Stochastic Variational Inference
Final Report

Keshaw Singh
13347

Prakhar Kulshreshtha
13485

Sandipan Mandal
13807616

1 Problem Description

Our project was to survey (read and understand) some literature on Stochastic Variational Inference (SVI), and then after we have understood the concepts, to apply SVI on a model on which it has not yet been applied. So after reading a few papers and getting comfortable with the concepts we did some experiments with an SVI-model. We picked the Poisson Matrix Factorization via Variational Bayes (VB), and its SVI version, and analysed both of them.
We formulated and implemented SVI version of Hierarchical Poisson Matrix Factorization [Gopalan et al., 2015], since the paper only had batch VB updates. This will result in scaling up of their model to large datasets.

2 Literature Survey

2.1 Introduction

One of the central tasks in Bayesian Machine Learning is to compute the posterior distribution given data and prior. However in most of the real world scenario computing this posterior is intractable. One of the the popular techniques is **Variational Inference**, in which we minimize the KL-divergence b/w a family of distributions $q(\theta|\lambda)$ and the posterior $p(\theta|X)$, to approximate p via q. Variational Inference in its batch setting is often not scalable to large datasets. Fortunately Variational Inference can done in stochastic setting. In the next subsection we review **Stochastic Variational Inference**.

2.2 Stochastic Variational Inference

In this subsection we briefly review stochastic variational inference [Hoffman et al., 2013].

Our class of models involves observations, global hidden variables, local hidden variables, and fixed parameters. The N observations are $x = x_{1:N}$; the vector of global hidden variables is β; the N local hidden variables are $z = z_{1:N}$, each of which is a collection of J variables $z_n = z_{n,1:J}$; the vector of fixed parameters is α. The idea and general algorithm has been discussed in the class-lecture 15.

The paper assumes that the complete conditional on the hidden variables are from the exponential family.

$$p(\beta|x, z, \alpha) = h(\beta) exp[\eta_g(x, z, \alpha)^T t(\beta) - a_g(\eta_g(x, z, \alpha))]$$

$$p(z_{nj}|x_n, z_{n,-j}, \beta) = h(z_{nj}) exp[\eta_l(x_n, z_{n,-j}, \beta)^T t(z_{nj}) - a_g(\eta_l(x_n, z_{n,-j}, \beta))]$$

Assume $q(\beta|\lambda)$ and $q(z_{nj}|\phi_{nj})$ to be variational distribution. Using mean field assumption

$$q(z, \beta) = q(\beta|\lambda) \prod_{n=1}^{N} \prod_{j=1}^{J} q(z_{nj}|\phi_{nj})$$

We further assume that the variational distributions are in the same exponential family as the complete conditional.

$$q(\beta|\lambda) = h(\beta)exp[\lambda^T t(\beta) - a_g(\lambda)]$$
$$q(z_{nj}|\phi_{nj}) = h(z_{nj})exp[\phi_{nj}^T t(z_{nj}) - a_l(\phi_{nj})]$$

.

In batch setting computing the Evidence Lower Bound, differentiating w.r.t variational parameters and setting it to zero we get

$$\lambda = \mathbb{E}_q[\eta_g(x, z, \alpha)]$$
$$\phi_{nj} = \mathbb{E}_q[\eta_l(x_n, z_{n,-j}, \beta)]$$

In stochastic setting, we sample a data point x_i from the dataset at each iteration. Then we compute the local variational parameter ϕ_i (corresponding to x_i) as

$$\phi_i = \mathbb{E}_{\lambda^{(t-1)}}[\eta_g(x_i^{(N)}, z_i^{(N)})]$$

where $\lambda^{(t-1)}$ is the optimal value of λ from the previous iteration.
For global variational parameter we compute an intermedite paramater $\hat{\lambda}$ as

$$\hat{\lambda} = \mathbb{E}_{\phi_i}[\eta_g(x_i^{(N)}, z_i^{(N)})]$$

and update current best λ as

$$\lambda^{(t)} = (1 - \rho_t)\lambda^{(t-1)} + \rho_t\hat{\lambda}$$

where $(x_i^{(N)}, z_i^{(N)})$ is a data set consisting of (x_i, z_i) replicated N times and ρ_t is the learning rate. The algorithm is summarized in Algorithm 1. Instead of one random sample we can take a minibatch of random samples.

Algorithm 1 Stochastic variational inference for Exponential Family.

```
Initialize $\lambda^{(0)}$ randomly.
Set step size $\rho_t$ appropriately.
repeat
    Sample a data point $x_i$ from the dataset.
    Compute its local variational parameter.
        $\phi_i = \mathbb{E}_{\lambda^{(t-1)}}[\eta_g(x_i^{(N)}, z_i^{(N)})]$
    Compute intermediate global parameters as though $x_i$ is replicated N times
        $\hat{\lambda} = \mathbb{E}_{\phi_i}[\eta_g(x_i^{(N)}, z_i^{(N)})]$
    Update the current estimate of the global variational parameters
        $\lambda^{(t)} = (1 - \rho_t)\lambda^{(t-1)} + \rho_t\hat{\lambda}$
until forever
```

2.3 LDA Topic Model

Topic models are probabilistic models of document collections that use latent variables to encode recurring patterns of word use. In this subsection we review SVI for LDA topic model given in [Hoffman et al., 2013].

- Observations are words, organized into documents.
- The global hidden variables are the topics.
- There are two sets of local hidden variables - one for topic proportion of each document and the other topic assignment of each word.

- LDA assumes we know the number of topics - K in advance.

The topics are drawn from Dirichlet distribution and for each document topic proportion is drawn from another dirichlet distribution. The words and its topic assignment are drawn from Multinomial distribution.
Only difference it has with standard SVI is that it has two set of local variables. We sample a document randomly from the dataset. For each word we compute local variable corresponding to topic proportion and topic assignment alternatively until convergence. After that the global variable is updated as in standard SVI.
The paper also reports that Stochastic variational inference on the full data converges faster and to a better place than batch variational inference on a reasonably sized subset.

2.4 An Adaptive Learning Rate for SVI

Choosing appropriate learning rate often boosts the performance of learning algorithm w.r.t to rate of convergence and quality of end results. Stochastic Variational inference makes use of learning rate ρ_t in the computation of global variational parameter in each time step. To assure convergence of the algorithm the step sizes ρ_t need to satisfy the following conditions: $\sum_{t=1}^{\infty} \rho_t = \infty$ and $\sum_{t=1}^{\infty} \rho_t^2 < \infty$.
Choosing this sequence can be difficult and time-consuming. A sequence that decays too quickly may take a long time to converge; a sequence that decays too slowly may cause the parameters to oscillate too much.
[Ranganath et al., 2013] gives an alternative way to set a better learning rate. The learning rate is analytical and does not depend on developer's experience to set learning rate.
If batch update of global variational parameter is λ_t^* and stochastic update using one data point is λ_{t+1} then the learning rate minimizes the squared error between these two.

$$J(\rho_t) = (\lambda_{t+1} - \lambda_t^*)^T (\lambda_{t+1} - \lambda_t^*)$$

. If we define $\hat{\lambda}$ to be intermediate global parameter, optimal ρ_t is given by

$$\rho_t^* = \frac{(-\lambda_t + \lambda_t^*)^T (-\lambda + \lambda_t^*)}{(-\lambda_t + \lambda_t^*)^T (-\lambda_t + \lambda_t^*) + tr(\Sigma)}$$

where $\Sigma = cov(\hat{\lambda})$
But here λ_t^* is unknown. Fortunately we are able to estimate ρ_t^* using sampled natural gradient at time t (g_t). The paper shows that optimal ρ_t can also be computed as

$$\rho_t^* = \frac{\mathbb{E}[g_t]^T \mathbb{E}[g_t]}{\mathbb{E}[g_t^T g_t]}$$

. Further we can approximate ρ_t^* using the following approximations

$$\mathbb{E}[g_t] \approx (1 - \tau_t^{-1}) \mathbb{E}[g_{t-1}] + \tau_t^{-1} g_t$$

$$\mathbb{E}[g_t^T g_t] \approx (1 - \tau_t^{-1}) \mathbb{E}[g_{t-1}^T g_{t-1}] + \tau_t^{-1} g_t^T g_t$$

where τ_t is the window size of the exponential moving average at time t which has been shown to be updated as

$$\tau_{t+1} = \tau_t (1 - \rho_t^*) + 1$$

2.5 Black Box Variational Inference

[Hoffman et al., 2013] makes an additional assumption of conjugacy amongst the components of $q(\beta|\lambda)$ obtained via Mean Field Assumption. However, this reduces the possible kinds of models which can be tested. [Ranganath et al., ober] modifies the global parameter update step of SVI algorithm by approximating gradients (here they are not using natural gradients, just normal gradients over hyperparameters) using S monte-carlo samples from q. This helps in escaping calculation of local parameters, and, more importantly, from calculating the $\mathbb{E}_q(.)$, which is usually intractable in the absence of conjugacy.

Seeing the algorithm we can easily observe that $q(z_i|\beta, z_{-i})$ are used separately, and so we don't need conjugacy at all. The algorithm is fast and only assumes the capability of calculating $p(x, z[s])$,

Algorithm 2 Black Box Variational Inference

```
**Input:** data $x$, joint distribution $p$, mean field variational family $q$.
**Initialize** $\lambda_{1:n}$ randomly, $t = 1$.
**repeat**
  // Draw $S$ samples from $q$
  for $s = 1$ to S do
    $z[s] \sim q$
  end for
  $\rho$ = $t$th value of a Robbins Monro sequence (rm)
  $\lambda$ = $\lambda + \rho\frac{1}{S}\sum_{s=1}^{S}\nabla_\lambda \log q(z[s]|\lambda)(\log p(x, z[s]) - \log q(z[s]|\lambda))$
  $t = t + 1$
until change of $\lambda$ is less than 0.01.
```

$z[s] \sim q$ along with knowledge of $\log q(z[s]|\lambda)$. This allows us to test a vast no. of different models to choose the one that fits the data best. The estimate of gradient is unbiased, and to reduce its variance, authors modify the basic algorithm, by consecutively using **Rao-Blackwellization** and **Control-Variates** to give a modified BBVI [Ranganath et al., ober]. Finally, they emperically show their method superior to Gibbs Sampling on a dataset, and then show the 'black-boxedness' by doing inferences on a medical-dataset using various non-conjugate models like Gamma-Gamma, Gamma-Normal, etc.

2.6 Streaming Variational Bayes

SVI is based on the conceptual existence of a full data set containing D points, for a fixed value of D. The posterior being targeted is based on the D data points and posterior for D' points, $D' \neq D$, is not obtained as a part of this process. [Broderick and Jordan., 2013] develops a truly streaming procedure, in the sense that it yields an approximate posterior for each collection of D' obtained data points. Also, this method makes use of distributed and asynchronous computations.

2.6.1 Streaming Bayesian Updating

If Θ is the set of parameters and C_i is the i^{th} minibatch of data, then the posterior after observing b minibatches is calculated as follows:

$$p(\Theta|C_1, ..., C_b) \propto p(C_b|\Theta)p(\Theta|C_1, ..., C_{b-1})$$

The above equation holds if the two distributions on the LHS are conjugate to each other. $p(\Theta|C_1, ..., C_{b-1})$ acts as the prior over Θ after (b-1) minibatches have been observed.
When posteriors cannot be computed exactly, a case often encountered, approximation algorithms are used.

2.6.2 Distributed Bayesian Updating

Posterior calculations can be made faster by parallelizing computations. Making use of Bayes theorem, the algorithm can be made to calculate posteriors for individual minibatches and then combine them to get the full posterior.

2.6.3 Asynchronous Bayesian Updating

To maximize the potential gains from distributed computation, asynchronous updating is used.
In this case, processors called *workers* are each assigned a subproblem (calculating posterior for a minibatch). When a worker finishes, it reports its results to a single *master* processor. The master gives the worker a new problem without waiting for other workers to finish the work.

2.7 Bayesian Non negative Matrix Factorization[Schmidt et al., 2009]

Bayesian Matrix Factorization has wide range of applications in machine learning. The traditional Bayesian Matrix Factorization can be defined as follows:

- There is a sparse matrix X of dimension $N \times M$. This can be interpreted as a matrix where each row corresponds to a user while each column to an item. X_{ij} gives the rating given by user i to item j.
- X can be expressed as $X = UV^T + E$ where dimension of U is $N \times K$ and V is $M \times K$. K is the number of latent factors. U matrix contains the latent variables of the users and V contains the latent variables of the items. $E = \{\epsilon_{ij}\}$ is the noise.
- Each element of X can be represented as $X_{ij} = \boldsymbol{u_i}^T \boldsymbol{v_j} + \epsilon_{ij}$.
- $\epsilon_{ij} \sim N(0, \beta^{-1})$. So $p(X_{ij}|\boldsymbol{u_i}, \boldsymbol{v_j}) = N(X_{ij}|\boldsymbol{u_i}^T \boldsymbol{v_j}, \beta^{-1})$. $\boldsymbol{u_i}$ and $\boldsymbol{v_j}$ has Gaussian Prior.

But Bayesian Gaussian matrix factorization may not be a good model in the given scenario. (Ratings are non-negative and count valued, but our model can take negative and real value etc.) Hence a better model might be **Poisson Matrix Factorization** (PMF).
The generative model for this scheme is given below. (Matrix X is factorized into matrices $\Theta = \{\theta_{uk}\}$ and $B = \{\beta_{ik}\}$)

- For each user u and latent factor k, $\theta_{uk} \sim$ Gamma(a,b)
- For each item i and latent factor k, $\beta_{ik} \sim$ Gamma(c,d) $\forall \boldsymbol{v_j} \in V$
- For each user i and item j, rating X_{ij} is sampled as $X_{ij} \sim$ Poisson($\boldsymbol{\theta_u}^T \boldsymbol{\beta_i}$)

Poisson Matrix Factorization has many advantages over Gaussian Matrix Factorization which are -

- Ratings are generally positive and count valued and Poisson distribution gives samples from the same domain.
- The inference process only depends on the non zero entries in the X matrix. Since X is generally very sparse it scales up the process a lot.

Details of update rules are mentioned in the next section.

2.8 Scalable Recommendation with Hierarchical Poisson Factorization[Gopalan et al., 2015]

This subsection covers an extension of Poisson Matrix Factorization from [Gopalan et al., 2015]. Here associated with each user and items an extra set of latent variables are kept to infer user activity and item popularity. This makes the model closer to the real world. This model is callled Hierarchical Poisson Matrix Factorization.
The generative model for this Hierarchical Poisson matrix factorization is given below.

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

Details of update rules are mentioned in the next section.

3 Implementation of PMF

3.1 Overview

After doing literature survey, we did some experiments with Matrix Factorization. Initially we tried implementing variational inference for poisson matrix factorization on **Edward**. But after some time we realized that Edward does not support posterior estimation over gamma distribution. So we moved to implementation from scratch.
We used the code from the paper [Liang et al., 2014] for base code of Poisson Matrix Factorization. We did two modification to the code which are

- Returning local latent variable matrix for SVI (The base code only returned the global matrix)
- Added code for SVI of Hierarchical Poisson Matrix Factorization [Gopalan et al., 2015].

3.2 Details

In this section we review the SVI update rules for Poisson Matrix Factorization[Schmidt et al., 2009] and Hierarchical Poisson Matrix Factorization[Gopalan et al., 2015].

3.2.1 Poisson Matrix Factorization[Schmidt et al., 2009]

Let the variational distributions approximating the posterior be the following:

- For user u and latent factor k : $q(\theta_{uk}|a^{uk}_{new}, b^{uk}_{new}) =$ Gamma$(a^{uk}_{new}, b^{uk}_{new})$
- For item i and latent factor k : $q(\beta_{ik}|c^{ik}_{new}, d^{ik}_{new}) =$ Gamma$(c^{ik}_{new}, d^{ik}_{new})$

We assume that the latent variables corresponding to the items and users are global and local variables respectively. The update steps for the parameters in this model in SVI setting with minibatch size D are are:

1. Local Variables - Users (Only update the ones corresponding to the mini batch)
   (a) $a^{nk}_{new} = a + \sum_{m=1}^{M} X_{nm}\phi_k^{(nm)}$
   (b) $b^{nk}_{new} = b + \sum_{m=1}^{M} \mathbb{E}_q[\beta_{mk}]$

2. Global Variables - Items (The quantity inside the square bracket of c^{mk}_{new} and d^{mk}_{new} is the time step)
   (a) $c^{mk}_{new}[t] = (1-\rho)c^{mk}_{new}[t-1] + \rho(c + \frac{N}{D}\sum_{n=1}^{N} X_{nm}\phi_k^{(nm)})$
   (b) $d^{mk}_{new}[t] = (1-\rho)d^{mk}_{new}[t-1] + \rho(d + \frac{N}{D}\sum_{n=1}^{N} \mathbb{E}_q[\theta_{nk}])$

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

3.2.2 Hierarchical Poisson Matrix Factorization[Gopalan et al., 2015]

Let the variational distributions approximating the posterior be the following:

- For activity of user u : $q(\xi_u|\kappa_u^{shp}, \kappa_u^{rte}) =$ Gamma$(\kappa_u^{shp}, \kappa_u^{rte})$
- For user u and latent factor k : $q(\theta_{uk}|\gamma_{uk}^{shp}, \gamma_{uk}^{rte}) =$ Gamma$(\gamma_{uk}^{shp}, \gamma_{uk}^{rte})$
- For popularity of item i : $q(\eta_i|\tau_i^{shp}, \tau_i^{rte}) =$ Gamma$(\tau_i^{shp}, \tau_i^{rte})$
- For item i and latent factor k : $q(\beta_{ik}|\lambda_{ik}^{shp}, \lambda_{ik}^{rte}) =$ Gamma$(\lambda_{ik}^{shp}, \lambda_{ik}^{rte})$
- For each user u and item i K auxiliary latent variables are introduced
  $z_{uik} \sim$ Poisson$(\theta_{uk}\beta_{ik})$.
  Variational Distribution for these variables : $q(z_{ui}|\phi_{ui})$ = Multinomial(ϕ_{ui})

We assume that the latent variables corresponding to the items and their popularity to be global variables while latent variables corresponding to users and their activity to be local variables.
The update steps for the parameters in this model in SVI setting with minibatch of size D are:

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 D, update only the relevant local parameters) -
   - $\gamma_{uk}^{shp} = a + \sum_i y_{ui}\phi_{uik}$

Parameter/Model	PMF	SPMF	SPMFAP	GMF
RMSE	1.2963	1.532	1.543	0.266
Accuracy	0.619	0.659	0.671	0.940

Table 1: RMSE and Accuracy comparision of various models for MovieLens Dataset

- $\gamma_{uk}^{rte} = \frac{\kappa_u^{shp}}{\kappa_u^{rte}} + \sum_i \frac{\lambda_{ik}^{shp}}{\lambda_{ik}^{rte}}$
- $\kappa_u^{rte} = \frac{a'}{b'} + \sum_k \frac{\gamma_{uk}^{shp}}{\gamma_{uk}^{rte}}$

5. Global Parameters ($scale =$ (Number of samples)$/|D|$)
   - $\lambda_{ik}^{shp}[t] = (1-\rho)\lambda_{ik}^{shp}[t-1] + \rho(c + scale \times \sum_u y_{ui}\phi_{uik})$
   - $\lambda_{ik}^{rte}[t] = (1-\rho)\lambda_{ik}^{rte}[t-1] + \rho(\frac{\tau_i^{shp}}{\tau_i^{rte}} + scale \times \sum_u \frac{\gamma_{uk}^{shp}}{\gamma_{uk}^{rte}})$
   - $\tau_i^{rte}[t] = (1-\rho)\tau_i^{rte}[t-1] + \rho(\frac{c'}{d'} + scale \times \sum_k \frac{\lambda_{ik}^{shp}}{\lambda_{ik}^{rte}})$

3.3 Experiments

3.3.1 Notations

- RMSE - Root mean squared error
- PMF - Poisson Matrix Factorization Model (Batch VB) [Schmidt et al., 2009]
- SPMF - Poisson Matrix Factorization Model (SVI) [Schmidt et al., 2009]
- SPMFAP - Hierarchical Poisson Matrix Factorization Model (SVI) [Gopalan et al., 2015]
- GMF - Gaussian Matrix Factorization (Batch VB)
- K - Number of Latent Factors
- N - Number of rows in data matrix (corresponding to users)
- M - Number of columns in data matrix (corresponding to items)

3.3.2 Movie Lens 1 Million Dataset

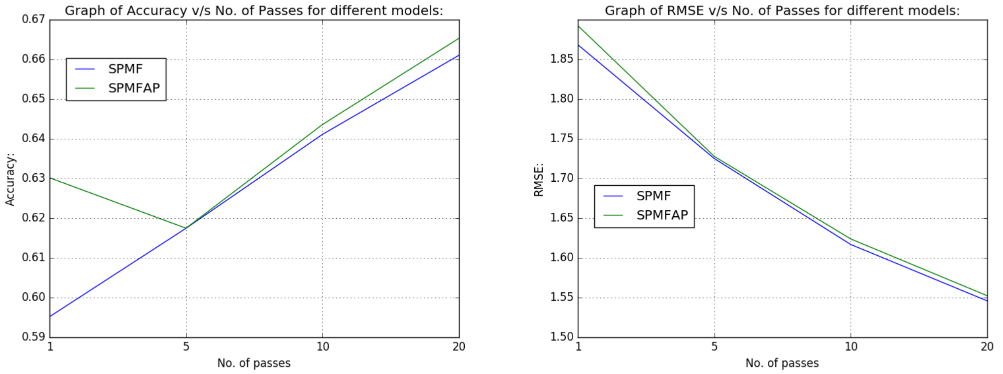
The first experiment we did is on Movie Lens (1M) Dataset. The dataset contains 1,000,209 anonymous ratings of 3,900 movies and 6,040 MovieLens users.

We simulataneously executed Poisson Matrix Factorization (Batch VB as well as SVI), Hierarchical Poisson Matrix Factorization (SVI) and Gaussian Matrix Factorization (VB) on the whole data set. For this we set number of latent factors K to 200. For batch VBs we set maximum number of iteration to 100 and for stochastic VBs, we set number of passes over the data to 5 and number of iteration for updating local parameters in a mini batch to 100.

The result of this experiment is summarized in Table 1.

To get better insight we repeated the experiments in the following settings ($K = 100$ in all cases except where it is varied)-

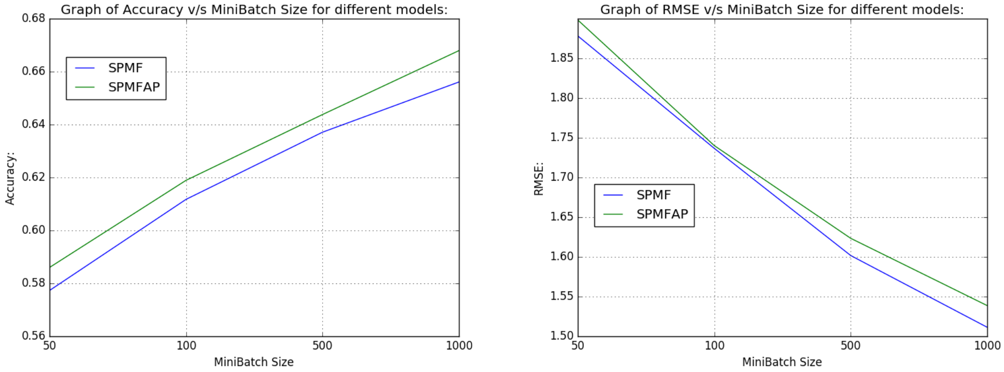
- For stochastic VBs (both PMF and hierarchical PMF) we varied the number of passes over the data and plotted the variation of RMSE and accuracy. The result is plotted in Figure 1.
- For stochastic VBs (both PMF and hierarchical PMF) we varied the size of mini batch and plotted the variation of RMSE and accuracy. The result is plotted in Figure 2.
- For all the models we varied the number of latent factors, K and plotted the variation of RMSE and accuracy. The result is plotted in Figure 3.



(a) Accuracy

(b) RMSE

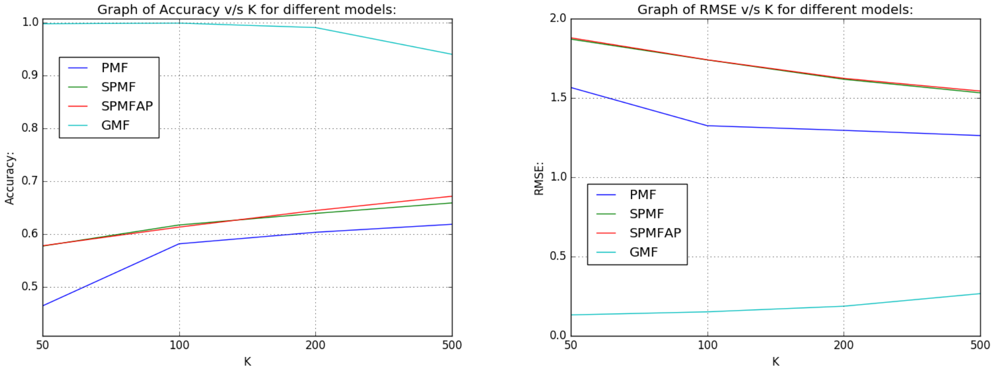
Figure 1: Variation of Accuracy and RMSE with number of passes through the data



(a) Accuracy

(b) RMSE

Figure 2: Variation of Accuracy and RMSE with size of mini batch



(a) Accuracy

(b) RMSE

Figure 3: Variation of Accuracy and RMSE with number of latent factors

Topic-1	Topic-2	Topic-3	Topic-4	Topic-5
united	eng	church	fbi	good
states	launch	religion	crime	players
america	moon	christ	government	win
clinton	project	faith	police	season
president	writes	people	fire	writes
today	flight	christians	weapons	games
years	earth	bible	guns	year
government	nasa	christian	law	play
war	gov	jesus	people	team
american	space	god	gun	game

Table 2: Words from selected topics for PMF with SVI

Topic-1	Topic-2	Topic-3	Topic-4	Topic-5
kill	security	faith	road	teams
crime	government	christ	front	seasons
guns	public	life	engine	players
police	communications	church	dod	win
state	system	christians	speed	hockey
children	keys	christian	good	year
fire	clipper	bible	buy	games
law	encryption	jesus	cars	play
gun	chip	people	bike	team
people	key	god	car	game

Table 3: Words from selected topics for Hierarchical PMF with SVI

3.4 20-Newsgroup Dataset

The original 20-Newsgroup Dataset contains 20,000 documents belonging to 20 categories or newsgroups.
The dataset we used had 11,284 documents and the vocubulary size of the dataset is 2000. The feature matrix X is such that X_{ij} contains the number of times word j appears in document i.
We executed both Poisson Matrix Factorization and Hierarchical Matrix Factorization on this dataset and extracted top 10 words corresponding to each inferred topic. So we used Matrix Factorization as a way to model topics. One can observe that the words in most of the inferred topics make sense (related words appearing together). We took 50 latent variables (i.e. 50 topics) for this experiment. We report some of the best looking topics from both models in Table 2 and Table 3.

4 Conclusion

We started with Literature survey of Stochastic Variational Inference and came across some interesting papers. Some of these papers included LDA topic modelling[Hoffman et al., 2013], Black Box Variational Inference[Ranganath et al., ober] (doesn't require local conjugacy), streaming Variational Bayes[Broderick and Jordan., 2013] (where we don't have a fixed amount of data), adaptive learning rate for SVI [Ranganath et al., 2013] (for faster convergence to better optima) etc.
Finally we focussed our attention to implementation and experimentation task. We used the code given in [Liang et al., 2014] as our base code for Poisson Matrix Factorization and tailored it suit our requirements. We extended it to include SVI for Hierarchical PMF. We executed the codes on two datasets - MovieLens and 20-Newsgroup, and finally reported our results.
From the results of experiments on MovieLens dataset we could observe that when it came to RMSE for MovieLens dataset, ordinary PMF performed better that Hierarchical PMF but the trend reversed for accuracy measure. We also experimented by varying the number of latent factors, no. of full-passes through the data etc (for MovieLens dataset).

Also the topic modelling on 20-NewsGroup dataset using Matrix Factorization worked pretty well. The inferred topics for both PMF and Hierarchical PMF made sense.
We also tried doing some classification task with PMF output as features but we aren't getting good results (probably due to some fault in code) so we haven't reported it.
So we were able to get two flavors in one project. We read and explored some literature on SVI, and we were also able to experience the delights and disappointments in actually working on an SVI model.

References

[Broderick and Jordan., 2013] 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 et al., 2015] Gopalan, P., Hofman, J. M., and Blei, D. M. (2015). Scalable recommendation with hierarchical poisson factorization. In *UAI*, pages 326–335.

[Hoffman et al., 2013] Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. W. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347.

[Liang et al., 2014] Liang, D., Paisley, J., Ellis, D., et al. (2014). Codebook-based scalable music tagging with poisson matrix factorization. In *ISMIR*, pages 167–172. Citeseer.

[Ranganath et al., ober] Ranganath, R., Gerrish, S., and Blei, D. M. (2014, October). Black box variational inference. *AISTATS (pp. 814-822).*

[Ranganath et al., 2013] Ranganath, R., Wang, C., Blei, D. M., and Xing, E. P. (2013). An adaptive learning rate for stochastic variational inference. In *ICML (2)*, pages 298–306.

[Schmidt et al., 2009] Schmidt, M. N., Winther, O., and Hansen, L. K. (2009). Bayesian non-negative matrix factorization. In *International Conference on Independent Component Analysis and Signal Separation*, pages 540–547. Springer.