Collaborative Filtering

Case Study 4: Collaborative Filtering

Goal: Find movies of interest to a user based on movies watched by the user and others

Methods: matrix factorization, GraphLab
Cold-Start Problem

- **Challenge:** Cold-start problem (new movie or user)
- **Methods:** use features of movie/user
Netflix Prize

- Given 100 million ratings on a scale of 1 to 5, predict 3 million ratings to highest accuracy.
- 17770 total movies
- 480189 total users
- Over 8 billion total ratings
- How to fill in the blanks?

Matrix Completion Problem

- Filling missing data?
- $X_{ij}$ known for black cells
- $X_{ij}$ unknown for white cells
- Rows index users
- Columns index movies
- Well-approx. by rank $k$ matrix $L$
Interpreting Low-Rank Matrix Completion (aka Matrix Factorization)

\[ X = L R' \]

Identifiability of Factors

- If \( r_{uv} \) is described by \( L_u, R_v \), what happens if we redefine the "topics" as \( \tilde{L}_u = L_u M \), \( \tilde{R}_v = R_v M \) where \( M = I \) (orthonormal matrix).

- Then,
  \[ \tilde{L}_u \cdot \tilde{R}_v = L_u M M^T R_v = L_u R_v = r_{uv} \]
  Invariant to orthonormal transformations.
Matrix Completion via Rank Minimization

- Given observed values: $(u, v, r_{uv}) \in X$ some $r_{uv} = ?$
- Find matrix $\Theta$
- Such that: $\Theta_{uv} = r_{uv}$ \forall $r_{uv} \neq ?$ \leftarrow all obs. ratings fit $r_{uv}$ perfectly
- But… want $\Theta$ to be low-rank
- Introduce bias: $\min \text{rank}(\Theta)$ $\Theta$ s.t. $\Theta_{uv} = r_{uv}$ \forall $r_{uv} \neq ?$ $\leftarrow \text{for } k < \min(\text{rank})$
- Two issues: NP-hard $\langle \text{you can’t hope to get exact matching}$

Approximate Matrix Completion

- Minimize squared error:
  - (Other loss functions are possible)
    $\min \sum (\Theta_{uv} - r_{uv})^2$
  - Choose rank $k$:
    $\Theta = \mathbb{L} \mathbb{R}^k$
- Optimization problem:
  $\min_{\mathbb{L}, \mathbb{R}} \sum_{r_{uv}} (\mathbb{L} u \cdot \mathbb{R} v - r_{uv})^2$
  non-convex opt. problem … local optima only
Coordinate Descent for Matrix Factorization

\[
\min_{L,R} \sum_{(u,v): r_{uv} \neq ?} (L_u \cdot R_v - r_{uv})^2
\]

- Fix movie factors, optimize for user factors.

First observation:

\[
\min_L \sum_u \sum_{r_{uv} \neq ?} (L_u \cdot R_v - r_{uv})^2 = \text{set of movies user } u \text{ rated}
\]

\[
\min_L \sum_u \sum_v (L_u \cdot R_v - r_{uv})^2 = \text{ind. opt. problem for each user}
\]

\[
\sum_u \min_L \sum_v (L_u \cdot R_v - r_{uv})^2 = \text{data parallel problem}
\]

Minimizing Over User Factors

- For each user \( u \):

\[
\min_{L_u} \sum_{v \in V_u} (L_u \cdot R_v - r_{uv})^2
\]

- In matrix form:

\[
\begin{bmatrix}
V_u & 1
\end{bmatrix}
\begin{bmatrix}
L_u
\end{bmatrix}
- W
\begin{bmatrix}
1
\end{bmatrix}
\begin{bmatrix}
r_{uv}
\end{bmatrix}
\right) \left( \begin{bmatrix}
V_u^T
\end{bmatrix}
\begin{bmatrix}
L_u
\end{bmatrix}
- W
\begin{bmatrix}
1
\end{bmatrix}
\begin{bmatrix}
r_{uv}
\end{bmatrix}
\right) \right)^2
= \| X \beta - y \|_2^2
\]

Think of as normal LS problem

- Matrix inversion
- Gradient methods

Second observation: Solve by