Kate Lassiter

Problem 1. (25 points)

In the first two problems, you will derive and implement an EM algorithm for the object recommendation problem discussed in class. Here, we have a data set of the form $\mathcal{R} = \{r_{ij}\}$ restricted to a subset of pairs $(i,j) \in \Omega$, where i can range from $1,\ldots,N$ and j can range from $1,\ldots,M$ and we let $r_{ij} \in \{\pm 1\}$. The goal of matrix factorization is to find a low rank approximation of this data.

To this end, let the unknown model variables be $u_i \in \mathbb{R}^d$ and $v_j \in \mathbb{R}^d$ for $i = 1,\ldots,N$ and $j = 1,\ldots,M$. Let $U = \{u_i\}$ and $V = \{v_j\}$. The model and priors are,

$$r_{ij}|U,V \overset{ind}{\sim} \text{Bernoulli}(\Phi(u_i^T v_j/\sigma)), \text{ for all } (i,j) \in \Omega, \qquad u_i \overset{iid}{\sim} N(0,cI), \quad v_j \overset{iid}{\sim} N(0,cI).$$

The function $\Phi(\cdot)$ is the CDF of a standard normal random variable, as discussed in class. (Hint: You will find the probit regression discussion useful for this homework.)

The goal of this homework will be to derive and implement an EM algorithm for maximizing

$$\ln p(\mathcal{R},U,V) = \underbrace{\int q(\phi)\ln\frac{p(\mathcal{R},U,V,\phi)}{q(\phi)}d\phi}_{\mathcal{L}(U,V)} + \int q(\phi)\ln\frac{q(\phi)}{p(\phi|\mathcal{R},U,V)}d\phi,$$

where the auxiliary variable ϕ is used in the probit setup similar to that discussed in class for probit regression: $\phi = \{\phi_{ij}\}$ for $(i,j) \in \Omega$ and $r_{ij} = \text{sign}(\phi_{ij})$, $\phi_{ij} \sim \text{Normal}(u_i^T v_j, \sigma^2)$.[1]

[1] Note: In the data, $r_{ij} \in \{-1,+1\}$ instead of $\{0,1\}$. This won't impact the final derivation.

Problem 2 will focus on implementing the algorithm. To this end:

a) Derive $q(\phi)$ for the EM algorithm. Hint: You will be able to show $q(\phi) = \prod_{(i,j)\in\Omega} q(\phi_{ij})$.

$$\ln p(R, U, V) = \int q(\phi) \ln \frac{p(R, U, V, \phi)}{q(\phi)} d\phi + \int q(\phi) \ln \frac{q(\phi)}{p(\phi| R, U, V)} d\phi$$

<u>Step one</u>: Set $q_t(\phi)=p(\phi|X,\theta_{t-1})$

$q_t(\phi) = p(\phi| R, U, V)$

$$p(\phi| R, U, V) = \frac{\prod_{(i,j) \text{ in } \Omega} p(r_{ij} | u_i,v_j, \phi_{ij}) p(\phi_{ij}| u_i,v_j)}{\prod_{(i,j) \text{ in } \Omega} \int p(r_{ij} | u_i,v_j, \phi_{ij}) p(\phi_{ij}| u_i,v_j) d\phi_{ij}}$$

$$= \prod_{(i,j) \text{ in } \Omega} \frac{p(r_{ij} | u_i,v_j, \phi_{ij}) p(\phi_{ij}| u_i,v_j)}{\int p(r_{ij} | u_i,v_j, \phi_{ij}) p(\phi_{ij}| u_i,v_j) d\phi_{ij}}$$

$$p(\phi| R, U, V) = \prod_{(i,j) \text{ in } \Omega} p(\phi_{ij}| u_i,v_j, r_{ij})$$

Kate Lassiter

We set: $q_t(\phi) = p(\phi| R, U, V)$

Thus:

$$q_t(\phi) = \prod_{(i,j)\ in\ \Omega} p(\phi_{ij}| u_i, v_j, r_{ij})$$

$$\prod_{(i,j)\ in\ \Omega} q_t(\phi_{ij}) = \prod_{(i,j)\ in\ \Omega} q_t(\phi_{ij}| u_i, v_j, r_{ij})$$

$$\boxed{q_t(\phi) = \prod_{(i,j)\ in\ \Omega} q_t(\phi_{ij}| u_i, v_j, r_{ij})}$$

Recall:
$\phi_{ij} \sim Normal(u_i^T v_j, \sigma^2)$
$r_{ij} = sign(\phi_{ij})$

Joint likelihood:
$p(r_{ij}=1,\phi_{ij}| u_i,v_j) = p(r_{ij}=1|\phi_{ij})\ p(\phi_{ij}| u_i,v_j)$

$p(r_{ij}=1,\phi_{ij}| u_i,v_j) = sign(\phi_{ij})\ Normal(u_i^T v_j, \sigma^2)$

$p(r_{ij}=1,\phi_{ij}| u_i,v_j) = sign(\phi_{ij})\ (2\pi\sigma^2)^{-1/2} e^{(-1/2\sigma 2)(\phi ij\ -\ uiTvj)\hat{}2}$

Marginal distribution:

$$\int p(r_{ij}=1,\phi_{ij}| u_i,v_j) d\phi_{ij} = \int_{-inf}^{inf} sign(\phi_{ij})(2\pi\sigma^2)^{-1/2} e^{(-1/2\sigma 2)(\phi ij\ -\ uiTvj)\hat{}2} d\phi_{ij}$$

$$= \int_{0}^{inf} (2\pi\sigma^2)^{-1/2} e^{(-1/2\sigma 2)(\phi ij\ -\ uiTvj)\hat{}2} d\phi_{ij}$$

$$= P(\phi_{ij} > 0)$$

Posterior:

$$p(\phi_{ij}| u_i, v_j, r_{ij}) = \frac{p(r_{ij}=1|\phi_{ij})\ p(\phi_{ij}| u_i,v_j)}{\int p(r_{ij}=1|\phi_{ij})\ p(\phi_{ij}| u_i,v_j) d\phi_{ij}}$$

$$= \frac{\mathbf{I}(sign(\phi_{ij}) = r_{ij})\ e^{(-1/2\sigma 2)(\phi ij\ -\ uiTvj)\hat{}2}}{\int_{-inf}^{inf} \mathbf{I}(sign(\phi_{ij}) = r_{ij})\ e^{(-1/2\sigma 2)(\phi ij\ -\ uiTvj)\hat{}2} d\phi_{ij}}$$

$p(\phi_{ij}| u_i, v_j, r_{ij}) \sim TN(u_i^T v_j, \sigma^2)$

Set $q_t(\phi) = p(\phi_{ij}| u_i, v_j, r_{ij})$

$q_t(\phi_{ij}) = TN(u_i^T v_j, \sigma^2)$

$$\boxed{q_t(\phi) = TN(u^Tv, \sigma^2)}$$

---

b) Derive $\mathcal{L}(U, V)$. You can use the $\mathbb{E}_q[\cdot]$ notation and write the relevant expectation separately.

$$p(R, U, V) = \prod_{(i,j)\ in\ \Omega} p(r_{ij} \mid v_j, u_i) \prod_{i=1}^{N1} p(u_i) \prod_{j=1}^{N2} p(v_j)$$

$$\ln p(R, U, V) = \sum_{(i,j)\ in\ \Omega} \ln p(r_{ij} \mid v_j, u_i) + \sum_{i=1}^{N1} \ln p(u_i) + \sum_{j=1}^{N2} \ln p(v_j)$$

$$\ln p(R, U, V, \phi) = \sum_{(i,j)\ in\ \Omega} \ln p(r_{ij} \mid \phi_{ij})p(\phi_{ij} \mid v_j, u_i) + \sum_{i=1}^{N1} \ln p(u_i) + \sum_{j=1}^{N2} \ln p(v_j)$$

* **Note:** $c = \lambda$

$$L(U, V) = \sum_{(i,j)\ in\ \Omega} \mathbf{E}_q[\ln p(r_{ij}, \phi_{ij} \mid v_j, u_i)] + \frac{-c}{2} u^Tu + \frac{-c}{2} v^Tv + \text{Constant}$$

$$L(U, V) = \sum_{(i,j)\ in\ \Omega} \underline{\mathbf{E}_q[\ln I(sign(\phi_{ij}) = r_{ij})]} + \left(\frac{-1}{2\sigma^2}\right) \mathbf{E}_q[(\phi_{ij} - u_i^Tv_j)]^2 + \frac{-c}{2} u^Tu + \frac{-c}{2} v^Tv + \text{Constant}$$

= 0 *Because it is truncated normal

$$L(U, V) = \sum_{(i,j)\ in\ \Omega} \left(\frac{-1}{2\sigma^2}\right) \mathbf{E}_q[(\phi_{ij} - u_i^Tv_j)]^2 + \frac{-c}{2} u^Tu + \frac{-c}{2} v^Tv + \text{Constant}$$

$$L(U, V) = \sum_{(i,j)\ in\ \Omega} \left(\frac{-1}{2\sigma^2}\right) \mathbf{E}_q[(\phi_{ij} - u_i^Tv_j)(\phi_{ij} - u_i^Tv_j)] + \frac{-c}{2} u^Tu + \frac{-c}{2} v^Tv + \text{Constant}$$

$$L(U, V) = \sum_{(i,j)\ in\ \Omega} \left(\frac{-1}{2\sigma^2}\right) [u_i^Tv_jv_j^Tu - 2u_i^Tv_j\mathbf{E}_q[(\phi_{ij})]] + \frac{-c}{2} u^Tu + \frac{-c}{2} v^Tv + \text{Constant}$$

$$\boxed{L(U, V) = \sum_{(i,j)\ in\ \Omega} \left(\frac{-1}{2\sigma^2}\right) [u_i^Tv_jv_j^Tu - 2u_i^Tv_j\mathbf{E}_q[(\phi_{ij})]] + \frac{-c}{2} u^Tu + \frac{-c}{2} v^Tv + \text{Constant}}$$

---

c) Derive the U, V that optimize $\mathcal{L}(U, V)$. Hint: Derive for one (and therefore all) u_i and v_j.

$$U = \underset{u}{\operatorname{argmax}}\ L(U,V)$$

$$\frac{d}{du} L(U,V) = \sum_{(i,j)\ in\ \Omega} \left(\frac{-1}{2\sigma^2}\right) [2u_i^T v_j v_j^T - 2v_j \mathbf{E}_q[(\phi_{ij})]] - \sum_{i=1}^{N1} cu_i$$

$$\frac{d}{du} L(U,V) = \mathbf{\underline{0}}$$

$$0 = \sum_{(i,j)\ in\ \Omega} \left(\frac{-1}{2\sigma^2}\right)[2u_i^T v_j v_j^T - 2v_j \mathbf{E}_q[(\phi_{ij})]] - \sum_{i=1}^{N1} cu_i$$

$$\sum_{i=1}^{N1} cu_i = \sum_{(i,j)\ in\ \Omega} \left(\frac{-1}{2\sigma^2}\right)[2u_i^T v_j v_j^T] - \sum_{(i,j)\ in\ \Omega} \left(\frac{-1}{2\sigma^2}\right)2v_j \mathbf{E}_q[(\phi_{ij})]$$

$$\sum_{i=1}^{N1} cu_i = \sum_{(i,j)\ in\ \Omega} \left(\frac{-1}{\sigma^2}\right)[u_i^T v_j v_j^T] - \sum_{(i,j)\ in\ \Omega} \left(\frac{-1}{\sigma^2}\right)v_j \mathbf{E}_q[(\phi_{ij})$$

$$\sum_{i=1}^{N1} cu_i = -\sum_{(i,j)\ in\ \Omega} \frac{[u_i^T v_j v_j^T]}{\sigma^2} + \sum_{(i,j)\ in\ \Omega} \frac{v_j \mathbf{E}_q[(\phi_{ij})}{\sigma^2}$$

Consider for one u_i:

* $j\ \varepsilon\ \Omega ui$ **indicates all movies j that that specific user ui has rated.**

$$cu_i = -\sum_{j\ \varepsilon\ \Omega ui} \frac{[u_i^T v_j v_j^T]}{\sigma^2} + \sum_{j\ \varepsilon\ \Omega ui} \frac{v_j \mathbf{E}_q[(\phi_{ij})}{\sigma^2}$$

$$(u_i)^{-1} cu_i = -\sum_{j\ \varepsilon\ \Omega ui} \frac{[u_i^T v_j v_j^T](u_i)^{-1}}{\sigma^2} + \sum_{j\ \varepsilon\ \Omega ui} \frac{v_j \mathbf{E}_q[(\phi_{ij})](u_i)^{-1}}{\sigma^2}$$

$$c = -\sum_{j\ \varepsilon\ \Omega ui} \frac{[v_j v_j^T]}{\sigma^2} + \sum_{j\ \varepsilon\ \Omega ui} \frac{v_j \mathbf{E}_q[(\phi_{ij})]\ (u_i)^{-1}}{\sigma^2}$$

$$c + \sum_{j\ \varepsilon\ \Omega ui} \frac{[v_j v_j^T]}{\sigma^2} = \sum_{j\ \varepsilon\ \Omega ui} \frac{v_j \mathbf{E}_q[(\phi_{ij})]\ (u_i)^{-1}}{\sigma^2}$$

$$\left(c + \sum_{j\ \varepsilon\ \Omega ui} \frac{[v_j v_j^T]}{\sigma^2}\right)\left(\sum_{j\ \varepsilon\ \Omega ui} \frac{v_j \mathbf{E}_q[(\phi_{ij})]}{\sigma^2}\right)^{-1} = (u_i)^{-1} \sum_{j\ \varepsilon\ \Omega ui} \frac{v_j \mathbf{E}_q[(\phi_{ij})]}{\sigma^2} \left(\sum_{j\ \varepsilon\ \Omega ui} \frac{v_j \mathbf{E}_q[(\phi_{ij})]}{\sigma^2}\right)^{-1}$$

$$\left(c + \sum_{j\ \varepsilon\ \Omega ui} \frac{[v_j v_j^T]}{\sigma^2}\right)\left(\sum_{j\ \varepsilon\ \Omega ui} \frac{v_j \mathbf{E}_q[(\phi_{ij})]}{\sigma^2}\right)^{-1} = (u_i)^{-1}$$

$$\boxed{\left(cI + \frac{\sum_{j\,\varepsilon\,\Omega ui} [v_j v_j^T]}{\sigma^2}\right)^{-1} \left(\frac{\sum_{j\,\varepsilon\,\Omega ui} v_j \mathbf{E}_q[(\phi_{ij})]}{\sigma^2}\right) = u_i}$$

* This is symmetric for v, so:

$$\boxed{\left(cI + \frac{\sum_{i\,\varepsilon\,\Omega vj} [u_i u_i^T]}{\sigma^2}\right)^{-1} \left(\frac{\sum_{i\,\varepsilon\,\Omega vj} u_i \mathbf{E}_q[(\phi_{ij})]}{\sigma^2}\right) = v_j}$$

* **Note:** c = λ
* $j\,\varepsilon\,\Omega\,ui$ **indicates all movies j that that specific user u_i has rated.**
* $i\,\varepsilon\,\Omega\,vj$ **indicates all users i that have rated that specific movie v_j**

---

d) Summarize the final EM algorithm in a list of steps that are easy to read, yet detailed enough for someone to be able to implement it. (Be sure to write out $\ln p(\mathcal{R}, U, V)$.)

<u>First calculating lnp(R, U, V):</u>

Recall:

$r_{ij}\,|U, V \sim \text{Bernoulli}(\boldsymbol{\Phi}(u_i^T v_j/\sigma))$

$u_i \sim N(0, cI)$

$v_j \sim N(0, cI)$

$p(u) = \left(\frac{c}{(2\pi)}\right)^{d/2} e^{(-c/2)uTu}$

$p(v) = \left(\frac{c}{(2\pi)}\right)^{d/2} e^{(-c/2)vTv}$

$p(r_{ij}) = p^k(1-p)^{1-k} = \boldsymbol{\Phi}(u_i^T v_j/\sigma)^{rij}(1-\boldsymbol{\Phi}(u_i^T v_j/\sigma))^{1-ri}$ **

** Treating r_{ij} as $\mathbf{I}[((\phi_{ij}) > 0)]$

Joint Likelihood:

$p(u|V, R) = \prod_{i=1}^{N2} p(r_{ij}\,|\,v_j, u)p(u)$

$= \text{Bernoulli}(\boldsymbol{\Phi}(u_i^T v_j/\sigma))\ N(0, cI)$

$p(U,V|R) = p(U)p(V) \prod_{(i,j)\ in\ \Omega} p(r_{ij}\,|\,u_i, v_j)$

$= N(0,cI)\ N(0,cI)\ \text{Bernoulli}(\boldsymbol{\Phi}(u_i^T v_j/\sigma))$

Kate Lassiter

$$p(R, U, V) = \left(\frac{c}{2\pi}\right)^{d/2} e^{(-c/2)u^Tu} \left(\frac{c}{2\pi}\right)^{d/2} e^{(-c/2)v^Tv} \prod_{(i,j)\ in\ \Omega} \boldsymbol{\Phi}(u_i^T v_j/\sigma)^{r_{ij}} (1-\boldsymbol{\Phi}(u_i^T v_j/\sigma))^{1-r_i}$$

Log Joint Likelihood:

$$\ln p(R, U, V) = 2\left(\frac{d}{2}\ln\left(\frac{c}{2\pi}\right)\right) - \frac{c}{2}u^Tu - \frac{c}{2}v^Tv + \sum_{(i,j)\ in\ \Omega} r_{ij}\ln\boldsymbol{\Phi}(u_i^T v_j/\sigma) + \sum_{(i,j)\ in\ \Omega} \mathbf{(1-} r_{ij})\ln(1-\boldsymbol{\Phi}(u_i^T v_j/\sigma))$$

$$\ln p(R, U, V) = -\frac{c}{2}u^Tu - \frac{c}{2}v^Tv + \sum_{(i,j)\ in\ \Omega} r_{ij}\ln\boldsymbol{\Phi}(u_i^T v_j/\sigma) + \sum_{(i,j)\ in\ \Omega} \mathbf{(1-} r_{ij})\ln(1-\boldsymbol{\Phi}(u_i^T v_j/\sigma)) + \text{Constant}$$

$$\ln p(R, U, V) = -\frac{c}{2}u^Tu - \frac{c}{2}v^Tv + \sum_{(i,j)\ in\ \Omega} r_{ij}\ln\boldsymbol{\Phi}(u_i^T v_j/\sigma) + \sum_{(i,j)\ in\ \Omega} \mathbf{(1-} r_{ij})\ln(1-\boldsymbol{\Phi}(u_i^T v_j/\sigma)) + \text{Constant}$$

$$\boxed{\ln p(R, U, V) = -\frac{c}{2}u^Tu - \frac{c}{2}v^Tv + \sum_{(i,j)\ in\ \Omega} r_{ij}\ln\boldsymbol{\Phi}(u_i^T v_j/\sigma) + \mathbf{(1-} r_{ij})\ln(1-\boldsymbol{\Phi}(u_i^T v_j/\sigma)) + \text{Constant}}$$

***Notes:**

** $c = \lambda$

** Constant = $2\left(\frac{d}{2}\ln\left(\frac{c}{2\pi}\right)\right)$

** Treating r_{ij} as $\mathbf{I}[((\phi_{ij}) > 0)]$

Algorithm Summary:

An EM algorithm for probit matrix factorization

1. Initialized U_o and V_0 as ~Normal(0,0.1I)
2. For iteration **t =1, ...,T:**
   a. E – step: calculate the matrix $\mathbf{E}_{qt}[\phi] = (\mathbf{E}_{qt}[\phi_{1j}],..,\mathbf{E}_{qt}[\phi_{N1j}],..,\mathbf{E}_{qt}[\phi_{i1}],...,\mathbf{E}_{qt}[\phi_{iN2}])$, where:
      i. If the true observation r_{ij} is positive (r_{ij}=1):

      $$\mathbf{E}_{qt}[\phi_{ij}] = u_{i\,t-1}^T v_{j\,t-1} + \text{sigma} \times \frac{\boldsymbol{\Phi}'(-u_{i\,t-1}^T v_{j\,t-1}/\text{sigma})}{1-\boldsymbol{\Phi}(-u_{i\,t-1}^T v_{j\,t-1}/\text{sigma})}$$

      ii. If the true observation r_{ij} is negative (r_{ij}=0):

      $$\mathbf{E}_{qt}[\phi_{ij}] = u_{i\,t-1}^T v_{j\,t-1} + \text{sigma} \times \frac{-\boldsymbol{\Phi}'(-u_{i\,t-1}^T v_{j\,t-1}/\text{sigma})}{\boldsymbol{\Phi}(-u_{i\,t-1}^T v_{j\,t-1}/\text{sigma})}$$

         1. Note:
            a. $\boldsymbol{\Phi}'(s)$ = PDF of Normal(0,1) distribution
            b. $\boldsymbol{\Phi}(s)$ = CDF of Normal(0,1) distribution
   b. M – step:
      i. Update the matrix U_t using the equations above in the following equation:
         1. for $u_{i\,t}$ =1,….,N1 update users:

$$\left(cI + \frac{\sum_{j \in \Omega ui} [v_j v_j^T]}{\sigma^2}\right)^{-1}\left(\frac{\sum_{j \in \Omega ui} v_j \mathbf{E}_{qt}[(\phi_{ij})]}{\sigma^2}\right) = u_i$$

* **Note:** c = λ

* $j \in \Omega\, ui$ **indicates all movies j that that specific user ui has rated.**

* **Use all v_j from the current time step = t**

ii. Update the matrix V_t using the equations above in the following equation:

1. for $v_{j\,t}$=1,….,N2 update objects:

$$\left(cI + \frac{\sum_{i \in \Omega vj} [u_i u_i^T]}{\sigma^2}\right)^{-1}\left(\frac{\sum_{i \in \Omega vj} u_i \mathbf{E}_{qt}[(\phi_{ij})]}{\sigma^2}\right) = v_j$$

* **Note:** c = λ

* $i \in \Omega\, vj$ **indicates all users i that have rated that specific movie vj.**

* **Use all u_i from the current time step = t**

c. Calculate ln(R,U,V) using the equation

$$\ln p(R, U_t, V_t) = -\frac{c}{2} u_t^T u_t - \frac{c}{2} v_t^T v_t + \sum_{(i,j)\, in\, \Omega} r_{ij} \ln \Phi(u_{it}^T v_{jt}/\sigma) + (1 - r_{ij}) \ln(1 - \Phi(u_{it}^T v_{jt}/\sigma)) + \text{Constant}$$

Notes:

** c = λ

** Constant = $2\left(\frac{d}{2} \ln\left(\frac{c}{2\pi}\right)\right)$

** Treating r_{ij} as $\mathbf{I}[((\phi_{ij}) > 0)]$

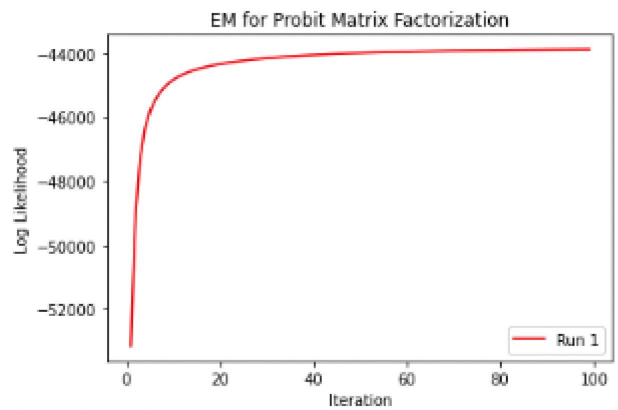
Kate Lassiter

Problem 2. (25 points)

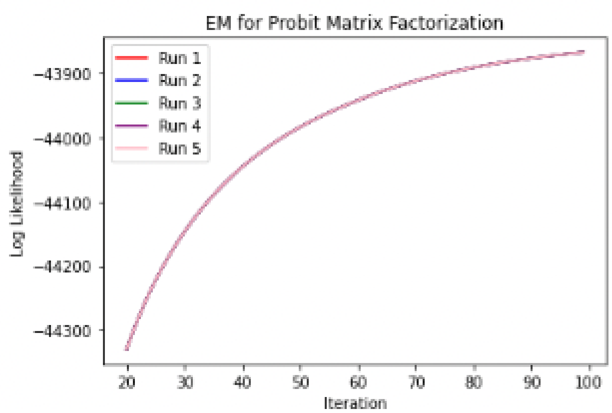
In this problem, you will implement the EM algorithm you derived in Problem 1 for finding a local optimal solution to $\ln p(\mathcal{R}, U, V)$. The data is provided on Courseworks along with a README file explaining the data. For your implementation, set $d = 5$, $c = 1$ and $\sigma^2 = 1$. Treat the users as U and the movies as V when implementing the algorithm.[2]

You will need to initialize your algorithm in some way. One possibility is to start by generating each u_i and v_j from Normal$(0, 0.1I)$. Please use this method for this assignment.

a) Run your algorithm for 100 iterations and plot $\ln p(\mathcal{R}, U, V)$ for iterations 2 through 100.



b) Rerun your algorithm for 100 iterations using 5 different random starting points. Plot the 5 different objective functions for iterations 20 through 100. Note: This is simply a repeat of Problem 2(a), only showing 5 objective functions instead of one and changing the x-axis.



c) Predict the values given in the test set and show your results in a confusion matrix. Show the raw counts in this confusion matrix.

Kate Lassiter

Per Run:

	-1	1
-1	1424	619
1	842	2115

prediction accuracy: 0.7078

	-1	1
-1	1419	600
1	847	2134

prediction accuracy: 0.7106

	-1	1
-1	1430	603
1	836	2131

prediction accuracy: 0.7122

	-1	1
-1	1433	601
1	833	2133

prediction accuracy: 0.7132

	-1	1
-1	1431	594
1	835	2140

prediction accuracy: 0.7142

Total:

	-1	1
-1	7137	3017
1	4193	10653

prediction accuracy: 0.7116

Problem 3. (25 points)

In this problem, you will implement the MCMC sampling approach discussed in Lecture 3 on the data from Problem 2. In this scenario, treat r_{ij} as a real-valued observation as described in the notes. Therefore, ϕ_{ij} does not appear in this problem. This is not ideal, but can still produce useful results. Again set $d = 5$, $c = 1$ and $\sigma^2 = 1$. Initialize U and V to all zeros. (It's worth thinking why this initialization wouldn't work for EM, but works for MCMC.) Since the algorithm was already derived in the notes, you don't need to rederive the Gibbs sampler.

a) Write out the log joint likelihood for the model (for any d, c and σ^2), showing all steps in how you arrived at the final mathematical equation.

Recall:

$r_{ij} \sim \text{Normal}(u_i^T v_j, \sigma^2) = (2\pi\sigma^2)^{-1/2} e^{(-1/2\sigma 2)(\phi ij - uiTvj)^{\wedge}2}$

$u_i \sim \text{Normal}(0, cI) = \frac{(c)^{(d/2)}}{(2\pi)} e^{(-c/2)uTu}$

$v_j \sim \text{Normal}(0, cI) = \frac{(c)^{(d/2)}}{(2\pi)} e^{(-c/2)vTv}$

Joint likelihood:

Kate Lassiter

$$p(R, U, V) = \prod_{(i,j)\ in\ \Omega} p(r_{ij} \mid v_j, u_i) \prod_{i=1}^{N1} p(u_i) \prod_{j=1}^{N2} p(v_j)$$

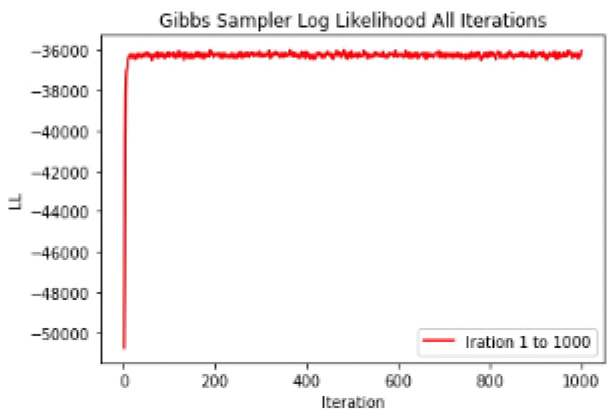
$$\text{lnp}(R, U, V) = \sum_{(i,j)\ in\ \Omega} ln p(r_{ij} \mid v_j, u_i) + \sum_{i=1}^{N1} \text{lnp}(u_i) + \sum_{j=1}^{N2} ln p(v_j)$$

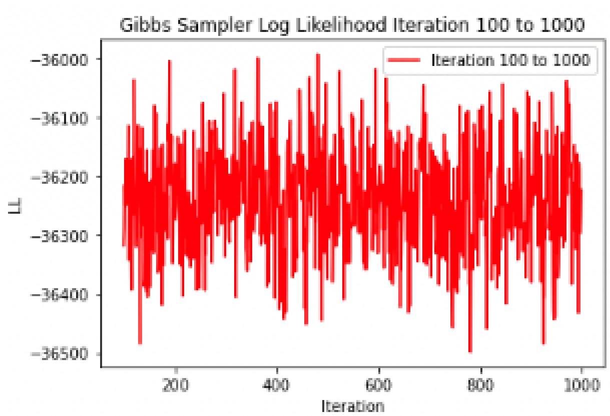
$$\text{lnp}(R, U, V) = \sum_{(i,j)\ in\ \Omega} ln[(2\pi\sigma^2)^{-1/2} e^{(-1/2\sigma 2)(\phi ij\ -\ uiTvj)\text{^}2}] + \sum_{i=1}^{N1} \text{ln}[(\frac{c}{(2\pi)})^{(d/2)} e^{(-c/2)uTu}] + \sum_{j=1}^{N2} ln[(\frac{c}{(2\pi)})^{(d/2)} e^{(-c/2)vTv}]$$

$$\boxed{\text{lnp}(R, U, V) = -\sum_{(i,j)\ in\ \Omega} \frac{1}{2\sigma^2}(r_{ij} - u_i^T v_j)^2 - \sum_{i=1}^{N1} \frac{c}{2} u_i^T u_i - \sum_{j=1}^{N2} \frac{c}{2} v_j^T v_j + \text{Constant}}$$

---

b) Run your code for 1000 iterations. In two separate figures, plot the log joint likelihood for all iterations, and the log joint likelihood for iterations 100 to 1000.





---

c) Using samples collected after every 25 iterations (starting with iteration 100), calculate a Monte Carlo approximation to $\mathbb{E}[r_{ij}|R]$ for each value in the test set. Predict $+1$ if this expectation is positive, and -1 otherwise. Show your test results in a confusion matrix as you showed them in Problem 2(c).

Last Iteration:

```
--  ----  ----
      -1     1
-1  1435   550
1    831  2184
--  ----  ----
prediction accuracy: 0.7238
```

Total Across Iterations:

```
--  -----  -----
       -1      1
-1  52905  20270
1   30937  80888
--  -----  -----
prediction accuracy: 0.7232054054054055
```