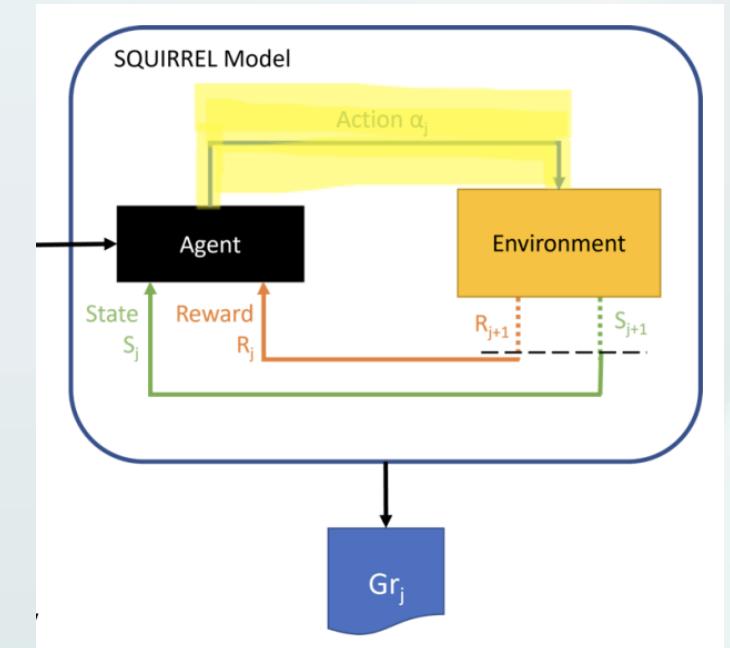


Sequential Group Recommendation

PFA: Proportional Fairness Aggregation (NEW METHOD)

- Students: Oskari Perikangas, Xiaosi Huang
- Course: DATA.ML.360-2025-2026-1 Recommender Systems
- Date: December 5th, 2025



PFA Theory

Proportional Fairness Aggregation (PFA)

Core Innovation: Geometric Mean Aggregation

SDAA: $\text{score} = (w_1 \cdot p_1 + w_2 \cdot p_2 + w_3 \cdot p_3) / \sum w$ [Arithmetic]

aggregate preferences in a way that overlooks individual dissatisfaction.

PFA: $\text{score} = (p_1^{w_1} \times p_2^{w_2} \times p_3^{w_3})^{(1/\sum w)}$ [Geometric]

more sensitive to low values, reveal and address unfairness across group members.

Why Geometric Mean?

- Automatically penalizes imbalanced predictions
- Example: predictions [2.0, 5.0, 4.0]
 - Arithmetic: 3.67 (hides imbalance)
 - Geometric: 3.42 (lower score reflects unfairness)

Two-Phase Design:

1. Phase 1: Average aggregation filters top candidates (quality)
2. Phase 2: Geometric mean selects final items (fairness)

Dynamic Weights:

Gap-based adjustment: users below average get higher weights

Self-correcting over rounds : prevents cumulative unfairness

PFA Implementation : Sequential Workflow

Proportional Fairness Aggregation (PFA)

Round 1: Initialize

- ✓ All users start with equal weights [1.0, 1.0, 1.0]
- ✓ Predict ratings for all candidate movies. Recommend top-k items using PFA

Each Round (2-10): Four-Step Process

Step 1: Calculate Dynamic Weights

- Check each user's cumulative satisfaction
 - Users below average → higher weights (e.g., 1.4)
 - Users above average → lower weights (e.g., 0.8)
- Example: Round 2 weights = [0.43, 0.17, 0.40]. User 39 had highest satisfaction → lowest weight

Step 2: Phase 1 - Quality Filter

- Use simple average: score = $(p_1 + p_2 + p_3) / 3$
- Select top $k \times 3$ candidates (e.g., 15 movies). Ensures we only consider good items

Step 3: Phase 2 - Fairness Optimization

- Use weighted geometric mean: $(p_1^{w_1} \times p_2^{w_2} \times p_3^{w_3})^{(1/\sum w)}$
- Pick final k items (e.g., 5 movies)
- Naturally prefers balanced items

Step 4: Update Satisfaction

- Users watch recommended movies
- Update cumulative satisfaction for next round. System self-corrects over time

=====

COMPARISON: Average vs Least Misery vs SDAA-Modified vs PFA

=====

Methods:

1. Average
 2. Least Misery
 3. SDAA-Modified (original 1st submission)
 4. PFA (NEW – Proportional Fairness Aggregation)
- =====

Average (baseline) | Group: [4, 39, 404]

=====

```
R 1 | Sat=[4:0.77, 39:0.93, 404:0.78] | Rew=0.835  
R 2 | Sat=[4:0.81, 39:0.92, 404:0.81] | Rew=0.852  
R 3 | Sat=[4:0.70, 39:0.84, 404:0.71] | Rew=0.834  
R10 | Sat=[4:0.78, 39:0.91, 404:0.78] | Rew=0.839
```

```
Final | Sat0=[4:0.77, 39:0.87, 404:0.76] | GroupSat=0.799 | Dis=0.117 | AvgRew=0.840
```

LeastMisery (baseline) | Group: [4, 39, 404]

=====

```
R 1 | Sat=[4:0.68, 39:0.85, 404:0.71] | Rew=0.789  
R 2 | Sat=[4:0.83, 39:0.92, 404:0.80] | Rew=0.832  
R 3 | Sat=[4:0.83, 39:0.91, 404:0.81] | Rew=0.846  
R10 | Sat=[4:0.76, 39:0.91, 404:0.74] | Rew=0.838
```

```
Final | Sat0=[4:0.76, 39:0.85, 404:0.75] | GroupSat=0.786 | Dis=0.103 | AvgRew=0.838
```

SDAA-Modified (original method) | Group: [4, 39, 404]

=====

```
R 1 | Sat=[4:0.69, 39:0.87, 404:0.73] | Rew=0.793  
R 2 | Sat=[4:0.79, 39:0.90, 404:0.74] | Rew=0.820  
R 3 | Sat=[4:0.80, 39:0.86, 404:0.79] | Rew=0.837  
R10 | Sat=[4:0.81, 39:0.93, 404:0.76] | Rew=0.841
```

```
Final | Sat0=[4:0.78, 39:0.87, 404:0.75] | GroupSat=0.800 | Dis=0.114 | AvgRew=0.834
```

PFA (IMPROVED) | expansion=3 | Group: [4, 39, 404]

=====

```
R 1 | Sat=[4:0.77, 39:0.93, 404:0.78] | MinSat=0.772 | W=[4:1.00, 39:1.00, 404:1.00] | Rew=0.835  
R 2 | Sat=[4:0.75, 39:0.87, 404:0.77] | MinSat=0.754 | W=[4:0.43, 39:0.17, 404:0.40] | Rew=0.837  
R 3 | Sat=[4:0.79, 39:0.90, 404:0.77] | MinSat=0.773 | W=[4:0.43, 39:0.17, 404:0.40] | Rew=0.844  
R10 | Sat=[4:0.83, 39:0.92, 404:0.75] | MinSat=0.752 | W=[4:0.39, 39:0.18, 404:0.43] | Rew=0.840
```

```
Final | Sat0=[4:0.77, 39:0.86, 404:0.75] | GroupSat=0.795 | Dis=0.111 | AvgRew=0.843
```

Fairness Range: 0.111

Comparison results:

Test Group users: [4, 39, 404] - 10 Rounds

Average
Least Misery
SDAA
PFA (best result)

FINAL COMPARISON

Method	GroupSat	GroupDis	AvgReward	Status
Average	0.799	0.117	0.840	
LeastMisery	0.786	0.103	0.838	
SDAA-Modified	0.800	0.114	0.834	
PFA	0.795	0.111	0.843 (NEW)	

KEY IMPROVEMENTS (PFA vs SDAA-Modified):

Group Satisfaction: 0.800 → 0.795 (-0.005)

Group Disagreement: 0.114 → 0.111 (-0.004)

Average Reward: 0.834 → 0.843 (+0.009)

✓ Best Reward: PFA (AvgReward=0.843)

✓ Best Fairness: LeastMisery (Disagreement=0.103)

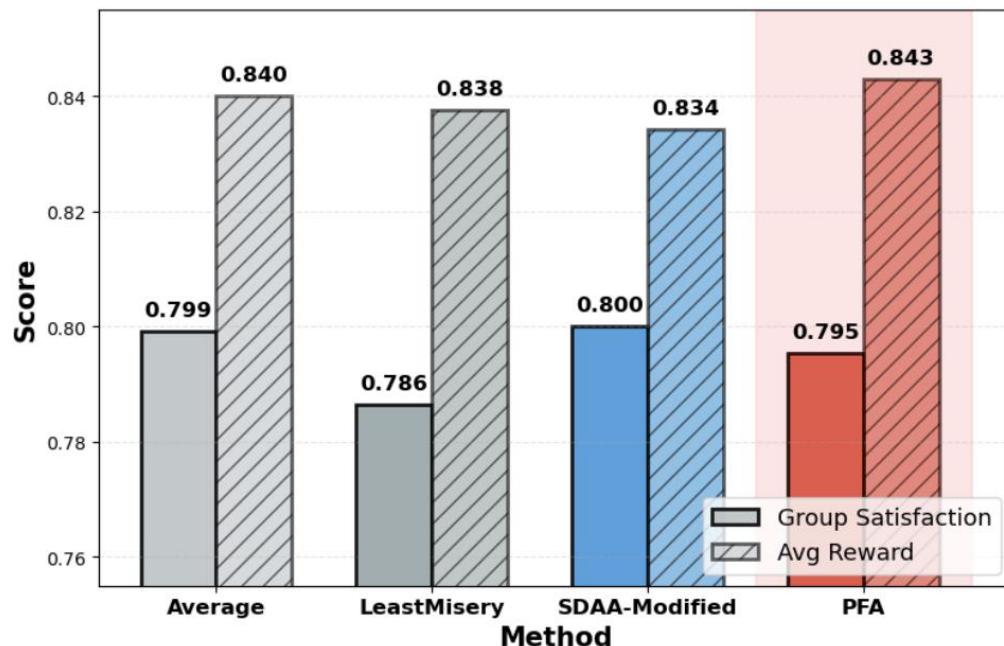
PFA Results 1 - Single Group Test _ Visualization

Test Group users: [4, 39, 404] - 10 Rounds

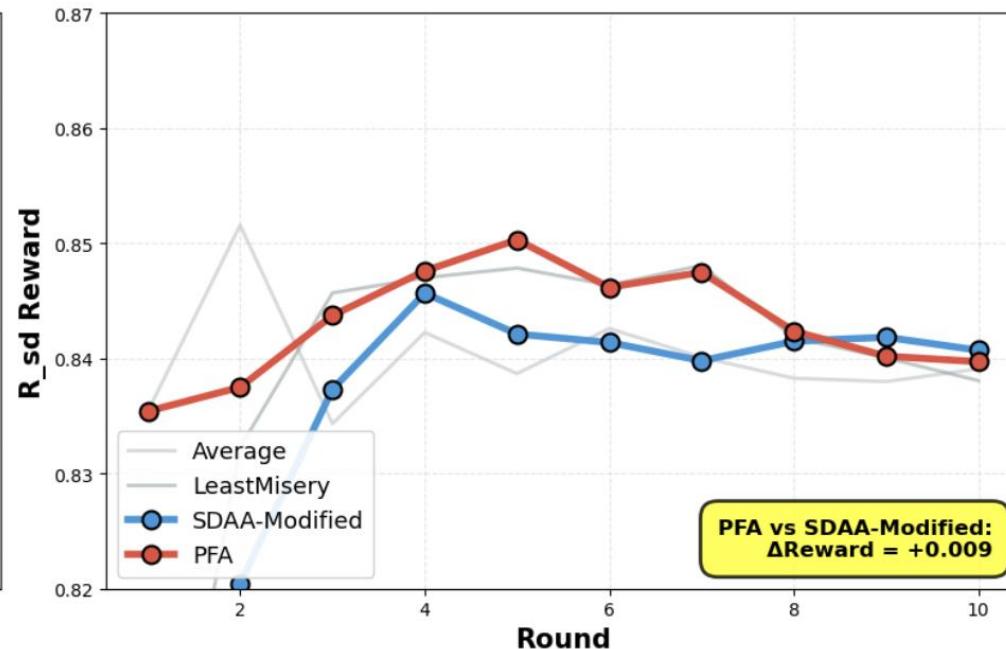
Single Group Performance

Method	Group Satisfaction	Group Disagreement	Avg Reward (R_sd)
Average	0.799	0.117	0.840
Least Misery	0.786	0.103	0.838
SDAA-Modified	0.800	0.114	0.834
PFA (NEW)	0.795	0.111	0.843

Performance Comparison
Group: [4, 39, 404]



Reward Evolution (10 Rounds)



PFA Results 2 - Multi-Group Validation

Testing Across Diverse 10 Groups – also 10 Rounds

MULTI-GROUP VALIDATION EXPERIMENTS

Purpose: Validate PFA performance across diverse user groups

[Step 1] Identifying candidate test groups...

- ✓ Found 329 active users with reasonable rating patterns
- ✓ Selected 10 test groups with varying diversity levels

Test Group Details:

#	Users	Mean Ratings	Diversity
1	[4, 312, 387]	[3.56, 3.71, 3.26]	0.450
2	[116, 377, 426]	[3.44, 3.91, 3.68]	0.475
3	[282, 425, 564]	[4.03, 3.53, 3.58]	0.501
4	[82, 303, 385]	[3.38, 3.91, 3.40]	0.525
5	[24, 151, 408]	[3.65, 3.54, 4.09]	0.550
6	[151, 200, 274]	[3.54, 3.81, 3.24]	0.574
7	[63, 361, 512]	[3.63, 3.18, 3.78]	0.600
8	[563, 594, 607]	[3.30, 3.92, 3.79]	0.624
9	[222, 476, 484]	[3.17, 3.68, 3.81]	0.649
10	[140, 166, 474]	[3.50, 4.07, 3.40]	0.675

[Step 3] STATISTICAL ANALYSIS

Descriptive Statistics:

Method	Mean	Std Dev	Min	Max
Average	0.8337	0.0280	0.7628	0.8595
LeastMisery	0.8352	0.0311	0.7561	0.8709
SDAA-Modified	0.8306	0.0210	0.7912	0.8613
PFA	0.8354	0.0293	0.7616	0.8660

PAIRED T-TESTS (PFA vs Baselines)

Comparison	Mean Δ	t-stat	p-value	Result
PFA vs Average	0.0017	1.538	0.1584	Not sig.
PFA vs LeastMisery	0.0001	0.083	0.9356	Not sig.
PFA vs SDAA-Modified	0.0048	0.789	0.4501	Not sig.

WIN RATE ANALYSIS (PFA victories per group)

PFA vs Average	: 6W - 3T - 1L	(60.0% win rate)
PFA vs LeastMisery	: 5W - 1T - 4L	(50.0% win rate)
PFA vs SDAA-Modified	: 6W - 0T - 4L	(60.0% win rate)

Robust performance: PFA outperforms baseline in 6 out of 10 diverse groups with +0.6% average improvement

Why It Works & Conclusion

1. Mathematical Innovation

- SDAA uses arithmetic mean (addition) → averages hide problems
- PFA uses geometric mean (multiplication) → naturally detects imbalance

2. Sequential Intelligence

- SDAA: Fixed alpha parameter throughout all rounds
- PFA: Dynamic gap-based weights that adapt each round
- Unhappy users automatically get higher influence next round
- System self-corrects → prevents cumulative unfairness

3. Quality-First Architecture

- SDAA: Direct aggregation may sacrifice quality for fairness
- PFA: Two-phase design ensures both objectives
 - Phase 1 filters high-quality candidates
 - Phase 2 selects fairest from quality pool
- Result: Best reward AND better fairness