Rating Estimation in MD Recommender Systems
Combined Reduction-Based and Traditional CF Approaches
Multi-Level Rating Estimation Problem

Implementation and Evaluation of MD Approach
Experimental Setup
Evaluating the Reduction-Based Approach
Performance Metric

Will help to determine which method performs better

- MAE is an example of statistical accuracy measure
- F-measure is an example of decision-support accuracy metric
- The latter suit recommender systems better

\( \mu_{A,X}(Y) \) is an **abstract performance metric** where

- \( A \) – a recommendation algorithm
- \( X \) – training set of known ratings
- \( Y \) – evaluation set of known ratings, where \( X \cup Y = \emptyset \)
Performance Metric II

For each \( d \in Y \)
- \( d.R \) is user-specified rating for that data point
- \( d.R_{A,X} \) is rating predicted by algorithm \( A \) trained on \( X \)

Then \( \mu_{A,X}(Y) \) for MAE is defined as

\[
\mu_{A,X}(Y) = \frac{1}{|Y|} \sum_{d \in Y} |d.R_{A,X} - d.R|
\]

We assume that \( A \) is a traditional collaborative filtering method
Combined Approach

1. Use known ratings to **determine contextual segments** that outperform traditional CF method (offline)

2. **Predict the rating** by using the best contextual segment and the 2D recommendation algorithm (real-time)
Determining High-Performing Contextual Segments

**Input:** $T, R_{A,T}, \mu, N$

**Output:** $SEGM(T)$ - a set of contextual segments on which the reduction-based approach on algorithm A outperforms the pure algorithm A.

**Algorithm:**

1. Let $SEGM(T)$ initially be the set of all **large** contextual segments for the set or ratings $T$.

2. For each segment $S \in SEGM(T)$ compute $\mu_{A,S}(S)$ and $\mu_{A,T}(S)$. Keep only those for which $\mu_{A,S}(S) \gg \mu_{A,T}(S)$

3. Discard $S \in SEGM(T)$ for which $\exists Q : S \subset Q, \mu_{A,Q}(Q) > \mu_{A,S}(S)$
Estimating the Rating

Input: \( SEGM(T) = \{S_1, \ldots, S_k\} \) where \( \mu_{A,S_i}(S_i) \geq \mu_{A,S_j}(S_j), i \geq j \)

\( d \) – data point for which we want to estimate the rating

Output: \( d.R \) – estimated rating for \( d \).

Algorithm:
A picture from paper goes here
Outline

Rating Estimation in MD Recommender Systems
  Combined Reduction-Based and Traditional CF Approaches
  Multi-Level Rating Estimation Problem

Implementation and Evaluation of MD Approach
  Experimental Setup
  Evaluating the Reduction-Based Approach
Multi-Level Rating Estimation Problem

Estimate unknown individual ratings in terms of known aggregate and known individual ratings.

- $R(JD, action) = 6$
- $R(JD, Gladiator) = 7$
- $R(JD, Matrix) = 3$

$\Rightarrow R(JD, OtherMovie) = ?$
More Formal Definition

Assume

- \( R_a(JD, action) \) actual rating assigned by JD himself
- \( R_c(JD, action) \) rating computed from individual ratings \( R(JD, x) \)
- \( X_r \) a set of action movies that John has already rated
- \( X_{nr} \) a set of yet unrated action movies, \( X_r \cup X_{nr} = action \)

Assign ratings to \( R(JD, x), x \in X_{nr} \) to minimize

\[ |R_a(JD, action) - R_c(JD, action)| \]

- There might be infinite number of solutions
Linear Example

Assume

- $AVG$ is the aggregation function
- $X_{nr} = \{y_1, \ldots, y_k\}$

Then we want to find $R(JD, y_1), \ldots, R(JD, y_k)$ s.t.

- $R(JD, y_1) + \ldots + R(JD, y_k) = c$
- $c = (|X_r| + |X_{nr}|) \cdot R_a(JD, action) - \sum_{x \in X_r} R(JD, x)$
Another Reason for Aggregate Hierarchies

Under some assumptions

- The estimation error for aggregate rating is smaller than the estimation error for individual ratings.

The assumptions are

- the rating estimation function $R_c(u, i) = R_a(u, i) + \varepsilon(\mu, \sigma^2)$
- the rating aggregation function (AVG)
- the accuracy measure (MAE)

The general case is an open research question.
Outline

Rating Estimation in MD Recommender Systems
Combined Reduction-Based and Traditional CF Approaches
Multi-Level Rating Estimation Problem

Implementation and Evaluation of MD Approach
Experimental Setup
Evaluating the Reduction-Based Approach
Setup for Evaluation

- Implement a movie RS
- Unlike MovieLens add contextual information
  - *When, Where, With whom*
- No aggregation hierarchies
  - Show that contextual information does matter
  - Reduction-based approach does not fit with aggregation hierarchies
  - Use of hierarchies requires larger amounts of more detailed data
Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach

Implementation and Evaluation of MD Approach

Experimental Setup

Contextual Dimensions

- **Time** (*Weekday, weekend, don’t remember*)
  - If seen on weekend, was it opening weekend for the movie (*yes, no, don’t remember*)

- **Place** (*cinema, home, don’t remember*)

- **Companion** (*alone, with friends, boyfriend/girlfriend, family, others*)
Data Collection

- Rate movies from 1 to 13
- Participants were 117 college students
- 1755 ratings entered over a period of 12 months
- Those who rated fewer than 10 movies were dropped out
- Finally 1457 ratings from 62 students for 202 movies
- 10% - evaluation dataset ($D_E$)
- 90% - modelling dataset ($D_M$)
Significance of Dimensions

Which dimensions make significant difference in rating estimations?

- Partition ratings into categories, e.g. *Time* and *Place*
- Compute average rating per student in each category
- Apply a paired comparison test (t-test)

All dimensions appear to be significant.
Outline

Rating Estimation in MD Recommender Systems
Combined Reduction-Based and Traditional CF Approaches
Multi-Level Rating Estimation Problem

Implementation and Evaluation of MD Approach
Experimental Setup
Evaluating the Reduction-Based Approach
Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach

- Implementation and Evaluation of MD Approach
- Evaluating the Reduction-Based Approach