Towards a Fair Marketplace

R. Mehrotra[1], J. McInerney[1], H. Bouchard[1], M. Lalmas[1], F. Diaz[2]
[1]Spotify Research, [2]Microsoft Research

As narrated by: Alan Gee
and Dany Haddad

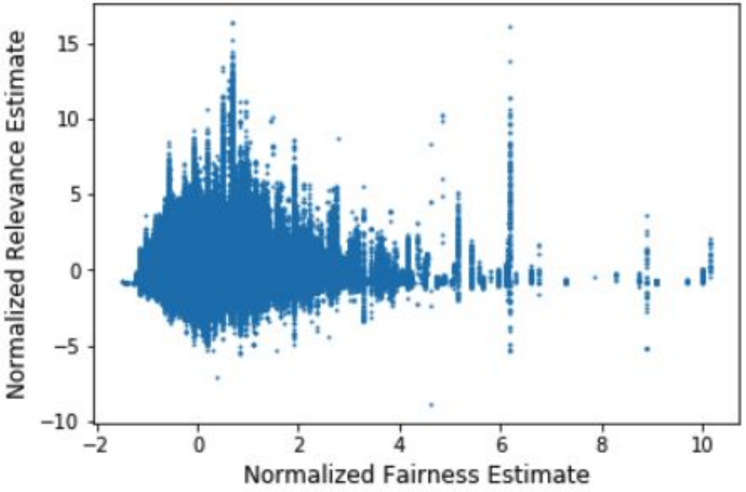
Paper Roadmap

- Recommendation systems as 2-sided marketplace (suppliers, consumers)
- Fairness towards the suppliers to avoid *superstar economics*
    - *S.E. : boosts popular choices, impedes new or less popular choices*
- Propose a *group fairness measure* ψ *, and other metrics*
    - Take advantage of some users being more open accepting of lower relevance results
- Evaluate unbiased user satisfaction using counterfactual estimation instead of A/B testing

Problem Setup

- Music recommendation and streaming service
- Goal is to maximize user satisfaction
    - Maximize the number of songs a user listens to

Key question: How can we be fair in recommending less popular suppliers to consumers but maintain good relevance?



Learning Relevance (Music) Recommendation

- Train skip-gram vector for each song and user based on historical listening logs
- Relevance of a track (set of tracks) to a user is given by cosine-similarity (average cosine-similarity)

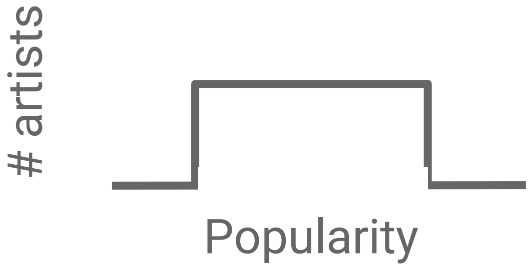
$$\sum_{u \in \mathcal{U}} \sum_{t \in \mathcal{P}_u} \log p(t|u)$$

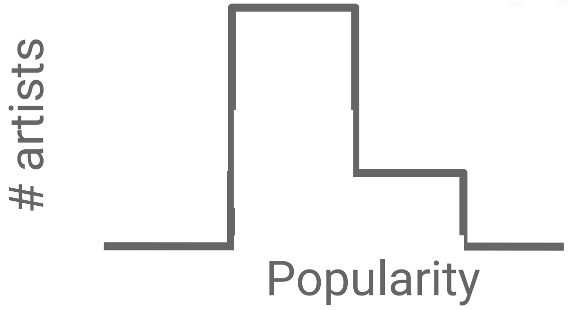
$$p(t|u) = \frac{u^T t}{\sum_{t' \in \mathcal{T}} u^T t'}$$

Fairness Measure

- Artists are binned into groups based on popularity
- Measure resembles a set-cover penalty, encouraging songs from artists from each popularity group to be represented

$$\psi(s) = \sum_{i}^{K} \sqrt{|G_i|}$$





$\psi\,(s_1) = \text{sqrt}(2) + \text{sqrt}(2) = 2.8 \quad > \quad \psi\,(s_2) = \text{sqrt}(3) + \text{sqrt}(1) = 2.7$

Relevance vs Fairness trade-off

- Optimizing Relevance (personalization of recommendations for customers)
  - Learned embeddings from historical logs

Relevance function

Optimal set for user $$s_u^* = argmax_{s \in S} \ \phi(u, s)$$

- Optimizing Fairness (showing content across popularity spectrum for suppliers)

$$s_u^* = argmax_{s \in S_u} \ \psi(s)$$

Fairness function

Sets pertinent to user

Group Level: Combining Relevance and Fairness

- Weighted Mixture (ø: relevance, ψ: fairness)

$$s_u^* = argmax_{s \in S_u} \;\; ((1-\beta)\,\phi(u,s) + \beta\,\psi(s)) \qquad\qquad \beta \in [0,1]$$

- Probabilistic Mixture (random draw p)

$$s_u^* = \begin{cases} argmax_{s \in S_u} \;\; \psi(s) & \text{if } p < \beta \quad p \in [0,1] \\ argmax_{s \in S} \;\; \phi(u,s) & \text{otherwise} \end{cases}$$

Choose a random p and see where it compares to tolerance β

Group Level: Combining Relevance and Fairness

- Guaranteeing Minimum Relevance with Fairness

$$s_u^* = argmax_{s \in S_u} \; \psi(s)$$
$$s.t. \; \phi(s, u) \geq \beta$$

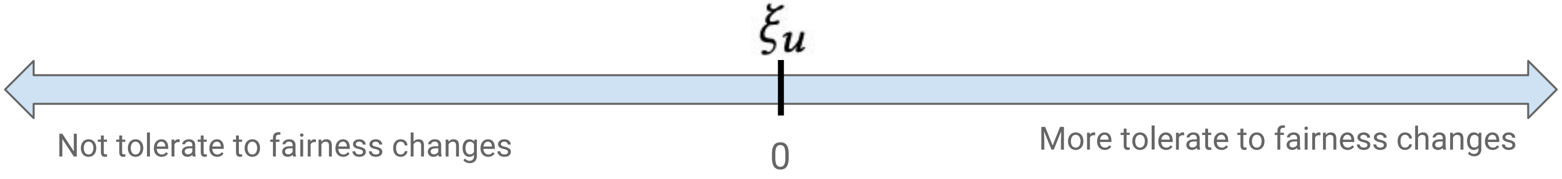
Constraint ensures the minimum relevance of recommendation

- Allows for recommendation sets with less relevance (due to fairness)

Personalized Recommendation

- Defined a user's affinity for fair content as*:

$$\xi_u = \frac{1}{|N_{.>\alpha_1}|} \underbrace{\Sigma_{\phi(u,s)>\alpha_1} \varsigma(u,s)}_{\text{Historic User Satisfaction with relevance content}} - \frac{1}{|N_{.>\alpha_2}|} \underbrace{\Sigma_{\psi(u,s)>\alpha_2} \varsigma(u,s)}_{\text{Historic User Satisfaction with fairness content}}$$



Score can then be used to understand a user's recommendation policy

*believed there is a typo (fixed here)

Adaptive Recommender: Managing Trade-off

- Adaptive - I (strict bifurcation of affinity scores)

$$s_u^* = \begin{cases} argmax_{s \in S_u} \;\; \psi(s) & \text{if } \xi_u >= 0 \\ argmax_{s \in S} \;\; \phi(u,s) & \text{if } \xi_u < 0 \end{cases}$$

Fairness optimized when scores are positive

Relevance optimized when scores are negative

- Adaptive - II

$$\hat{\xi_u} = \frac{\xi_u - \mu_{\xi_u}}{\sigma_{\xi_u}}$$

Normalized score across all users

$$s_u^* = argmax_{s \in S_u} \;\; ((1 - \hat{\xi_u})\, \phi(u,s) + \hat{\xi_u}\, \psi(s))$$

Reweight recommendation

Contextual Bandit Formulation

- A policy π gives a distribution over actions
- An action corresponds to recommending a specific set of songs
- The resulting reward is the satisfaction of a user (how many tracks they listened to)

Contextual Bandit

https://towardsdatascience.com/contextual-bandits-and-reinforcement-learning-6bdfeaece72a

Contextual Bandit Formulation

- Value of a policy π where:
  - Context, x, is drawn from the distribution over states, D
    - User and song features, time of day etc
  - Action, a, is chosen from distribution given by the policy, π
  - Reward, r, received is based on the context and action chosen

$$V(\pi) = E_{(x,r)\sim D}[r_{\pi(x)}]$$
$$= E_{x\sim D}E_{a\sim\pi(\cdot|x)}E_{r\sim D(\cdot|x,a)}[r_{\pi(x)}]$$

Offline evaluation of recommendations

- Allows us to avoid A/B testing!
- Collect randomized data
  - Assign playlists uniformly at random for each user
    - Unclear how the subset is chosen or how it affects the results
  - Record the user’s satisfaction (number of tracks listened to in playlist)

Offline evaluation of recommendations

$$\hat{V}_{offline} = \sum_{\forall(x,a,r_a,p_a)} \frac{r_a I\{\pi(x) = a\}}{p_a}$$

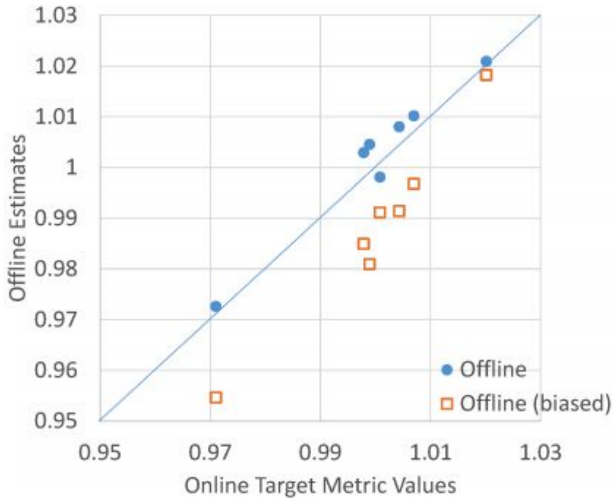


Figure 2: Online vs. offline target metric values. Each point corresponds to one of the 7 days in the data collection period.

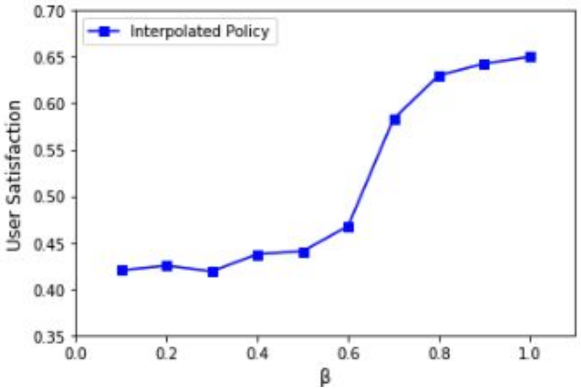
Lihong Li, Shunbao Chen, Jim Kleban, and Ankur Gupta. Counterfactual estimation and optimization of click metrics in search engines: A case study. In Proceedings WWW 2015.

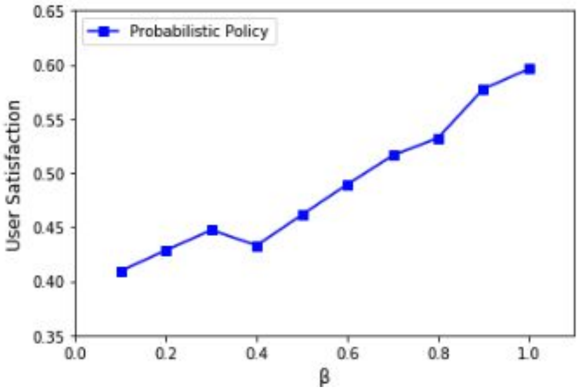
Tricks/Subtleties

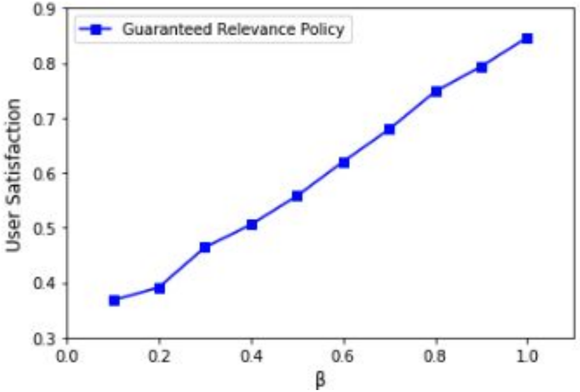
- 400k users' rewards for 5k playlists from 50k artists
- Store random seed at each point to avoid issues from pseudo-random number generation

$$\mathbf{Var}\left[\hat{V}_{\text{offline}}(\pi)\right] = \mathbf{E}_{(x,\vec{r})\sim D}\left[r^2_{\pi(x)}\left(\frac{1}{p_{\pi(x)}} - 1\right)\right]$$

Relevance vs Fairness Trade-off







- Favoring fairness over relevance decreases user satisfaction
  - Is this an acceptable trade-off?
- Guaranteed Relevance Policy seems to have the widest range of user satisfaction scores

Performance of Recommendation Policies

Recommendation Policy	β	% Loss in Fairness	% Loss in Relevance	% Gain in Satisfaction
Only Fairness	N/A	0	57.7	-35.3
Only Relevant	N/A	69.1	0	0
Interpolated	0.5	3.32	48.7	-32.2
	0.7	42.7	9.8	-10.2
	0.9	64.7	0.06	-1.1
probPolicy	0.5	16.3	44.8	-29.0
	0.7	30.8	32.7	-20.6
	0.9	48.2	17.6	-11.1
GuaranteedR	0.5	37.7	19.6	-14.2
	0.7	51.7	7.8	4.4
	0.9	63.9	0.59	22.1
Adaptive - I	N/A	17	20.2	9.0
Adaptive - II	N/A	15	21.2	12.1

Table 3: Comparing loss in fairness & relevance, with gains in satisfaction for different recommendation policies.

Summary

- Suggest approach to overcome *superstar economics*
- Give a heuristic to compute fairness towards a group of suppliers
- Provided a counterfactual estimation framework to estimate their impact on consumer satisfaction (offline)
- Show that some users are more accepting of “fair content” than others

Fairness Measure: Alternative

- A more natural notion might be to assign inverse-popularity scores instead
    - Otherwise, you might just always recommend songs from the most popular artist in each group