0.1 Question 1: Using Linear Algebra for Optimization

In recommender system module, low-rank matrix factorization was used to execute latent factor modeling of movie ratings data.

Specifically, we calculated matrices U and V to solve the following optimization problem (if all ratings were given):

$$\min_{U,V} f(U,V) = \min_{U,V} \|R - VU^T\|_F^2 = \min_{U,V} \left\{ \sum_{m=1}^{M} \sum_{i=1}^{I} I_{mi}(r_{mi} - v_m u_i^T)^2 \right\},$$

where

$$I_{mi} = \begin{cases} 1, \text{ if } r_{mi} \text{ is observed} \\ 0, \text{ if } r_{mi} \text{ is missing.} \end{cases}$$

The best U and V were calculated iteratively by improving on current estimates:

$$u_i^{\text{new}} = u_i + 2\alpha(r_{mi} - v_m u_i^T) \cdot v_m$$
$$v_m^{\text{new}} = v_m + 2\alpha(r_{mi} - v_m u_i^T) \cdot u_i,$$

where α is the step-size that is to be chosen by the user. (We won't discuss the role in this class, but treat it as an arbitrary, but given, parameter)

We can make calculating the updates more efficient by calculating them with matrix operations. For example, instead of calculating each deviation $\gamma_{mi} = r_{mi} - v_m u_i^T$ separately for all $m = 1, 2, \ldots, M$ and $i = 1, 2, \ldots, I$, matrix Γ of all deviations can be computed together using matrix operation *(verify for yourself)*:

$$\Gamma = R - VU^T$$

Similarly, updating U and V can be combined into matrix calculations which makes the optimization procedure more efficient.

First, note that updates for u_i, $i = 1, 2, \ldots, I$ can be rewritten as

$$\begin{aligned} u_1^{\text{new}} &= u_1 + 2\alpha\gamma_{m1} \cdot v_m \\ u_2^{\text{new}} &= u_2 + 2\alpha\gamma_{m2} \cdot v_m \\ \vdots \qquad & \qquad \vdots \\ u_I^{\text{new}} &= u_I + 2\alpha\gamma_{mI} \cdot v_m. \end{aligned}$$

Stacking all I equations into a matrix form,

$$U^{\text{new}} = U + 2\alpha\Gamma_{m-}^T v_m,$$

where Γ_{m-} is the m-th row of Γ (use the notation Γ_{-i} for the i-th column).

Note that there are M such update equations (one for each $m = 1, 2, \ldots, M$) that can also be combined into one matrix update equation involving matrices U, V, Γ and scalars. As stated earlier, since α is assumed to be an arbitrary step-size parameter, we can replace α/M with α.

0.1.1 Question 1a: Using Linear Algebra for Optimization

Complete the following update equations:

$$U^{\text{new}} = U + 2\alpha[\text{some function of } \Gamma][\text{some function of } V]$$
$$V^{\text{new}} = V + 2\alpha[\text{some function of } \Gamma][\text{some function of } U]$$

$U^{\text{new}} = U + 2\alpha\Gamma^T V$
$V^{\text{new}} = V + 2\alpha\Gamma U$

By referring back to the function used to calculate the quantities in each figure, describe what each figure is showing and interpret the behavior of the optimization algorithm.

The rmse graph is showing that as our number of iterations increases, the root mean squared error decreases significantly. This shows that the optimization algorithm is working and growing more efficient and accurate with each update.

The max residual change graph shows a significant decrease over iterations as well, pointing to the effectiveness of the optimization algorithm.

The max update graph shows a more unique shape, with a lot of fluctuation in the first 750 iterations, a significant drop in max update, and then sudden peaks around 2,000-2,500 and 3,500-4,000 iterations. This means that every so often, the algorithm encounters data that is significantly different from the rest of the data, and so requires a much bigger update. The fact that it is accounting for these outliers/influential points shows that the algorithm is working properly.

Consider the above plot. By reading the code, comment on what the plot is illustrating. What happens when you add `counts=True` to `transform_density`? What can you conclude?

The plot illustrates the density of 1-, 2-, 3-, 4-, and 5-star ratings on 100 movies by 943 users. The bars grow much taller when we add `counts=True` *— the scale on the density axis changes from 0.5 to 1,000. However, the bars for observed = {1, 2} are much shorter and can hardly be analyzed. It seems like 3-5 star ratings are far more common than 1- or 2-star ratings.*

0.1.2 Question 1f: Make Recommendation

What movies would you recommend to `user id` 601? Do you see any similarities to movies the user rated high?

```
In [148]: rates = Rbig.iloc[:, 600].dropna().sort_values(ascending=True)
          print(rates)
```

```
movie id  movie title
58        Quiz Show (1994)                          1.0
15        Mr. Holland's Opus (1995)                 1.0
82        Jurassic Park (1993)                      1.0
71        Lion King, The (1994)                     1.0
39        Strange Days (1995)                       1.0
96        Terminator 2: Judgment Day (1991)         2.0
98        Silence of the Lambs, The (1991)          3.0
69        Forrest Gump (1994)                       3.0
99        Snow White and the Seven Dwarfs (1937)    3.0
8         Babe (1995)                               3.0
47        Ed Wood (1994)                            3.0
21        Muppet Treasure Island (1996)             3.0
12        Usual Suspects, The (1995)                3.0
56        Pulp Fiction (1994)                       3.0
64        Shawshank Redemption, The (1994)          4.0
65        What's Eating Gilbert Grape (1993)        4.0
22        Braveheart (1995)                         4.0
87        Searching for Bobby Fischer (1993)        4.0
9         Dead Man Walking (1995)                   4.0
100       Fargo (1996)                              4.0
50        Star Wars (1977)                          5.0
91        Nightmare Before Christmas, The (1993)    5.0
Name: (rating, 601), dtype: float64
```

The user seems to enjoy dramas and specifically crime dramas consistently, but surprisingly enjoyed The Nightmare Before Christmas and Star Wars despite them not fitting that genre. I would recommend more crime dramas as well as the rest of the Star Wars movies.

0.2 Question 2 (PSTAT 234): Improving the Model

0.2.1 Question 2a: Logistic function

Note the reconstructed ratings can be smaller than 1 and greater than 5. To confine ratings to between the allowed range, we can use the logistic function. Logistic function is defined as

$$h(x) = \frac{1}{1 + e^{-x}}.$$

It is straightforward to show the derivative is

$$h'(x) = \frac{e^{-x}}{(1 + e^{-x})^2} = h(x)(1 - h(x)).$$

Therefore, we can rescale the ratings from $r_{mi} \in [1, 5]$ to $r_{mi} \in [0, 1]$. Then, we can find the best U and V to optimize the following:

$$\min_{U,V} \|R - h(VU^T)\|_F^2 = \sum_{m,i} I_{mi}(r_{mi} - h(v_m u_i^T))^2,$$

where function h is applied elementwise and

$$I_{mi} = \begin{cases} 1, \text{ if } r_{mi} \text{ is observed} \\ 0, \text{ if } r_{mi} \text{ is missing.} \end{cases}$$

Derive new update expressions for the new objective function.

Type your answer here, replacing this text.

Consider the above plot. By reading the code, comment on what the plot is illustrating. How does this plot look different than part 1.e?

Type your answer here, replacing this text.