SIREN: a simulation framework for understanding the effects of recommender systems in online news environments — ACM FA(cc)T* Conference 2019

Dimitrios Bountouridis, Jaron Harambam, Mykola Makhortykh, Mónica Marrero, Nava Tintarev and Claudia Hauff

c.hauff@tudelft.nl / TU Delft
Fairnews project

Equal access to news is a precondition for a well-functioning democracy.

Data analytics and personalized recommendations make it possible to pre-sort news based on individual user profiles and social sorting. This project investigates to what extent algorithms can and may go into filtering information for the purpose of fairness. Unequal access to information can have major consequences for freedom of expression and non-discrimination.
Recommender systems

Help consumers deal with information overload via filtered and personalized suggestions.

Help content providers to increase user engagement, satisfaction and boost sales.
Recsys algorithms …

➔ Opaque
➔ Complex
➔ No transparency
➔ Lack of user control
➔ Matthew effect

Are they reducing diversity?
Our domain: news industry

➔ Recommends deliver information in line with people’s interests and preferences (→ homogeneity)

➔ Lowers people’s chances to encounter different content, opinions, viewpoints

➔ Media form an arena for public debate in which a diversity of voices should be heard
Our contribution

- **Simulation framework SIREN**: visualization and analysis of the effects of different recommenders systems on news consumption

- Simulation setup based on empirical data and the literature

---

“FH” model

products
consumers

attribute space

consumer knows
center products and
proximity products

one decision per user/period: buying one
product or not

recsys
News consumption context

➔ Mechanics of article publishing and consumption

➔ Intent of users and content providers

➔ Unique news-article characteristics: large volume, short-term relevancy, editorial cues (font size, positioning, title, etc.)

➔ News-reading behavior departs from the typical “show me something interesting” attitude
SIREN

→ Enables content providers to insert their own specifications (specific to their values, publishing habits, readers) and to test different recommender algorithms

→ Adjustable parameters: articles (items), readers (users) and recommendation algorithms

→ Evaluation metrics: Expected Popularity Complement (long-tail) and Expected Profile Distance (unexpectedness) diversity) *

One simulation iteration

- Users and articles are placed in a 2D attribute space

- One news cycle (article pool and recommendations are updated)
  - A user is aware of articles in their proximity, promoted articles and personalized recommended articles

- Some of those will be read

- User preferences are updated
Articles

Articles and readers are placed in a topic space

- BBC 2K news dataset: TF.IDF representations followed by t-SNE projection

- Article $t_j$'s prominence encoded by prominence attribute $z$ (changes over time) $\rightarrow$ long-tail distribution
Articles

→ Articles and readers are placed in a topic space
  → BBC 2K news dataset: TF.IDF representations followed by t-SNE projection

→ Article $t_j$’s prominence encoded by prominence attribute $z$ (changes over time) → long-tail distribution

→ Starting prominence: article promotion on its first ($x=1$) day of publication

→ 90% of interactions with an article happen within the first 5 days of the publication lifespan

$$z^x_j = (-px + 1)z^0_j,$$ where $p = 0.1$
Users

- Preferences (i.e. a user’s ideal article) are represented as points in the topic space

\[ P_{\text{read}}(u_i \text{ drifts towards } t_j) = e^{-\frac{\text{distance}_{ij}^2}{\theta_i^*}} \]

- User choice: prior to choosing, a user is aware of only a subset of all articles

\[ P(u_i \text{ aware of } t_j) = \lambda \theta' \log (1 - z_j)^{-1} + (1 - \lambda)e^{-\frac{\text{distance}_{ij}^2}{\theta}} \]

prominent vs. neighbouring article

awareness fading wrt. prominence/proximity
# Case study based on US news

<table>
<thead>
<tr>
<th>Variable</th>
<th>Adjustable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>U</td>
<td>$</td>
<td>✓</td>
</tr>
<tr>
<td>$\theta$</td>
<td>✗</td>
<td>0.07</td>
<td>Awareness decay with distance.</td>
</tr>
<tr>
<td>$\theta'$</td>
<td>✗</td>
<td>0.5</td>
<td>Awareness decay with article prominence.</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>✓</td>
<td>0.6</td>
<td>Awareness weight placed on prominent versus neighborhood articles.</td>
</tr>
<tr>
<td>$w$</td>
<td>✓</td>
<td>40</td>
<td>Maximum size of awareness pool.</td>
</tr>
<tr>
<td>$k$</td>
<td>✗</td>
<td>3</td>
<td>Choice model: the user’s sensitivity to distance on the map.</td>
</tr>
<tr>
<td>$\theta_i^*$</td>
<td>✗</td>
<td>$\sim N(0.1, 0.03)$</td>
<td>User-drift: user’s sensitivity to distance on the map.</td>
</tr>
<tr>
<td>$m$</td>
<td>✗</td>
<td>$0.05 \times \text{distance}_{ij}$</td>
<td>User-drift: distance covered between the article $t_j$ and user $u_i$.</td>
</tr>
<tr>
<td>$s$</td>
<td>✓</td>
<td>$\sim N(6, 2)$</td>
<td>Amount of articles read per iteration per user (session size).</td>
</tr>
</tbody>
</table>

| Recommender settings        | | |
|-----------------------------|---|---|---|
| $n$                         | ✓ | 5 | Number of recommended articles per user per iteration.                      |
| $\delta$                   | ✓ | 1 | Factor by which distance decreases for recommended articles (salience).     |
| $\beta$                    | ✗ | 0.9 | Ranking-based decay of recommender salience.                                |
| $d$                         | ✓ | 30 | Number of simulation iterations per recommender.                            |

| Article topic weights       | | |
|-----------------------------|---|---|---|
| $|T|$                         | ✓ | $d \times 100$ | Total number of articles (number of iterations $\times$ articles per day). |
| $z^0$                       | ✓ | $U(0, 5)$ | Percentage of articles added per day/iteration per topic.                   |
| $\rho$                      | ✗ | 0.1 | Prominence decrease factor per iteration.                                    |
MostPopular has the worst performance on long-tail diversity.

Random does not perform much better.
The best performing algorithms are **ItemKNN** (simple approach) and **WeightedBPRFM** (more sophisticated), with the latter converging to the same EPC diversity as the number of iterations/days increases.
UserKNN shows the importance of user-drift modeling (solid lines) vs. no user drift (dashed lines).
Discussion

⇒ Recommenders’ effects wrt. diversity are dependent on the evolution of readers’ preferences

⇒ Studies based on snapshots of real-life data can only provide a short-term understanding of the recommender effects

⇒ Accurate modeling of user-drift is vital for accurate simulations

⇒ Content providers need to understand their users’ impulse to change preferences prior to adopting any algorithm
Conclusions

➔ SIREN: an online news consumption simulation framework

➔ Designed to aid content providers to decide between different recommender algorithms

➔ Based on seminal work by Fleder & Hosanagar, adapted to the news context

➔ Limitations
  ➔ Missing factors in news consumption (social media, user-user interactions)
  ➔ Standard parametrizations of common recommender algorithms
  ➔ Insightful in practice (i.e. for content providers)?

➔ https://github.com/dbountouridis/siren