Your Name
email@fas.harvard.edu
CS281-F17

Assignment #4
Due: 5pm November 13, 2017

Collaborators: Jane Doe

Homework 4: Approximations - Solutions

Grading Instructions

In the solutions, you will see several highlighted checkpoints. These each have a label that corresponds to an entry in the Canvas quiz for this problem set. The highlighted statement should clearly indicate the criteria for being correct on that problem. If you satisfy the criteria for a problem being correct, mark "Yes" on the corresponding position on the Canvas quiz. Otherwise, mark "No". Your homework scores will be verified by course staff at a later date.

That being said, many of the problems in this course will be proofs. If you find a proof that isn't referred to by the course solution, don't worry. If you're uncertain about the proof, make a Piazza post. If you're certain, mark it correct (and we'll look at it during verification).

Introduction

There is a mathematical component and a programming component to this homework. Please submit your PDF and Python files to Canvas, and push all of your work to your GitHub repository. If a question requires you to make any plots, please include those in the writeup.

Solution 1 (30 points)

1. See `q1.py` for the code.

$$p(y=0) = \begin{bmatrix} 1. & 1. & 1. \\ 0.99999998 & 1. & 0.99999998 \\ 0.00252378 & 0.00248464 & 0.00252378 \end{bmatrix} \tag{1}$$

$$p(y=1) = \begin{bmatrix} 5.79905571e-12 & 1.95516261e-15 & 5.79905571e-12 \\ 1.51930154e-08 & 6.56942344e-12 & 1.51930154e-08 \\ 9.97476218e-01 & 9.97515359e-01 & 9.97476218e-01 \end{bmatrix} \tag{2}$$

   Check 1.1 (5 pts): Correct posteriors

2. The lower bound we want to optimize is

$$\mathbb{E}_q[\log p(x,y)] - \mathbb{E}_q[\log q(y)]$$

   where $q(y) = \prod_{i=1}^{H} \prod_{j=1}^{W} q(y_{ij})$. For y_{ij}, denote its neighbors as $\mathcal{N}(ij)$, denote the variational parameters as $q(y_{ij}=k) = \lambda_{ij}^k$, then the relevant terms are:

$$\sum_{k=1}^{K} \lambda_{ij}^k 10 * \delta(x_{ij}=k) + \sum_{n \in \mathcal{N}(ij)} \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \lambda_{ij}^{k_1} \lambda_n^{k_2} \theta(k_1,k_2) - \sum_{k=1}^{K} \lambda_{ij}^k \log \lambda_{ij}^k \tag{3}$$

   where $\theta(k_1,k_2) = 10$ if $k_1 = k_2$, 2 if $|k_1 - k_2| = 1$ and 0 otherwise. Therefore,

$$\lambda_{ij}^k = \frac{1}{Z_{ij}} \exp\left(10 * \delta(x_{ij}=k) + \sum_{n \in \mathcal{N}(ij)} \sum_{k_2=1}^{K} \lambda_n^{k_2} \theta(k, k2)\right) \tag{4}$$

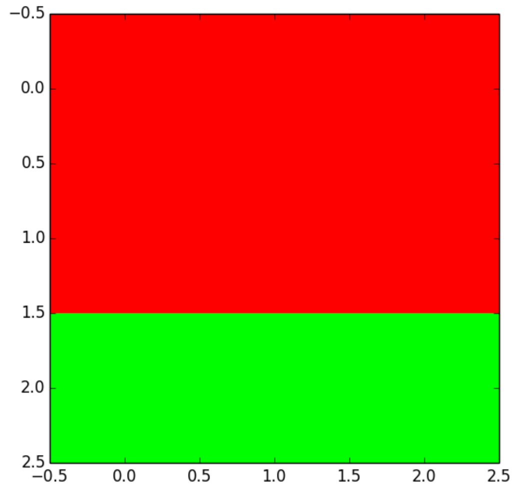
   where $Z_{ij} = \sum_{k=1}^{K} \lambda_{ij}^k$.
   Check 1.2 (5 pts): Correct updates

3. See `q1.py` for implementation. The learned variational parameters are:

$$\lambda^0 = \begin{bmatrix} 1.00000000e+00 & 1.00000000e+00 & 1.00000000e+00 \\ 9.99999985e-01 & 1.00000000e+00 & 9.99999985e-01 \\ 4.53978687e-05 & 1.52521207e-08 & 4.53978687e-05 \end{bmatrix} \tag{5}$$

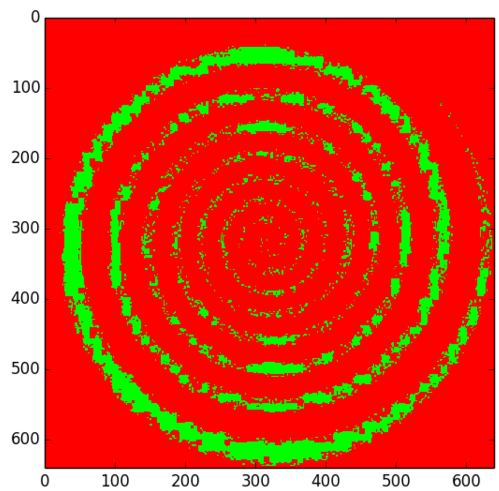
$$\lambda^1 = \begin{bmatrix} 5.10909027e-12 & 1.71390843e-15 & 5.10909027e-12 \\ 1.52189210e-08 & 5.10909027e-12 & 1.52189210e-08 \\ 9.99954602e-01 & 9.99999985e-01 & 9.99954602e-01 \end{bmatrix} \tag{6}$$

   The figure on `small` is



Check 1.3 (10 pts): Correct implementation and figure on `small`

The figure on `spiral` is



4. See code for implementation. The learned beliefs are:

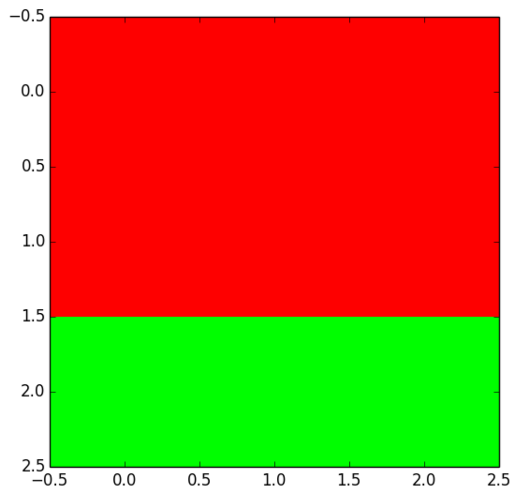
$$belief(0) = \begin{bmatrix} 1. & 1. & 1. \\ 0.99999998 & 1. & 0.99999998 \\ 0.00252378 & 0.00248464 & 0.00252378 \end{bmatrix} \tag{7}$$

$$belief(1) = \begin{bmatrix} 5.79881973e-12 & 1.71418509e-15 & 5.79881973e-12 \\ 1.51930125e-08 & 6.56530103e-12 & 1.51930125e-08 \\ 9.97476217e-01 & 9.97515360e-01 & 9.97476217e-01 \end{bmatrix} \tag{8}$$

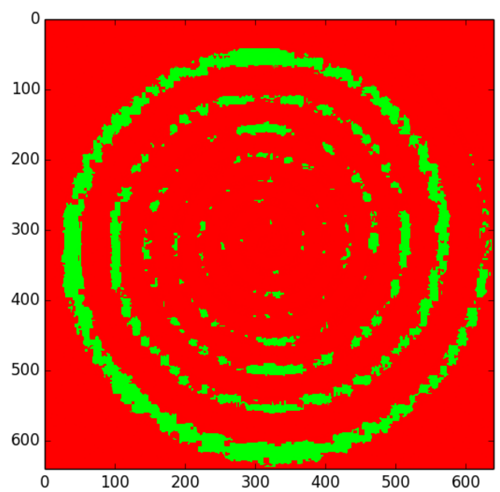
Note that this is very similar to exact marginals

Check 1.4 (10 pts): Correct beliefs and figure on `small`

The figure on `small` is



The figure on `spiral` is



Solution 2 (30 points)

1. KL between two Gaussians $p(x) = \mathcal{N}(x|\mu_1, \sigma_1^2)$ and $q(x) = \mathcal{N}(x|\mu_2, \sigma_2^2)$

$$KL(p||q) = -\int p(x) \log q(x) dx + \int p(x) \log p(x) dx$$

The first term is

$$\begin{aligned}
\int p(x) \log q(x) dx &= -\int p(x) \log \left[\frac{1}{(2\pi\sigma_q^2)^{1/2}} e^{-\frac{(x-\mu_q)^2}{2\sigma_q^2}}\right] dx \\
&= \frac{1}{2}\log(2\pi\sigma_q^2) - \int p(x) - \frac{(x-\mu_q)^2}{2\sigma_q^2} dx \\
&= \frac{1}{2}\log(2\pi\sigma_q^2) + \frac{\int p(x)x^2 dx}{2\sigma_q^2} - \frac{\int p(x) 2x\mu_q dx}{2\sigma_q^2} + \frac{\int p(x)\mu_q^2 dx}{2\sigma_q^2} \\
&= \frac{1}{2}\log(2\pi\sigma_q^2) + \frac{\mathbb{E}[x^2] - 2\mathbb{E}[x]\mu_q + \mu_q^2}{2\sigma_q^2} \\
&= \frac{1}{2}\log(2\pi\sigma_q^2) + \frac{\sigma_p^2 + \mu_p^2 - 2\mu_p\mu_q + \mu_q^2}{2\sigma_q^2} \\
&= \frac{1}{2}\log(2\pi\sigma_q^2) + \frac{\sigma_p^2 + (\mu_p - \mu_q)^2}{2\sigma_q^2}
\end{aligned}$$

The second term is

$$\int p(x) \log p(x) dx = \frac{1}{2}(1 + \log 2\pi\sigma_p^2)$$

Putting both terms together

$$\begin{aligned}
KL(p||q) &= \frac{1}{2}\log(2\pi\sigma_q^2) + \frac{\sigma_p^2 + (\mu_p - \mu_q)^2}{2\sigma_q^2} - \frac{1}{2}(1 + \log 2\pi\sigma_p^2) \\
&= \log\frac{\sigma_q}{\sigma_p} + \frac{\sigma_p^2 + (\mu_p - \mu_q)^2}{2\sigma_q^2} - \frac{1}{2}
\end{aligned}$$

Check 2.1 (3 pts): Correct expression for the KL divergence

2.

$$KL(q_\lambda(U)||p(U)) = \sum_i \sum_k \log\frac{\sigma_p}{\sqrt{\lambda_{ik}^{\sigma^2 U}}} + \frac{\lambda_{ik}^{\sigma^2 U} + (\lambda_{ik}^{\mu U})^2}{2\sigma_p^2} - \frac{1}{2}$$

$$KL(q_\lambda(V)||p(V)) = \sum_j \sum_k \log\frac{\sigma_p}{\sqrt{\lambda_{jk}^{\sigma^2 V}}} + \frac{\lambda_{jk}^{\sigma^2 V} + (\lambda_{jk}^{\mu V})^2}{2\sigma_p^2} - \frac{1}{2}$$

Check 2.2 (2 pts): Correct expression for the KL divergence components of the ELBO.

3. Take samples of u_i and v_j instead of taking an expectation

$$\tilde{u}_i \sim \mathcal{N}(\lambda_i^{\mu U}, \lambda_i^{\sigma^2 U}\mathbf{I})$$
$$\tilde{v}_j \sim \mathcal{N}(\lambda_j^{\mu V}, \lambda_j^{\sigma^2 V}\mathbf{I})$$

Maximize this function with respect to the parameters of the distributions from which u_i and v_j are drawn

$$\mathcal{L}(\lambda) = -KL(q_\lambda(U) \mid\mid p(U)) - KL(q_\lambda(V) \mid\mid p(V)) + \sum_{n=1}^{N} \log\left[\mathcal{N}(r_n|\tilde{u}_i^\top \tilde{v}_j, \sigma_\epsilon^2)\right]$$

Check 2.3 (3 pts): Correct expression for estimating the ELBO via samples. It's also ok to sum over all pairs (i, j) as we didn't specify that the matrix was sparse.

4. Draw noise

$$\begin{aligned}
\tilde{z}_i \in \mathbb{R}^k &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
\tilde{z}_j \in \mathbb{R}^k &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
-D_{qp} &= -KL(q_\lambda(U) \mid\mid p(U)) - KL(q_\lambda(V) \mid\mid p(V)) \\
u_i &= (\sqrt{\lambda_i^{\sigma^2 U}} \odot \tilde{z}_i) + \lambda_i^{\mu U} \\
v_j &= (\sqrt{\lambda_j^{\sigma^2 V}} \odot \tilde{z}_j) + \lambda_j^{\mu V} \\
\mathcal{L}(\lambda) &= -D_{qp} + \sum_{n=1}^{N} \log\left[\mathcal{N}(r_n|u_i^\top v_j, \sigma_\epsilon^2)\right]
\end{aligned}$$

where $\odot$ is the element-wise product

Check 2.4 (5 pts): Correct expression for computing the reparametrized ELBO. It's ok to sample $\tilde{z}_i, \tilde{z}_j$ as individual univariate Gaussians for each vector component, since they're isotropic Gaussians.

5. Our output on running the variational inference:

```
[Epoch 1 ]: ELBO 6.7972 | Time: 15.0038
Train        NLL: 5.2267
Validation   NLL: 5.2283
Test         NLL: 5.1857
[Epoch 2 ]: ELBO 5.0373 | Time: 56.7050
Train        NLL: 4.8471
Validation   NLL: 4.8394
Test         NLL: 4.8039
[Epoch 3 ]: ELBO 4.6998 | Time: 98.4535
Train        NLL: 4.2879
Validation   NLL: 4.2922
Test         NLL: 4.2572
[Epoch 4 ]: ELBO 3.5215 | Time: 137.1728
Train        NLL: 2.7612
Validation   NLL: 2.7889
Test         NLL: 2.7668
[Epoch 5 ]: ELBO 2.4658 | Time: 175.6854
```

```
Train        NLL: 2.1604
Validation   NLL: 2.1938
Test         NLL: 2.1749
[Epoch 6 ]: ELBO 2.0699 | Time: 216.8669
Train        NLL: 1.8965
Validation   NLL: 1.9279
Test         NLL: 1.9178
[Epoch 7 ]: ELBO 1.8799 | Time: 255.7482
Train        NLL: 1.7598
Validation   NLL: 1.7909
Test         NLL: 1.7838
[Epoch 8 ]: ELBO 1.7709 | Time: 293.7794
Train        NLL: 1.6738
Validation   NLL: 1.7052
Test         NLL: 1.7030
[Epoch 9 ]: ELBO 1.7041 | Time: 332.1034
Train        NLL: 1.6170
Validation   NLL: 1.6462
Test         NLL: 1.6479
[Epoch 10]: ELBO 1.6587 | Time: 370.7737
Train        NLL: 1.5779
Validation   NLL: 1.6072
Test         NLL: 1.6087
```

Check 2.5 (7 pts): 2 pts for plotting or outputting the correct quantities (i.e., ELBO on the train set, and NLL over 100 samples of U, V on the training and test sets for every epoch). 5 pts (in addition) for achieving a final train ELBO and test NLL ≤ 2.0. You may receive 2 pts of partial credit on the last 5 points if you did not get ≤ 2.0, but your ELBO and test NLL were on the same order of magnitude. You also receive full credit if you output the total ELBO or log-likelihood over the entire dataset, though you must note this in the partial credit section to explain (and please, show us the computation that checks that this is equivalent to 2.0 as above)..

6. The trajectories should be similar. We found that having $K \geq 3$ did not help significantly, since the increased effect of randomness added more noise to the model than was useful.

Check 2.6 (5 pts): 5 pts for having the required plots, and making a reasonable analysis of the results. If Check 2.5 was incorrect, you may still get full points for this problem by analyzing your results well.

Solution 3 (25 points)

1. First rewrite $p(U|V,R) \propto p(R|U,V)$. We only need to observe one row at a time, because each rating depends on only one row of U and V. Therefore we can define V_i as the rows of V consisting of jokes rated by user i. Define R_i as the vector of R consisting of ratings for user i. Define R_j as the vector of R consisting of ratings for joke j. Using the results from Murphy 4.125, the form of the conditionals is

$$\begin{aligned}
p(x) &= \mathcal{N}(x|\mu_x, \Sigma_x) \\
p(y|x) &= \mathcal{N}(y|Ax+b, \Sigma_y) \\
p(x|y) &= \mathcal{N}(x|\mu_{x|y}, \Sigma_{x|y}) \\
\Sigma_{x|y}^{-1} &= \Sigma_x^{-1} + A^\top \Sigma_y^{-1} A \\
\mu_{x|y} &= \Sigma_{x|y}\left[A^\top \Sigma_y^{-1}(y-b) + \Sigma_x^{-1}\mu_x\right] \\
p(y) &= \mathcal{N}(y|A\mu_x + b, \Sigma_y + A\Sigma_x A^\top)
\end{aligned}$$

So in our case, considering $p(U_i|V,R)$ for example

$$\begin{aligned}
p(x) &= \mathcal{N}(U|0, \sigma_U^2) \\
p(y|Ax+b, \Sigma_y) &= \mathcal{N}(R|U, V, \sigma_\epsilon^2) \\
p(x|y) &= p(U|V,R) \\
\mu_x &= \mu_U = 0 \\
A &= V_i \\
b &= 0
\end{aligned}$$

This gives us

$$\begin{aligned}
\left(\Sigma_i\right)^{-1} &= \frac{1}{\sigma_U^2}\mathbf{I} + \frac{1}{\sigma_\epsilon^2} V_i^\top V_i \\
\mu_i &= \frac{1}{\sigma_\epsilon^2}\Sigma_i V_i^\top R_i \\
\left(\Sigma_j\right)^{-1} &= \frac{1}{\sigma_V^2}\mathbf{I} + \frac{1}{\sigma_\epsilon^2} U_j^\top U_j \\
\mu_j &= \frac{1}{\sigma_\epsilon^2}\Sigma_j U_j^\top R_j
\end{aligned}$$

Such that

$$\begin{aligned}
p(U_i|V,R) &= \mathcal{N}(\mu_i, \Sigma_i) \\
p(V_j|U,R) &= \mathcal{N}(\mu_j, \Sigma_j)
\end{aligned}$$

Check 3.1 (3 pts): Correct conditional mean

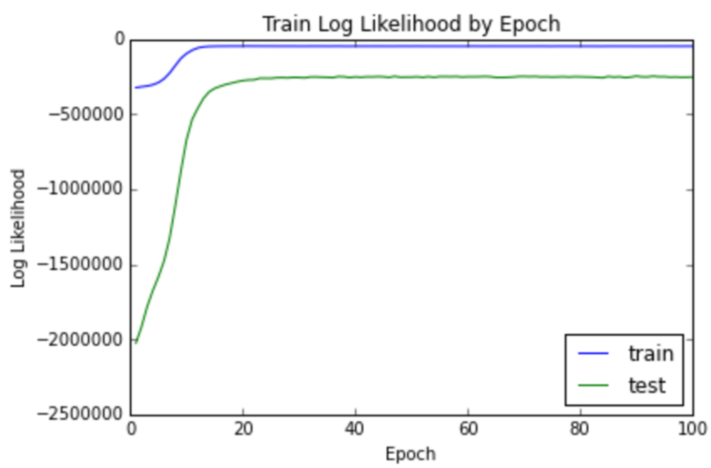
Check 3.2 (2 pts): Correct conditional covariance

2.
```
For some number of epochs
    for all rows i in U
        calculate sigma_u_i
        calculate mu_u_i
        U[i] = sample from N(mu_u_i,sigma_u_i)

    for all rows j in V
        calculate sigma_v_j
        calculate mu_v_j
        V[j] = sample from N(mu_v_j, sigma_v_j)
```

Check 3.3 (5 pts): Correct Gibbs sampling procedure

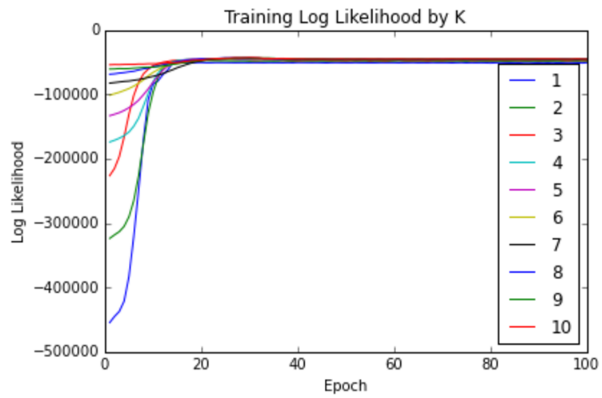
3. The training and test-set log likelihoods are plotted below.

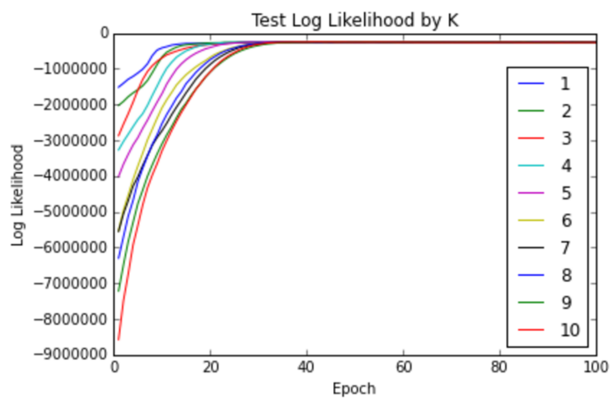


Check 3.4 (2 pts): Plot of Predictive probability for training set. The magnitude may not be the same as in the example figure, but the shapes should be similar.

Check 3.5 (3 pts): Plot of Predictive probability for test set.

4. The log likelihoods for K from 1 to 10 are displayed below. As we can see, the log likelihood steadily increases for every epoch. Moreover, the likelihoods start lower for lower K's, as we would expect; however, as the epochs increase, they appear to converge to similar log likelihoods. The primary difference between our plots and the homework 3 plots are that Gibbs sampling isn't prone to overfitting; even as the number of parameters increase in K, we achieve similar results.





Check 3.6 (5 pts): Plots of Predictive probability for training and test set. They should converge to similar log likelihood for different K.

Check 3.7 (5 pts): Explanation for the differences between Gibbs and MLE.