

PERSONALIZED REWARD LEARNING WITH INTERACTION-GROUNDED LEARNING (IGL)

Jessica Maghakian¹, Paul Mineiro², Kishan Panaganti³,
Mark Rucker⁴, Akanksha Saran², Cheng Tan²

¹Stony Brook University, ²Microsoft Research NYC, ³Texas A&M University, ⁴University of Virginia

ICLR 2023
Kigali, Rwanda
May 1 – May 5, 2023



ICLR

REWARD ENGINEERING FOR RECOMMENDER SYSTEMS IS COMPLICATED

Goal: show users content that they like and enjoy

REWARD ENGINEERING FOR RECOMMENDER SYSTEMS IS COMPLICATED

Goal: show users content that they like and enjoy

Challenge: explicit user feedback is rare in recommender systems

REWARD ENGINEERING FOR RECOMMENDER SYSTEMS IS COMPLICATED

Goal: show users content that they like and enjoy

Challenge: explicit user feedback is rare in recommender systems

SOTA: find a “good” weighted combination of implicit feedback

- 2016: $r = 1 \text{ 👍} + 5 \text{ ❤️} + 5 \text{ 😄} + 5 \text{ 😲} + 5 \text{ 😞} + 5 \text{ 😡}$

REWARD ENGINEERING FOR RECOMMENDER SYSTEMS IS COMPLICATED

Goal: show users content that they like and enjoy

Challenge: explicit user feedback is rare in recommender systems

SOTA: find a “good” weighted combination of implicit feedback

• 2016: $r = 1 \text{ 👍} + 5 \text{ ❤️} + 5 \text{ 😄} + 5 \text{ 😲} + 5 \text{ 😞} + 5 \text{ 😡}$

• 2023:

```
private def getLinearRankingParams: ThriftRankingParams = {
  ThriftRankingParams(
    `type` = Some(ThriftScoringFunctionType.Linear),
    minScore = -1.0e100,
    retweetCountParams = Some(ThriftLinearFeatureRankingParams(weight = 20.0)),
    replyCountParams = Some(ThriftLinearFeatureRankingParams(weight = 1.0)),
    reputationParams = Some(ThriftLinearFeatureRankingParams(weight = 0.2)),
    luceneScoreParams = Some(ThriftLinearFeatureRankingParams(weight = 2.0)),
    textScoreParams = Some(ThriftLinearFeatureRankingParams(weight = 0.18)),
    urlParams = Some(ThriftLinearFeatureRankingParams(weight = 2.0)),
    isReplyParams = Some(ThriftLinearFeatureRankingParams(weight = 1.0)),
    favCountParams = Some(ThriftLinearFeatureRankingParams(weight = 30.0)),
    langEngLishUIBoost = 0.5,
    langEngLishTweetBoost = 0.2,
    langDefaultBoost = 0.02,
    unknownLanguageBoost = 0.05,
    offensiveBoost = 0.1,
    inTrustedCircleBoost = 3.0,
    multipleHashtagsOrTrendsBoost = 0.6,
    inDirectFollowBoost = 4.0,
    tweetHasTrendBoost = 1.1,
    selfTweetBoost = 2.0,
    tweetHasImageUrLBoost = 2.0,
    tweetHasVideoUrLBoost = 2.0,
    useUserLanguageInfo = true,
    ageDecayParams = Some(ThriftAgeDecayRankingParams(slope = 0.005, base = 1.0))
  )
}
```

REWARD ENGINEERING FOR RECOMMENDER SYSTEMS IS COMPLICATED

Goal: show users content that they like and enjoy

Challenge: explicit user feedback is rare in recommender systems

SOTA: find a “good” weighted combination of implicit feedback

- 2016: $r = 1 \text{ 👍} + 5 \text{ ❤️} + 5 \text{ 😄} + 5 \text{ 😲} + 5 \text{ 😞} + 5 \text{ 😡}$
- 2023: $r = 20 \text{ 💬} + 1 \text{ ↻} + 0.5 \text{ ♥️}$



The screenshot shows a Twitter thread. The top tweet is from Szymon Kopeć (@szymonkopez), replying to @petergyang. The tweet text is: "I'm surprised these weights are hard-coded vs decided by the ML model". It was posted on Mar 31, 2023, at 9:28 PM, and has 88.5K views. Below the tweet are 24 retweets, 4 quotes, 542 likes, and 5 bookmarks. The bottom tweet is from Elon Musk (@elonmusk), replying to @szymonkopez and @petergyang. The tweet text is: "Should be & will be". It was posted on Apr 1. Below the tweet are 103 replies, 93 retweets, 1,320 likes, and 90.7K views.

...
Szymon Kopeć 
@szymonkopez

Replying to @petergyang

I'm surprised these weights are hard-coded vs decided by the ML model

9:28 PM · Mar 31, 2023 · 88.5K Views

24 Retweets 4 Quotes 542 Likes 5 Bookmarks

💬 ↻ ♥️ 📌 ↗️

Elon Musk 
@elonmusk · Apr 1

Replying to @szymonkopez and @petergyang

Should be & will be

💬 103 ↻ 93 ♥️ 1,320 📊 90.7K ↗️

REWARD ENGINEERING FOR RECOMMENDER SYSTEMS IS COMPLICATED

Goal: show users content that they like and enjoy

Challenge: explicit user feedback is rare in recommender systems

SOTA: find a “good” weighted combination of implicit feedback

- 2016: $r = 1 \text{ 👍} + 5 \text{ ❤️} + 5 \text{ 😄} + 5 \text{ 😲} + 5 \text{ 😞} + 5 \text{ 😡}$
- 2023: $r = 20 \text{ 💬} + 1 \text{ ↻} + 0.5 \text{ ♥️}$

The screenshot shows a Twitter thread. The top tweet is from Szymon Kopec (@szymonkopec), replying to @petergyang. The tweet text is: "I'm surprised these weights are hard-coded vs decided by". It has 24 retweets, 4 quotes, 542 likes, and 5 bookmarks. The bottom tweet is from Elon Musk (@elonmusk), replying to @szymonkopec and @petergyang. The tweet text is: "Should be & will be". It has 103 replies, 93 retweets, 1,320 likes, and 90.7K views. A green OpenAI logo is visible on the right side of the thread.

Goal: show users content that they like and enjoy

Challenge: explicit user feedback is rare in recommender systems

SOTA: find a “good” weighted combination of implicit feedback

- 2016: $r = 1 \text{ 👍} + 5 \text{ ❤️} + 5 \text{ 😄} + 5 \text{ 😲} + 5 \text{ 😞} + 5 \text{ 😡}$
- 2023: $r = 20 \text{ 💬} + 1 \text{ ↻} + 0.5 \text{ ♥️}$

Using fixed weighting of implicit feedback is not ideal...

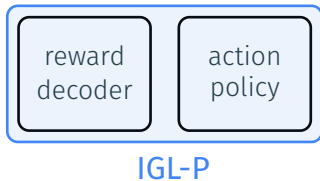
- exhaustive counterfactual search scales poorly
- ever-changing UI and non-stationary users
- unfairness due to one-size-fits-all rewards

PERSONALIZED REWARD LEARNING WITH IGL-P

Idea: *learn* personalized reward functions through user interactions

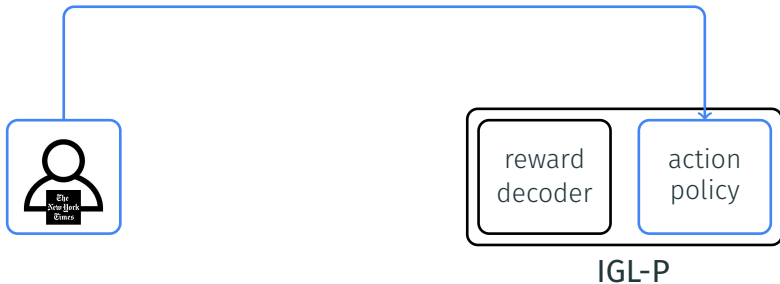
PERSONALIZED REWARD LEARNING WITH IGL-P

Idea: *learn* personalized reward functions through user interactions



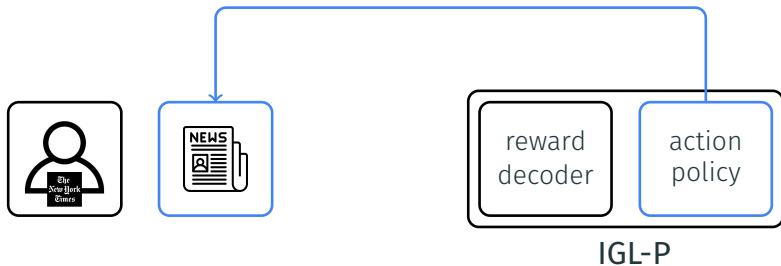
PERSONALIZED REWARD LEARNING WITH IGL-P

Idea: *learn* personalized reward functions through user interactions



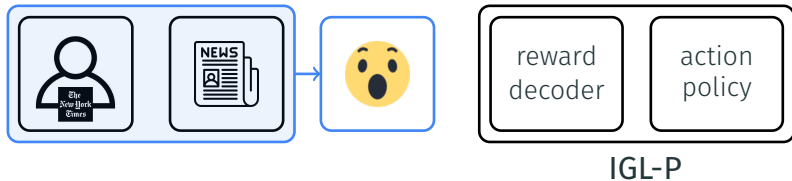
PERSONALIZED REWARD LEARNING WITH IGL-P

Idea: learn personalized reward functions through user interactions



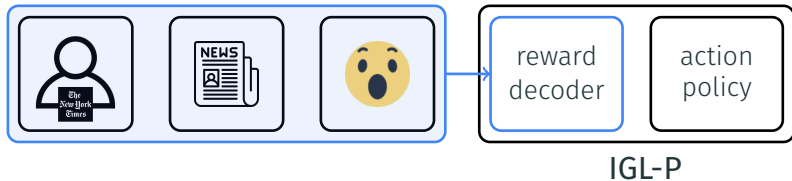
PERSONALIZED REWARD LEARNING WITH IGL-P

Idea: *learn* personalized reward functions through user interactions



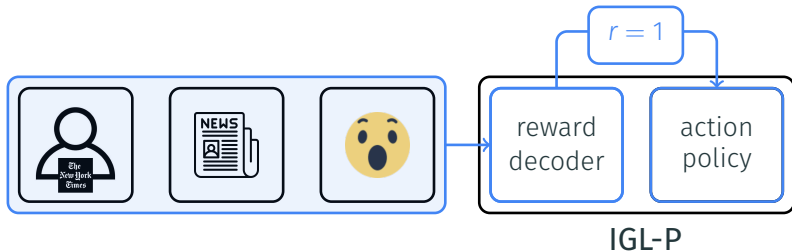
PERSONALIZED REWARD LEARNING WITH IGL-P

Idea: learn personalized reward functions through user interactions



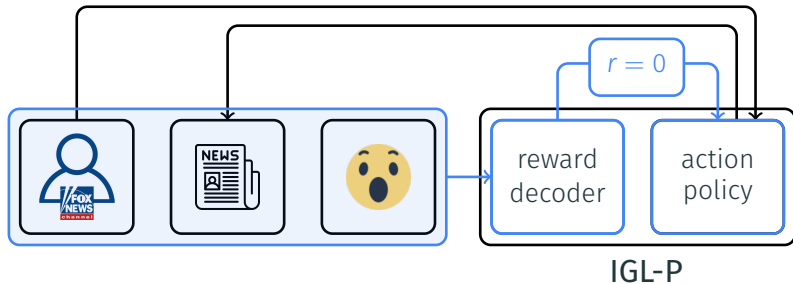
PERSONALIZED REWARD LEARNING WITH IGL-P

Idea: learn personalized reward functions through user interactions



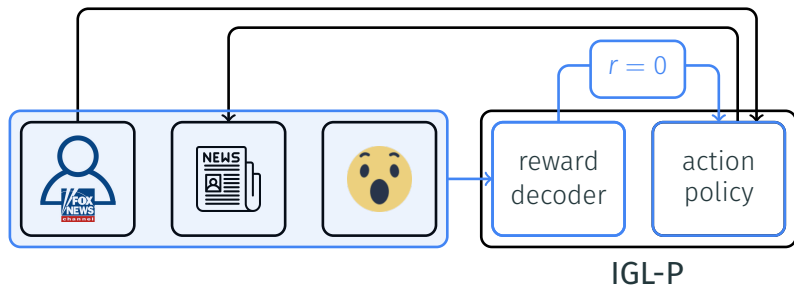
PERSONALIZED REWARD LEARNING WITH IGL-P

Idea: learn personalized reward functions through user interactions



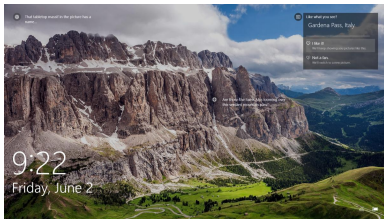
PERSONALIZED REWARD LEARNING WITH IGL-P

Idea: learn personalized reward functions through user interactions

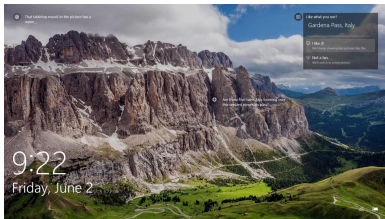


IGL-P only requires two simple conditions to succeed:
(1) rewards are rare and (2) users communicate consistently

Exp 1: Image recommendation for Windows users



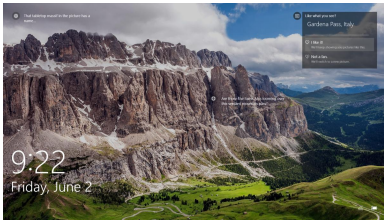
Exp 1: Image recommendation for Windows users



Compared to production policy, IGL-P received similar positive feedback and improved negative feedback, despite training on significantly less data.

IGL-P SUCCEEDS ON REAL WORLD DATA

Exp 1: Image recommendation for Windows users



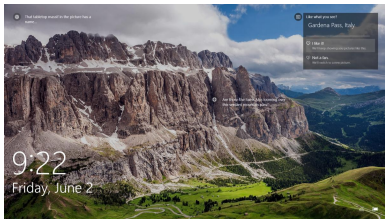
Exp 2: News recommendation for Facebook users



Compared to production policy, IGL-P received similar positive feedback and improved negative feedback, despite training on significantly less data.

IGL-P SUCCEEDS ON REAL WORLD DATA

Exp 1: Image recommendation for Windows users



Compared to production policy, IGL-P received similar positive feedback and improved negative feedback, despite training on significantly less data.

Exp 2: News recommendation for Facebook users



Policies trained with rewards used by Facebook circa 2017 had unfair performance. IGL-P performed consistently well across different user types.

SUMMARY OF OUR RESULTS



IGL-P can match
state-of-the-art
performance
at a fraction
of the cost

SUMMARY OF OUR RESULTS



IGL-P can match
state-of-the-art
performance
at a fraction
of the cost



IGL-P can easily
adapt and evolve
with changing
systems
and users

SUMMARY OF OUR RESULTS



IGL-P can match state-of-the-art performance at a fraction of the cost



IGL-P can easily adapt and evolve with changing systems and users



IGL-P uses personalized rewards to improve fairness for diverse users

SUMMARY OF OUR RESULTS



IGL-P can match state-of-the-art performance at a fraction of the cost



IGL-P can easily adapt and evolve with changing systems and users



IGL-P uses personalized rewards to improve fairness for diverse users

Although we introduced personalized reward learning for recommender systems, IGL-P can benefit any application that suffers from a one-size-fits-all approach!