Personalized Reward Learning with Interaction-Grounded Learning (IGL)

Jessica Maghakian[1], Paul Mineiro[2], Kishan Panaganti[3], Mark Rucker[4], Akanksha Saran[2], Cheng Tan[2]

[1]Stony Brook University, [2]Microsoft Research NYC, [3]Texas A&M University, [4]University 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: $r = 1$ 👍 + 5 ❤️ + 5 😆 + 5 😮 + 5 😢 + 5 😡

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: $r = 1$ 👍 + 5 ❤️ + 5 😆 + 5 😮 + 5 😢 + 5 😡
- 2023:

```scala
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: $r = 1$ 👍 + 5 ❤️ + 5 😆 + 5 😮 + 5 😢 + 5 😡
- 2023: $r = 20$ 💬 + 1 🔁 + 0.5 ♡

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: $r = 1$ 👍 + 5 ❤️ + 5 😆 + 5 😮 + 5 😢 + 5 😡
- 2023: $r = 20$ 💬 + 1 🔁 + 0.5 ♡

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: $r = 1$ 👍 + 5 ❤️ + 5 😆 + 5 😮 + 5 😢 + 5 😡
- 2023: $r = 20$ 💬 + 1 🔁 + 0.5 ♡

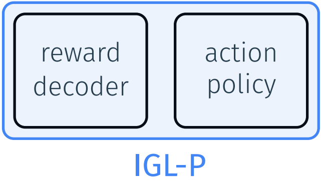
Using fixed weighting of implicit feedback is not ideal...

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

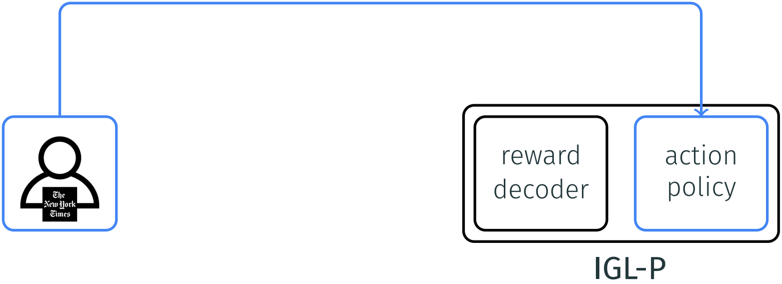
Personalized reward learning with IGL-P

Idea: *learn* personalized reward functions through user interactions

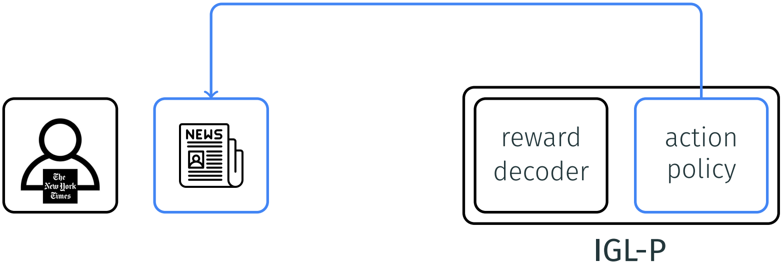
Idea: *learn* personalized reward functions through user interactions



Idea: *learn* personalized reward functions through user interactions

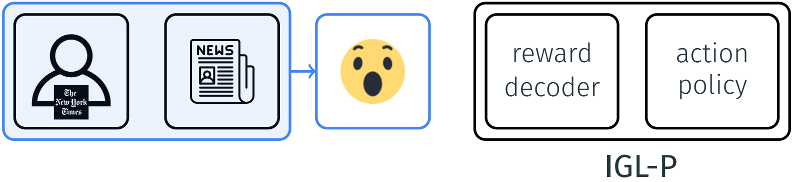


Idea: *learn* personalized reward functions through user interactions



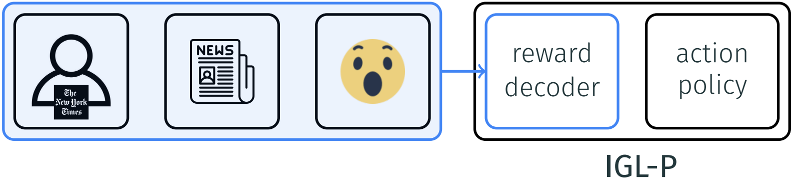
Personalized reward learning with IGL-P

Idea: *learn* personalized reward functions through user interactions



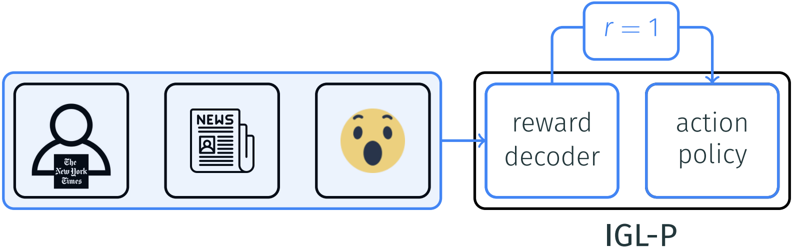
Personalized reward learning with IGL-P

Idea: *learn* personalized reward functions through user interactions



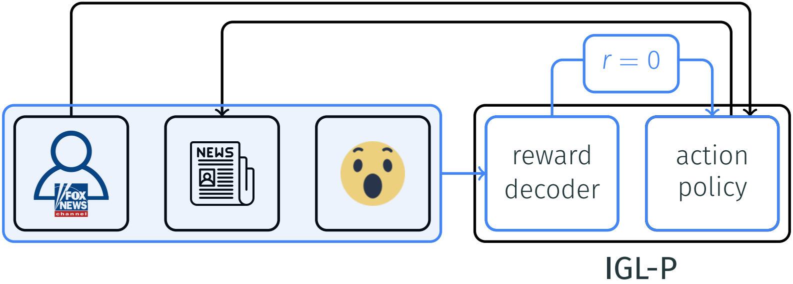
Personalized reward learning with IGL-P

Idea: *learn* personalized reward functions through user interactions



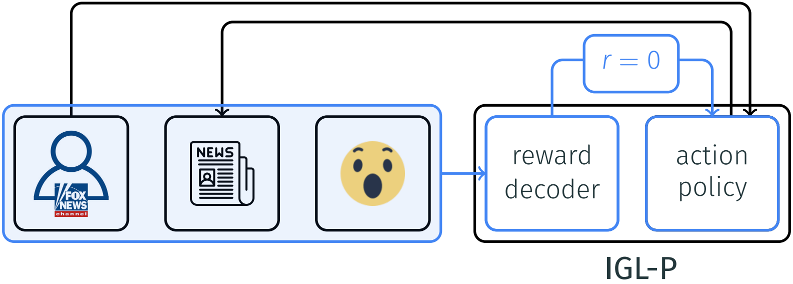
Personalized reward learning with IGL-P

Idea: *learn* personalized reward functions through user interactions



Personalized reward learning with IGL-P

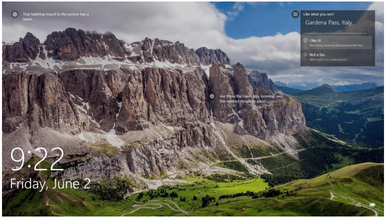
Idea: *learn* personalized reward functions through user interactions



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

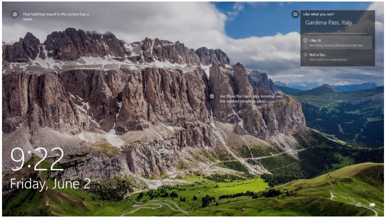
IGL-P SUCCEEDS ON REAL WORLD DATA

Exp 1: Image recommendation for Windows users



IGL-P SUCCEEDS ON REAL WORLD DATA

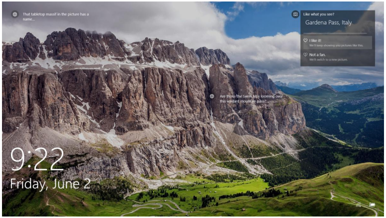
Exp 1: Image recommendation for Windows users



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

IGL-P SUCCEEDS ON REAL WORLD DATA

Exp 1: Image recommendation for Windows users

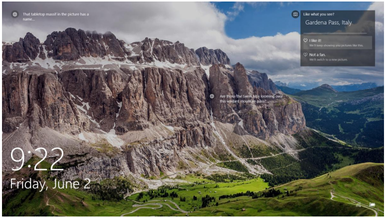


Exp 2: News recommendation for Facebook users

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



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.

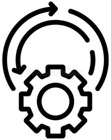
Policies trained with rewards used by Facebook circa 2017 had unfair performance. IGL-P performed consistently well across different user types.

Summary of our results

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

Summary of our results

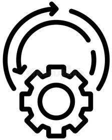
IGL-P can match state-of-the-art performance at a fraction of the cost



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

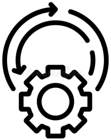


IGL-P can easily adapt and evolve with changing systems and users

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!