)$
- The joint posterior $p(\theta_1, \theta_2|\boldsymbol{y}) \propto p(\boldsymbol{y}|\theta_1, \theta_2)p(\theta_1, \theta_2)$
  - Easy the joint prior is conjugate to the likelihood (e.g., NIW prior for Gaussian likelihood)

Multiparameter Models

- Multiparameter models consist of two or more unknowns, say θ_1 and θ_2
- Given data $\boldsymbol{y}$, some examples for the simple two parameter case



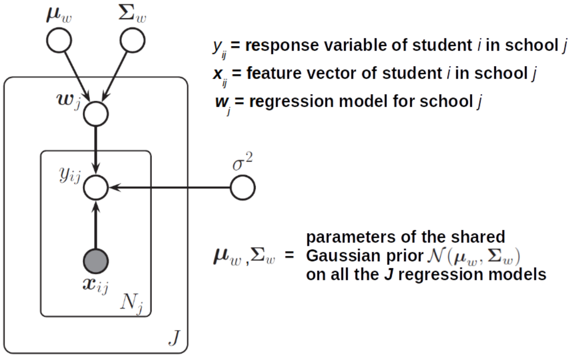
- Assume the likelihood model to be of the form $p(\boldsymbol{y}|\theta_1, \theta_2)$ (e.g., case 1 and 3 above)
- Assume a joint prior distribution $p(\theta_1, \theta_2)$
- The joint posterior $p(\theta_1, \theta_2|\boldsymbol{y}) \propto p(\boldsymbol{y}|\theta_1, \theta_2)p(\theta_1, \theta_2)$
  - Easy the joint prior is conjugate to the likelihood (e.g., NIW prior for Gaussian likelihood)
  - Otherwise needs more work, e.g., MLE-II, MCMC, VB, etc. (already saw MLE-II, will see more later)

Multiparameter Models: Some Examples

- Multiparameter models arise in many situations, e.g.,
  - Probabilistic models with unknown hyperparameters (e.g., Bayesian linear regression we just saw)

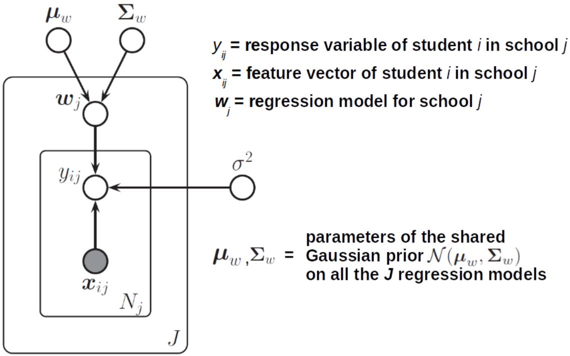
Multiparameter Models: Some Examples

- Multiparameter models arise in many situations, e.g.,
  - Probabilistic models with unknown hyperparameters (e.g., Bayesian linear regression we just saw)
  - Joint analysis of data from multiple (and possibly related) groups: Hierarchical models



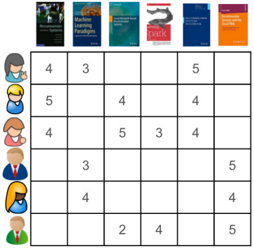
Multiparameter Models: Some Examples

- Multiparameter models arise in many situations, e.g.,
  - Probabilistic models with unknown hyperparameters (e.g., Bayesian linear regression we just saw)
  - Joint analysis of data from multiple (and possibly related) groups: Hierarchical models



- .. and in fact, pretty much in any non-toy example of probablistic model :)

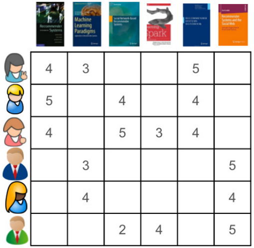
Another Example Problem: Matrix Factorization/Completion



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

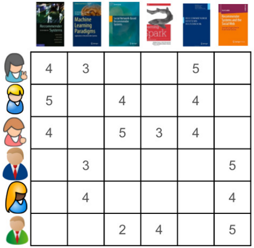
Another Example Problem: Matrix Factorization/Completion



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)

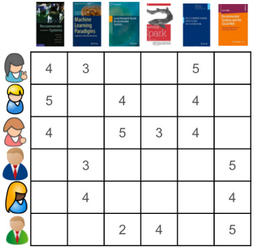
Another Example Problem: Matrix Factorization/Completion



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 $i = 1, \dots, N$ on $j = 1, \dots, M$ 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

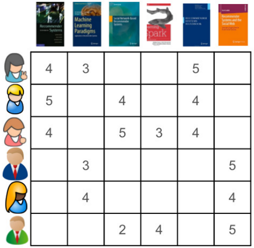
Another Example Problem: Matrix Factorization/Completion



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 $i = 1, \dots, N$ on $j = 1, \dots, M$ 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., $|\Omega| \ll NM$

Another Example Problem: Matrix Factorization/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

- Let's call the full matrix $\mathbf{R}$ and assume it to be an approximately low-rank matrix

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

Matrix Completion via Matrix Factorization

- Let's call the full matrix $\mathbf{R}$ and assume it to be an approximately low-rank matrix

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

- $\mathbf{U} = [\boldsymbol{u}_1 \ldots \boldsymbol{u}_N]^\top$ 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

Matrix Completion via Matrix Factorization

- Let's call the full matrix $\mathbf{R}$ and assume it to be an approximately low-rank matrix

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

- $\mathbf{U} = [\boldsymbol{u}_1 \dots \boldsymbol{u}_N]^\top$ 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

Matrix Completion via Matrix Factorization

- Let's call the full matrix $\mathbf{R}$ and assume it to be an approximately low-rank matrix

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

- $\mathbf{U} = [\boldsymbol{u}_1 \dots \boldsymbol{u}_N]^\top$ 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)

Matrix Completion via Matrix Factorization

- Let's call the full matrix $\mathbf{R}$ and assume it to be an approximately low-rank matrix

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

- $\mathbf{U} = [\boldsymbol{u}_1 \dots \boldsymbol{u}_N]^\top$ 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)
- We can write each element of matrix $\mathbf{R}$ as

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

Matrix Completion via Matrix Factorization

- Let's call the full matrix $\mathbf{R}$ and assume it to be an approximately low-rank matrix

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

- $\mathbf{U} = [\boldsymbol{u}_1 \dots \boldsymbol{u}_N]^\top$ 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)
- We can write each element of matrix $\mathbf{R}$ as

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

- Given $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$, any unobserved element in $\mathbf{R}$ can be predicted using the above

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

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

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

- The denominator (and hence the posterior) is intractable!
- Therefore, the posterior must be approximated somehow

The Posterior

- Our target posterior distribution for this model will be

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

- The denominator (and hence the posterior) is intractable!
- Therefore, the posterior must be approximated somehow
- One way to approximate is to compute Conditional Posterior (CP) over individual variables, 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!
- Therefore, the posterior must be approximated somehow
- One way to approximate is to compute Conditional Posterior (CP) over individual variables, e.g.,

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

- $\mathbf{U}_{-i}$ denotes all of $\mathbf{U}$ except $\boldsymbol{u}_i$. Note: $\mathbf{V},\mathbf{U}_{-i}$ is the set of all unknowns except $\boldsymbol{u}_i$

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!
- Therefore, the posterior must be approximated somehow
- One way to approximate is to compute Conditional Posterior (CP) over individual variables, e.g.,

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

- $\mathbf{U}_{-i}$ denotes all of $\mathbf{U}$ except $\boldsymbol{u}_i$. Note: $\mathbf{V},\mathbf{U}_{-i}$ is the set of all unknowns except $\boldsymbol{u}_i$
- $\mathbf{V}_{-j}$ denotes all of $\mathbf{V}$ except $\boldsymbol{v}_j$. Note: $\mathbf{U},\mathbf{V}_{-j}$ is the set of all unknowns except $\boldsymbol{v}_j$

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!
- Therefore, the posterior must be approximated somehow
- One way to approximate is to compute Conditional Posterior (CP) over individual variables, e.g.,

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

- $\mathbf{U}_{-i}$ denotes all of $\mathbf{U}$ except $\boldsymbol{u}_i$. Note: $\mathbf{V},\mathbf{U}_{-i}$ is the set of all unknowns except $\boldsymbol{u}_i$
- $\mathbf{V}_{-j}$ denotes all of $\mathbf{V}$ except $\boldsymbol{v}_j$. Note: $\mathbf{U},\mathbf{V}_{-j}$ is the set of all unknowns except $\boldsymbol{v}_j$
- Caveat: Each CP should be "computable" (but this is possible for models with "local conjugacy")

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!
- Therefore, the posterior must be approximated somehow
- One way to approximate is to compute Conditional Posterior (CP) over individual variables, e.g.,

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

- $\mathbf{U}_{-i}$ denotes all of $\mathbf{U}$ except $\boldsymbol{u}_i$. Note: $\mathbf{V},\mathbf{U}_{-i}$ is the set of all unknowns except $\boldsymbol{u}_i$
- $\mathbf{V}_{-j}$ denotes all of $\mathbf{V}$ except $\boldsymbol{v}_j$. Note: $\mathbf{U},\mathbf{V}_{-j}$ is the set of all unknowns except $\boldsymbol{v}_j$
- Caveat: Each CP should be "computable" (but this is possible for models with "local conjugacy")
- Since CP of each var. depends on all other vars, inference algos based on computing CPs usually work in alternating fashion, until each CP converges (e.g., Gibbs sampling which we'll look at later)

Conditional Posterior and Local Conjugacy

Conditional Posterior and Local Conjugacy

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

Conditional Posterior and Local Conjugacy

- Conditional Posteriors are easy to compute if the model admits local conjugacy
  - Note: Some researchers also call CP as Complete Conditional or Local Posterior

Conditional Posterior and Local Conjugacy

- Conditional Posteriors are easy to compute if the model admits local conjugacy
  - Note: Some researchers also call CP as Complete Conditional or Local Posterior
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$

Conditional Posterior and Local Conjugacy

- Conditional Posteriors are easy to compute if the model admits local conjugacy
  - Note: Some researchers also call CP as Complete Conditional or Local Posterior
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (will be the case if $p(\Theta)$ isn't conjugate)

Conditional Posterior and Local Conjugacy

- Conditional Posteriors are easy to compute if the model admits local conjugacy
  - Note: Some researchers also call CP as Complete Conditional or Local Posterior
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (will be the case if $p(\Theta)$ isn't conjugate)
- However suppose we can compute the following conditional posterior tractably

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

Conditional Posterior and Local Conjugacy

- Conditional Posteriors are easy to compute if the model admits local conjugacy
    - Note: Some researchers also call CP as Complete Conditional or Local Posterior
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (will be the case if $p(\Theta)$ isn't conjugate)
- However suppose we can compute the following conditional posterior 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

Conditional Posterior and Local Conjugacy

- Conditional Posteriors are easy to compute if the model admits local conjugacy
  - Note: Some researchers also call CP as Complete Conditional or Local Posterior
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (will be the case if $p(\Theta)$ isn't conjugate)
- However suppose we can compute the following conditional posterior 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

Conditional Posterior and Local Conjugacy

- Conditional Posteriors are easy to compute if the model admits local conjugacy
  - Note: Some researchers also call CP as Complete Conditional or Local Posterior
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (will be the case if $p(\Theta)$ isn't conjugate)
- However suppose we can compute the following conditional posterior 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 $p(\mathbf{X}|\theta_k, \Theta_{-k})$

Conditional Posterior and Local Conjugacy

- Conditional Posteriors are easy to compute if the model admits local conjugacy
  - Note: Some researchers also call CP as Complete Conditional or Local Posterior
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (will be the case if $p(\Theta)$ isn't conjugate)
- However suppose we can compute the following conditional posterior 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

Conditional Posterior and Local Conjugacy

- Conditional Posteriors are easy to compute if the model admits local conjugacy
  - Note: Some researchers also call CP as Complete Conditional or Local Posterior
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (will be the case if $p(\Theta)$ isn't conjugate)
- However suppose we can compute the following conditional posterior 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

Conditional Posterior and Local Conjugacy

- Conditional Posteriors are easy to compute if the model admits local conjugacy
  - Note: Some researchers also call CP as Complete Conditional or Local Posterior
- Consider a general model with data $\mathbf{X}$ and K unknown params/hyperparams $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$
- Suppose posterior $p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})}$ is intractable (will be the case if $p(\Theta)$ isn't conjugate)
- However suppose we can compute the following conditional posterior 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

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
- Need a way to "combine" these CPs to get the overall posterior

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
- Need a way to "combine" these CPs to get the overall posterior
- One way to get the overall representation of the posterior can be can be using sampling based inference algorithms like Gibbs sampling or MCMC (more on this later)

Detour: Gibbs Sampling (Geman and Geman, 1982)

- A general sampling algorithm to simulate samples from multivariate distributions

Detour: Gibbs Sampling (Geman and Geman, 1982)

- A general sampling algorithm to simulate samples from multivariate distributions
- Samples one component at a time from its conditional, conditioned on all other components

Detour: Gibbs Sampling (Geman and Geman, 1982)

- A general sampling algorithm to simulate samples from multivariate distributions
- Samples one component at a time from its conditional, conditioned on all other components
  - Assumes that the conditional distributions are available in a closed form

Detour: Gibbs Sampling (Geman and Geman, 1982)

- A general sampling algorithm to simulate samples from multivariate distributions
- Samples one component at a time from its conditional, conditioned on all other components
  - Assumes that the conditional distributions are available in a closed form

Suppose

$$\theta \sim N_2(0, \Sigma) \qquad \Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}.$$

Then

$$\begin{aligned} \theta_1|\theta_2 &\sim N\left(\rho\theta_2, [1-\rho^2]\right) \\ \theta_2|\theta_1 &\sim N\left(\rho\theta_1, [1-\rho^2]\right) \end{aligned}$$

are the conditional distributions.

Detour: Gibbs Sampling (Geman and Geman, 1982)

- A general sampling algorithm to simulate samples from multivariate distributions
- Samples one component at a time from its conditional, conditioned on all other components
  - Assumes that the conditional distributions are available in a closed form

Suppose

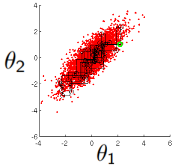
$$\theta \sim N_2(0, \Sigma) \qquad \Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}.$$

Then

$$\begin{aligned} \theta_1|\theta_2 &\sim N\left(\rho\theta_2, [1-\rho^2]\right) \\ \theta_2|\theta_1 &\sim N\left(\rho\theta_1, [1-\rho^2]\right) \end{aligned}$$

are the conditional distributions.

- The generated samples give a sample-based approximation of the multivariate distribution



Detour: Gibbs Sampling (Geman and Geman, 1982)

- Can be used to get a sampling-based approximation of a multiparameter posterior distribution

Detour: Gibbs Sampling (Geman and Geman, 1982)

- Can be used to get a sampling-based approximation of a multiparameter posterior distribution
- Gibbs sampler iteratively draws random samples from conditional posteriors

Detour: Gibbs Sampling (Geman and Geman, 1982)

- Can be used to get a sampling-based approximation of a multiparameter posterior distribution
- Gibbs sampler iteratively draws random samples from conditional posteriors
- When run long enough, the sampler produces samples from the joint posterior

Detour: Gibbs Sampling (Geman and Geman, 1982)

- Can be used to get a sampling-based approximation of a multiparameter posterior distribution
- Gibbs sampler iteratively draws random samples from conditional posteriors
- When run long enough, the sampler produces samples from the joint posterior
- For the simple two-parameter case $\theta = (\theta_1, \theta_2)$, the Gibb sampler looks like this

Detour: Gibbs Sampling (Geman and Geman, 1982)

- Can be used to get a sampling-based approximation of a multiparameter posterior distribution
- Gibbs sampler iteratively draws random samples from conditional posteriors
- When run long enough, the sampler produces samples from the joint posterior
- For the simple two-parameter case $\theta = (\theta_1, \theta_2)$, the Gibb sampler looks like this
  - Initialize $\theta_2^{(0)}$

Detour: Gibbs Sampling (Geman and Geman, 1982)

- Can be used to get a sampling-based approximation of a multiparameter posterior distribution
- Gibbs sampler iteratively draws random samples from conditional posteriors
- When run long enough, the sampler produces samples from the joint posterior
- For the simple two-parameter case $\theta = (\theta_1, \theta_2)$, the Gibb sampler looks like this
  - Initialize $\theta_2^{(0)}$
  - For $s = 1, \ldots, S$

Detour: Gibbs Sampling (Geman and Geman, 1982)

- Can be used to get a sampling-based approximation of a multiparameter posterior distribution
- Gibbs sampler iteratively draws random samples from conditional posteriors
- When run long enough, the sampler produces samples from the joint posterior
- For the simple two-parameter case $\theta = (\theta_1, \theta_2)$, the Gibb sampler looks like this
  - Initialize $\theta_2^{(0)}$
  - For $s = 1, \ldots, S$
    - Draw a random sample for θ_1 as $\theta_1^{(s)} \sim p(\theta_1 | \theta_2^{(s-1)}, \boldsymbol{y})$

Detour: Gibbs Sampling (Geman and Geman, 1982)

- Can be used to get a sampling-based approximation of a multiparameter posterior distribution
- Gibbs sampler iteratively draws random samples from conditional posteriors
- When run long enough, the sampler produces samples from the joint posterior
- For the simple two-parameter case $\theta = (\theta_1, \theta_2)$, the Gibb sampler looks like this
  - Initialize $\theta_2^{(0)}$
  - For $s = 1, \ldots, S$
    - Draw a random sample for θ_1 as $\theta_1^{(s)} \sim p(\theta_1 | \theta_2^{(s-1)}, \boldsymbol{y})$
    - Draw a random sample for θ_2 as $\theta_2^{(s)} \sim p(\theta_2 | \theta_1^{(s)}, \boldsymbol{y})$

Detour: Gibbs Sampling (Geman and Geman, 1982)

- Can be used to get a sampling-based approximation of a multiparameter posterior distribution
- Gibbs sampler iteratively draws random samples from conditional posteriors
- When run long enough, the sampler produces samples from the joint posterior
- For the simple two-parameter case $\theta = (\theta_1, \theta_2)$, the Gibb sampler looks like this
  - Initialize $\theta_2^{(0)}$
  - For $s = 1, \ldots, S$
    - Draw a random sample for θ_1 as $\theta_1^{(s)} \sim p(\theta_1|\theta_2^{(s-1)}, \boldsymbol{y})$
    - Draw a random sample for θ_2 as $\theta_2^{(s)} \sim p(\theta_2|\theta_1^{(s)}, \boldsymbol{y})$
- The set of S random samples $\{\theta_1^{(s)}, \theta_2^{(s)}\}_{s=1}^{S}$ represent the joint posterior distribution $p(\theta_1, \theta_2|\boldsymbol{y})$

Detour: Gibbs Sampling (Geman and Geman, 1982)

- Can be used to get a sampling-based approximation of a multiparameter posterior distribution
- Gibbs sampler iteratively draws random samples from conditional posteriors
- When run long enough, the sampler produces samples from the joint posterior
- For the simple two-parameter case $\theta = (\theta_1, \theta_2)$, the Gibb sampler looks like this
  - Initialize $\theta_2^{(0)}$
  - For $s = 1, \ldots, S$
    - Draw a random sample for θ_1 as $\theta_1^{(s)} \sim p(\theta_1 | \theta_2^{(s-1)}, \boldsymbol{y})$
    - Draw a random sample for θ_2 as $\theta_2^{(s)} \sim p(\theta_2 | \theta_1^{(s)}, \boldsymbol{y})$
- The set of S random samples $\{\theta_1^{(s)}, \theta_2^{(s)}\}_{s=1}^{S}$ represent the joint posterior distribution $p(\theta_1, \theta_2 | \boldsymbol{y})$
- More on Gibbs sampling when we discuss MCMC sampling algorithms (above is the high-level idea)

Back to Bayesian Matrix Factorization..

Bayesian Matrix Factorization: Conditional Posteriors

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

Bayesian Matrix Factorization: Conditional Posteriors

- 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: Conditional Posteriors

- 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: Conditional Posteriors

- 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: Conditional Posteriors

- 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: Conditional Posteriors

- 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: Conditional Posteriors

- 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: Conditional Posteriors

- 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: Conditional Posteriors

- 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: Conditional Posteriors

- 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: Conditional Posteriors

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

Bayesian Matrix Factorization: Conditional Posteriors

- 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: Conditional Posteriors

- 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: Conditional Posteriors

- 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: Conditional Posteriors

- 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: Conditional Posteriors

- 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

Bayesian Matrix Factorization: Conditional Posteriors

- 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
  - 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: Conditional Posteriors

- 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
  - 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

Bayesian Matrix Factorization

- The posterior predictive distribution for BMF (assuming other hyperparams known)

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

Bayesian Matrix Factorization

- The posterior predictive distribution for BMF (assuming other hyperparams known)

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

- In general, this is hard and needs approximation
  - If we are using Gibbs sampling, we can use the S samples $\{\boldsymbol{u}_i^{(s)}, \boldsymbol{v}_j^{(s)}\}_{s=1}^S$ to compute the mean
  - For the Gaussian likelihood case, the mean can be computed as

$$\mathbb{E}[r_{ij}] \approx \frac{1}{S}\sum_{s=1}^{S} \boldsymbol{u}_i^{(s)\top} \boldsymbol{v}_j^{(s)} \qquad \text{(Monte-Carlo averaging)}$$

  - Can also compute the variance of r_{ij} (think how)

Bayesian Matrix Factorization

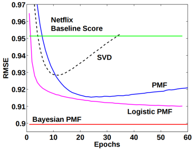
- The posterior predictive distribution for BMF (assuming other hyperparams known)

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

- In general, this is hard and needs approximation
  - If we are using Gibbs sampling, we can use the S samples $\{\boldsymbol{u}_i^{(s)}, \boldsymbol{v}_j^{(s)}\}_{s=1}^S$ to compute the mean
  - For the Gaussian likelihood case, the mean can be computed as

$$\mathbb{E}[r_{ij}] \approx \frac{1}{S}\sum_{s=1}^{S} \boldsymbol{u}_i^{(s)\top} \boldsymbol{v}_j^{(s)} \qquad \text{(Monte-Carlo averaging)}$$

  - Can also compute the variance of r_{ij} (think how)
- A comparison of Bayesian MF with other methods (from Salakhutdinov and Mnih (2008))



Summary and Some Comments

- Bayesian inference in even very complex probabilistic models can often be performed rather easily if the models have the local conjugacy property

Summary and Some Comments

- Bayesian inference in even very 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

- Bayesian inference in even very 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

- Bayesian inference in even very 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 Gibbs sampling, MCMC, EM, variational inference, etc.