

```

%-----
% Tony Hyun Kim
% CS 246, PS #3, Problem 4(b)
% Latent factor implementation
%-----
clear all;

% Data parameters
%-----
m = 1682; % Number of movies
n = 943; % Number of users

% Algorithm parameters
%-----
k = 20; % Latent factor dimension
lam = 0.2; % Regularization
eta = 0.001; % Learning rate

% User/item representation in latent space
%-----
Q = rand(m,k); % Movie
P = rand(n,k); % User

Niter = 40;
E = zeros(1,Niter);
source = 'ratings.train.txt';
counter = 0;
verbose = 0;
for iter = 1:Niter
    % We are "pretending" that the data is sufficiently large so
    % that we must stream the entries of R and perform stochastic
    % gradient descent.
    tic;
    fid = fopen(source);
    tline = fgetl(fid);

    while ischar(tline)
        counter = counter + 1;

        % Parse a single line of data from file in the format
        % [UserID MovieID Rating]
        R = sscanf(tline,'%d %d %d');

        % Stochastic update equations
        p_old = P(R(1),:);
        q_old = Q(R(2),:);
        error = R(3)-q_old*p_old';

        if(verbose)
            fprintf('Iter %3d, counter %09d: user %04d movie %04d rating %d error %02.3l
f\n',...
                    iter,counter,R(1),R(2),R(3),error);
        end

        if(isnan(error))
            fprintf('Detected divergence after %d stochastic updates!\n',counter);
            return;
        end

        p_new = p_old + eta*(error*q_old - lam*p_old);
        q_new = q_old + eta*(error*p_old - lam*q_old);

        P(R(1),:) = p_new;
        Q(R(2),:) = q_new;

        tline = fgetl(fid); % Read next line
    end
end
fclose(fid);

```

```
t1 = toc;

% Compute the error (including regularization)
tic;
fid = fopen(source);
tline = fgetl(fid);
while ischar(tline)
    R = sscanf(tline, '%d %d %d');
    p = P(R(1),:);
    q = Q(R(2),:);
    E(iter) = E(iter) + (R(3)-q*p')^2;
    tline = fgetl(fid);
end
fclose(fid);

% Regularization terms
for u = 1:n
    p = P(u,:);
    E(iter) = E(iter) + lam*(p*p');
end
for i = 1:m
    q = Q(i,:);
    E(iter) = E(iter) + lam*(q*q');
end
t2 = toc;

fprintf('Iteration %d: Elapsed time %.1f s + %.1f s = %.1f s\n', ...,
        iter, t1, t2, t1+t2);
end

% Automatically save
savename = sprintf('eta%01d_%04d.mat', floor(eta), floor(rem(eta,1)*10000));
save(savename);
```