Stochastic Variational Inference

CS698S Project Presentation

Keshaw Singh, 13347
Prakhar Kulshreshtha, 13485
Sandipan Mandal, 13807616

April 26, 2017

Outline

Introduction

- **Variational Inference** (VI) is a popular technique for estimating posterior given the data and the prior.

Introduction

- **Variational Inference** (VI) is a popular technique for estimating posterior given the data and the prior.
- It involves minimizing the KL-divergence b/w a family of distributions $q(\theta|\lambda)$ and the posterior $p(\theta|X)$, to approximate p via q.

Introduction

- **Variational Inference** (VI) is a popular technique for estimating posterior given the data and the prior.
- It involves minimizing the KL-divergence b/w a family of distributions $q(\theta|\lambda)$ and the posterior $p(\theta|X)$, to approximate p via q.
- However,VI in its batch setting is often not scalable to large datasets.

Introduction

- **Variational Inference** (VI) is a popular technique for estimating posterior given the data and the prior.
- It involves minimizing the KL-divergence b/w a family of distributions $q(\theta|\lambda)$ and the posterior $p(\theta|X)$, to approximate p via q.
- However,VI in its batch setting is often not scalable to large datasets.
- Fortunately Variational Inference can done in stochastic setting, popularly known as the **Stochastic Variational Inference**.

About the Project

- Our project had two phases:
  1. **Reading** : We read and understand some literature on SVI, to develop an understanding of its working and its scope.
  2. **PMF** : Next phase was to utilise SVI for PMF and then extend it further, to include activity and performance parameters (discussed later).

Stochastic Variational Inference (SVI)

- We first read some SVI related papers:
  1. Stochastic Variational Inference [Hoffman et al.,]
  2. An Adaptive Learning Rate for SVI [Ranganath et al., b]
  3. Black Box Variational Inference [Ranganath et al., a]
  4. Streaming Variational Bayes [Broderick and Jordan., 2013]
- More details in midsem report

Introduction

- In Matrix Factorisation, we have a sparsely filled ratings matrix **R** of size NxM, N: no. of users and M: no. of items.

Introduction

- In Matrix Factorisation, we have a sparsely filled ratings matrix **R** of size NxM, N: no. of users and M: no. of items.
- We assume $\mathbf{R} = \mathbf{UV}$, where U: NxK; V: MxK. K latent factors for each user and item.

Introduction

- In Matrix Factorisation, we have a sparsely filled ratings matrix **R** of size NxM, N: no. of users and M: no. of items.
- We assume $\mathbf{R} = \mathbf{UV}$, where U: NxK; V: MxK. K latent factors for each user and item.
- U and V both can be assumed to be generated from Poisson Distribution.

Introduction

- In Matrix Factorisation, we have a sparsely filled ratings matrix **R** of size NxM, N: no. of users and M: no. of items.
- We assume $\mathbf{R} = \mathbf{UV}$, where U: NxK; V: MxK. K latent factors for each user and item.
- U and V both can be assumed to be generated from Poisson Distribution.
- Should be a better choice than Gaussian MF, since it takes only +ve values.

Generative Model

The generative model for Poisson Matrix factorization can be depicted as follows:

1. 1. For each component k of each user u, sample preference:
   $\theta_{uk} \sim \text{Gamma}(a, b)$

Generative Model

The generative model for Poisson Matrix factorization can be depicted as follows:

1. 1. For each component k of each user u, sample preference: $\theta_{uk} \sim \text{Gamma}(a, b)$
2. 2. For each component k of each item i, sample attribute: $\beta_{ik} \sim \text{Gamma}(c, d)$.

Generative Model

The generative model for Poisson Matrix factorization can be depicted as follows:

1. 1. For each component k of each user u, sample preference: $\theta_{uk} \sim \text{Gamma}(a, b)$
2. 2. For each component k of each item i, sample attribute: $\beta_{ik} \sim \text{Gamma}(c, d)$.
3. 3. For each user u and item i, sample rating: $y_{ui} \sim \text{Poisson}({\theta_u}^T \beta_i)$.

Update Steps

The update steps for the parameters in this model are:

Update Steps

The update steps for the parameters in this model are:

1. Users
   (a) $a_{new} = a + \sum_{m=1}^{M} R_{nm}\phi_k^{(nm)}$
   (b) $b_{new} = b + \sum_{m=1}^{M} \mathbb{E}_q[\beta_{mk}]$

Update Steps

The update steps for the parameters in this model are:

1. Users
   (a) $a_{new} = a + \sum_{m=1}^{M} R_{nm}\phi_k^{(nm)}$
   (b) $b_{new} = b + \sum_{m=1}^{M} \mathbb{E}_q[\beta_{mk}]$
2. Items
   (a) $c_{new} = c + \sum_{n=1}^{N} R_{nm}\phi_k^{(nm)}$
   (b) $d_{new} = d + \sum_{n=1}^{N} \mathbb{E}_q[\theta_{nk}]$

where: $\phi_k^{nm} = e^{\mathbb{E}_q[z_k]} / \sum_{l=1}^{K} e^{\mathbb{E}_q[z_l]}$
and, $z_k = \theta_{nk}\beta_{mk}$

Outline

Performance and Activity

Extension

The generative model for Poisson Matrix factorization can be depicted as follows:

Performance and Activity
Extension

The generative model for Poisson Matrix factorization can be depicted as follows:

1. For each user u:
   (a) Sample activity: $\xi_u \sim \text{Gamma}(a', a'/b')$
   (b) For each component k, sample preference: $\theta_{uk} \sim \text{Gamma}(a, \xi_u)$.

Performance and Activity

Extension

The generative model for Poisson Matrix factorization can be depicted as follows:

1. For each user u:
   (a) Sample activity: $\xi_u \sim \text{Gamma}(a^{'}, a^{'}/b^{'})$
   (b) For each component k, sample preference: $\theta_{uk} \sim \text{Gamma}(a, \xi_u)$.
2. For each user i:
   (a) Sample popularity: $\eta_i \sim \text{Gamma}(c^{'}, c^{'}/d^{'})$
   (b) For each component k, sample attribute: $\beta_{ik} \sim \text{Gamma}(c, \eta_i)$.

Performance and Activity

Extension

The generative model for Poisson Matrix factorization can be depicted as follows:

1. For each user u:
   (a) Sample activity: $\xi_u \sim \text{Gamma}(a^{'}, a^{'}/b^{'})$
   (b) For each component k, sample preference: $\theta_{uk} \sim \text{Gamma}(a, \xi_u)$.
2. For each user i:
   (a) Sample popularity: $\eta_i \sim \text{Gamma}(c^{'}, c^{'}/d^{'})$
   (b) For each component k, sample attribute: $\beta_{ik} \sim \text{Gamma}(c, \eta_i)$.
3. For each user u and item i, sample rating: $y_{ui} \sim \text{Poisson}({\theta_u}^T \beta_i)$.

Outline

Inference on Heirarchical PMF

Notation

For Poisson Matrix Factorization with extra information (activity of users and popularity of items) we have assumed the following Variational Distributions for latent variables[Gopalan et al., 2015]:-

1. For Global Parameters -
   $q(\beta|\lambda) = \prod_{i,k} q(\beta_{ik}|\lambda_{ik}) = \prod_{i,k} Gamma(\beta_{ik}|\lambda_{ik}^{shp}, \lambda_{ik}^{rte})$

Inference on Heirarchical PMF

Notation

For Poisson Matrix Factorization with extra information (activity of users and popularity of items) we have assumed the following Variational Distributions for latent variables[Gopalan et al., 2015]:-

1. For Global Parameters -
   $q(\beta|\lambda) = \prod_{i,k} q(\beta_{ik}|\lambda_{ik}) = \prod_{i,k} Gamma(\beta_{ik}|\lambda_{ik}^{shp}, \lambda_{ik}^{rte})$
2. *Popularity* on Global Parameter -
   $q(\eta|\tau) = \prod_{i} q(\eta_i|\tau_i) = \prod_{i} Gamma(\eta_i|\tau_i^{shp}, \tau_i^{rte})$

Inference on Heirarchical PMF

Notation

For Poisson Matrix Factorization with extra information (activity of users and popularity of items) we have assumed the following Variational Distributions for latent variables[Gopalan et al., 2015]:-

1. For Global Parameters -
   $q(\beta|\lambda) = \prod_{i,k} q(\beta_{ik}|\lambda_{ik}) = \prod_{i,k} Gamma(\beta_{ik}|\lambda_{ik}^{shp}, \lambda_{ik}^{rte})$
2. *Popularity* on Global Parameter -
   $q(\eta|\tau) = \prod_{i} q(\eta_{i}|\tau_{i}) = \prod_{i} Gamma(\eta_{i}|\tau_{i}^{shp}, \tau_{i}^{rte})$
3. For Local Parameters -
   $q(\theta|\gamma) = \prod_{u,k} q(\theta_{uk}|\gamma_{uk}) = \prod_{u,k} Gamma(\theta_{uk}|\gamma_{uk}^{shp}, \gamma_{uk}^{rte})$

Inference on Heirarchical PMF

Notation

For Poisson Matrix Factorization with extra information (activity of users and popularity of items) we have assumed the following Variational Distributions for latent variables[Gopalan et al., 2015]:-

1. For Global Parameters -
   $q(\beta|\lambda) = \prod_{i,k} q(\beta_{ik}|\lambda_{ik}) = \prod_{i,k} Gamma(\beta_{ik}|\lambda_{ik}^{shp}, \lambda_{ik}^{rte})$
2. *Popularity* on Global Parameter -
   $q(\eta|\tau) = \prod_{i} q(\eta_{i}|\tau_{i}) = \prod_{i} Gamma(\eta_{i}|\tau_{i}^{shp}, \tau_{i}^{rte})$
3. For Local Parameters -
   $q(\theta|\gamma) = \prod_{u,k} q(\theta_{uk}|\gamma_{uk}) = \prod_{u,k} Gamma(\theta_{uk}|\gamma_{uk}^{shp}, \gamma_{uk}^{rte})$
4. *Activity* on Local Parameter -
   $q(\xi|\kappa) = \prod_{u} q(\eta_{u}|\kappa_{u}) = \prod_{u} Gamma(\eta_{u}|\kappa_{u}^{shp}, \kappa_{u}^{rte})$

Inference on Heirarchical PMF

Notation

For Poisson Matrix Factorization with extra information (activity of users and popularity of items) we have assumed the following Variational Distributions for latent variables[Gopalan et al., 2015]:-

1. For Global Parameters - $q(\beta|\lambda) = \prod_{i,k} q(\beta_{ik}|\lambda_{ik}) = \prod_{i,k} Gamma(\beta_{ik}|\lambda_{ik}^{shp}, \lambda_{ik}^{rte})$
2. *Popularity* on Global Parameter - $q(\eta|\tau) = \prod_i q(\eta_i|\tau_i) = \prod_i Gamma(\eta_i|\tau_i^{shp}, \tau_i^{rte})$
3. For Local Parameters - $q(\theta|\gamma) = \prod_{u,k} q(\theta_{uk}|\gamma_{uk}) = \prod_{u,k} Gamma(\theta_{uk}|\gamma_{uk}^{shp}, \gamma_{uk}^{rte})$
4. *Activity* on Local Parameter - $q(\xi|\kappa) = \prod_u q(\eta_u|\kappa_u) = \prod_u Gamma(\eta_u|\kappa_u^{shp}, \kappa_u^{rte})$
5. Auxilliary Variable $z_{uik} \sim Poisson(\theta_{uk}\beta ik)$ $q(z|\phi) = \prod_{u,i} q(z_{ui}|\phi_{ui}) = \prod_{u,i} Multinomial(z_{ui}|\phi_{ui})$

Inference on Heirarchical PMF

Batch VB update rules[Gopalan et al., 2015]

1. $\kappa_u^{shp} = a' + Ka$ (Available in closed form)

Inference on Heirarchical PMF

Batch VB update rules[Gopalan et al., 2015]

1. $\kappa_u^{shp} = a' + Ka$ (Available in closed form)
2. $\tau_i^{shp} = c' + Kc$ (Available in closed form)

Inference on Heirarchical PMF

Batch VB update rules[Gopalan et al., 2015]

1. $\kappa_u^{shp} = a' + Ka$ (Available in closed form)
2. $\tau_i^{shp} = c' + Kc$ (Available in closed form)
3. Auxilliary Parameter -
   $\phi_{ui} \propto exp\{\psi(\gamma_{uk}^{shp}) - log(\gamma_{uk}^{rte}) + \psi(\lambda_{ik}^{shp}) - log(\lambda_{ik}^{rte})\}$

Inference on Heirarchical PMF

Batch VB update rules[Gopalan et al., 2015]

1. $\kappa_u^{shp} = a' + Ka$ (Available in closed form)
2. $\tau_i^{shp} = c' + Kc$ (Available in closed form)
3. Auxilliary Parameter -
   $\phi_{ui} \propto exp\{\psi(\gamma_{uk}^{shp}) - log(\gamma_{uk}^{rte}) + \psi(\lambda_{ik}^{shp}) - log(\lambda_{ik}^{rte})\}$
4. Local Parameters -
   - $\gamma_{uk}^{shp} = a + \sum_i y_{ui}\phi_{uik}$
   - $\gamma_{uk}^{rte} = \frac{\kappa_u^{shp}}{\kappa_u^{rte}} + \sum_i \frac{\lambda_{ik}^{shp}}{\lambda_{ik}^{rte}}$
   - $\kappa_u^{rte} = \frac{a'}{b'} + \sum_k \frac{\gamma_{uk}^{shp}}{\gamma_{uk}^{rte}}$

Inference on Heirarchical PMF

Batch VB update rules[Gopalan et al., 2015]

1. $\kappa_u^{shp} = a' + Ka$ (Available in closed form)
2. $\tau_i^{shp} = c' + Kc$ (Available in closed form)
3. Auxilliary Parameter -
   $\phi_{ui} \propto exp\{\psi(\gamma_{uk}^{shp}) - log(\gamma_{uk}^{rte}) + \psi(\lambda_{ik}^{shp}) - log(\lambda_{ik}^{rte})\}$
4. Local Parameters -
   - $\gamma_{uk}^{shp} = a + \sum_i y_{ui}\phi_{uik}$
   - $\gamma_{uk}^{rte} = \frac{\kappa_u^{shp}}{\kappa_u^{rte}} + \sum_i \frac{\lambda_{ik}^{shp}}{\lambda_{ik}^{rte}}$
   - $\kappa_u^{rte} = \frac{a'}{b'} + \sum_k \frac{\gamma_{uk}^{shp}}{\gamma_{uk}^{rte}}$
5. Global Parameters -
   - $\lambda_{ik}^{shp} = c + \sum_u y_{ui}\phi_{uik}$
   - $\lambda_{ik}^{rte} = \frac{\tau_i^{shp}}{\tau_i^{rte}} + \sum_u \frac{\gamma_{uk}^{shp}}{\gamma_{uk}^{rte}}$
   - $\tau_i^{rte} = \frac{c'}{d'} + \sum_k \frac{\lambda_{ik}^{shp}}{\lambda_{ik}^{rte}}$

Inference on Heirarchical PMF

SVI update rules

1. $\kappa_u^{shp} = a' + Ka$ (Available in closed form)

Inference on Heirarchical PMF

SVI update rules

1. $\kappa_u^{shp} = a' + Ka$ (Available in closed form)
2. $\tau_i^{shp} = c' + Kc$ (Available in closed form)
3. Auxilliary Parameter -
   $\phi_{ui} \propto exp\{\psi(\gamma_{uk}^{shp}) - log(\gamma_{uk}^{rte}) + \psi(\lambda_{ik}^{shp}) - log(\lambda_{ik}^{rte})\}$

Inference on Heirarchical PMF

SVI update rules

1. $\kappa_u^{shp} = a' + Ka$ (Available in closed form)
2. $\tau_i^{shp} = c' + Kc$ (Available in closed form)
3. Auxilliary Parameter -
   $\phi_{ui} \propto exp\{\psi(\gamma_{uk}^{shp}) - log(\gamma_{uk}^{rte}) + \psi(\lambda_{ik}^{shp}) - log(\lambda_{ik}^{rte})\}$
4. Local Parameters (For a mini batch of size B, update only the relevant local parameters) -
   - $\gamma_{uk}^{shp} = a + \sum_i y_{ui}\phi_{uik}$
   - $\gamma_{uk}^{rte} = \frac{\kappa_u^{shp}}{\kappa_u^{rte}} + \sum_i \frac{\lambda_{ik}^{shp}}{\lambda_{ik}^{rte}}$
   - $\kappa_u^{rte} = \frac{a'}{b'} + \sum_k \frac{\gamma_{uk}^{shp}}{\gamma_{uk}^{rte}}$

Inference on Heirarchical PMF

SVI update rules

1. $\kappa_u^{shp} = a' + Ka$ (Available in closed form)
2. $\tau_i^{shp} = c' + Kc$ (Available in closed form)
3. Auxilliary Parameter -
   $\phi_{ui} \propto exp\{\psi(\gamma_{uk}^{shp}) - log(\gamma_{uk}^{rte}) + \psi(\lambda_{ik}^{shp}) - log(\lambda_{ik}^{rte})\}$
4. Local Parameters (For a mini batch of size B, update only the relevant local parameters) -
   - $\gamma_{uk}^{shp} = a + \sum_i y_{ui}\phi_{uik}$
   - $\gamma_{uk}^{rte} = \frac{\kappa_u^{shp}}{\kappa_u^{rte}} + \sum_i \frac{\lambda_{ik}^{shp}}{\lambda_{ik}^{rte}}$
   - $\kappa_u^{rte} = \frac{a'}{b'} + \sum_k \frac{\gamma_{uk}^{shp}}{\gamma_{uk}^{rte}}$
5. Global Parameters $(scale = (Numberofsamples)/|B|$
   - $\lambda_{ik}^{shp} = (1-\rho)\lambda_{ik}^{shp} + \rho(c + scale \times \sum_u y_{ui}\phi_{uik})$
   - $\lambda_{ik}^{rte} = (1-\rho)\lambda_{ik}^{rte} + \rho(\frac{\tau_i^{shp}}{\tau_i^{rte}} + scale \times \sum_u \frac{\gamma_{uk}^{shp}}{\gamma_{uk}^{rte}})$
   - $\tau_i^{rte} = (1-\rho)\tau_i^{rte} + \rho(\frac{c'}{d'} + scale \times \sum_k \frac{\lambda_{ik}^{shp}}{\lambda_{ik}^{rte}})$

Outline

Extended v/s Normal

Experiments

- We utilised [Liang,]'s implementation of PMF and Stochastic-PMF for our experiments.

Extended v/s Normal

Experiments

- We utilised [Liang,]'s implementation of PMF and Stochastic-PMF for our experiments.
- The code returns the hyperparameter matrix for both user and item.

Extended v/s Normal

Experiments

- We utilised [Liang,]'s implementation of PMF and Stochastic-PMF for our experiments.
- The code returns the hyperparameter matrix for both user and item.
- We have tried on a subset of Netflix Data:

Extended v/s Normal

Experiments

- We utilised [Liang,]'s implementation of PMF and Stochastic-PMF for our experiments.
- The code returns the hyperparameter matrix for both user and item.
- We have tried on a subset of Netflix Data:
- For prediction, we sample $U = Gamma(A_u, B_u)$ and $V = Gamma(A_v, B_v)$ followed by sampling from $Poisson(U^T V)$

Extended v/s Normal

Experiments

- We utilised [Liang,]'s implementation of PMF and Stochastic-PMF for our experiments.
- The code returns the hyperparameter matrix for both user and item.
- We have tried on a subset of Netflix Data:
- For prediction, we sample $U = Gamma(A_u, B_u)$ and $V = Gamma(A_v, B_v)$ followed by sampling from $Poisson(U^T V)$
- This is repeated 100 times and then average is taken.

Extended v/s Normal

Experiments

- We utilised [Liang,]'s implementation of PMF and Stochastic-PMF for our experiments.
- The code returns the hyperparameter matrix for both user and item.
- We have tried on a subset of Netflix Data:
- For prediction, we sample $U = Gamma(A_u, B_u)$ and $V = Gamma(A_v, B_v)$ followed by sampling from $Poisson(U^T V)$
- This is repeated 100 times and then average is taken.
- We also extended the Original Stochastic PMF to incorporate the Activity and Performance [Gopalan et al., 2015].

Netflix Data
Experiments

- We have tested algorithms on a small subset of netflix data.

Netflix Data

Experiments

- We have tested algorithms on a small subset of netflix data.
- No. of customers = 5000; No. of movies = 200. K=50. No. of Iterations: 10 for PMF(SVI/VI), 1000 for Gaussian Matrix Factorisation (GMF).

	Batch PMF	SVI PMF	PA SVI PMF	Batch GMF
RMSE	1.887	2.071	2.314	0.576
Accuracy	0.475	0.485	0.522	0.648

Table: Results on a subset of Netflix Data

MovieLens Movie Rating Dataset

Experiments

- We have tested algorithms on a small subset of Movielens data.

MovieLens Movie Rating Dataset

Experiments

- We have tested algorithms on a small subset of Movielens data.
- No. of customers = 5000; No. of movies = 200. K=50. No. of Iterations: 10 for PMF(SVI/VI), 1000 for Gaussian Matrix Factorisation (GMF).

	Batch PMF	SVI PMF	PA SVI PMF	Batch GMF
RMSE	1.412	1.814	1.993	0.564
Accuracy	0.509	0.548	0.661	0.652

Table: Results on a subset of Movielens Data

Challenges

Conclusion

- Althought the Heirarchical PMF-SVI model is giving higher accuracy than normal PMF-SVI.

Challenges
Conclusion

- Althought the Heirarchical PMF-SVI model is giving higher accuracy than normal PMF-SVI.
- All variants of PMF give inferior performance to basic GMF.

Challenges

Conclusion

- Althought the Heirarchical PMF-SVI model is giving higher accuracy than normal PMF-SVI.
- All variants of PMF give inferior performance to basic GMF.
- Increasing no. of iterations more than 10 doesn't improve its performance. For GMF, increasing iters to 1000 gives RMSE = $\mathbf{0.141}$ and accuracy $\mathbf{0.999}$

Challenges

Conclusion

- Althought the Heirarchical PMF-SVI model is giving higher accuracy than normal PMF-SVI.
- All variants of PMF give inferior performance to basic GMF.
- Increasing no. of iterations more than 10 doesn't improve its performance. For GMF, increasing iters to 1000 gives RMSE $= \mathbf{0.141}$ and accuracy $\mathbf{0.999}$
- We suspect that our Matrix reconstruction by sampling might be wrong.

Conclusion

- We did a small survey on some literature of SVI.

Conclusion

- We did a small survey on some literature of SVI.
- We developed and implemented Stochastic version of Heirarchical PMF [Gopalan et al., 2015].

Conclusion

- We did a small survey on some literature of SVI.
- We developed and implemented Stochastic version of Heirarchical PMF [Gopalan et al., 2015].
- We'll try to improve the method and report some more experiments on larger data and different datasets.

References I

Broderick, Tamara, N. B. A. W. A. C. W. and Jordan., M. I. (2013).
Streaming variational bayes.
In Advances in Neural Information Processing Systems (pp. 1727-1735.).

Gopalan, P., Hofman, J. M., and Blei, D. M. (2015).
Scalable recommendation with hierarchical poisson factorization.
In *UAI*, pages 326–335.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J.
Stochastic variational inference.

Liang, D.
Stochastic pmf: https://github.com/dawenl/stochastic$_p mf$.

Ranganath, R., Gerrish, S., and Blei, D. M.
Black box variational inference.

References II

Ranganath, R., Wang, C., Blei, D. M., and Xing, E. P.
An adaptive learning rate for stochastic variational inference.