# Reducing Offline Evaluation Bias in Recommendation Systems

A. de Myttenaere, B. Golden, B. Legrand, F. Rossi

Viadeo & Paris 1 University (SAMM, CRI)

## Recommendation Systems at Viadeo

- The number one professional social network in France and emerging markets (60M members worldwide), Viadeo enables professionals to :
  - develop their network,
  - enhance their career prospects and uncover new business opportunities,
  - stay connected with their contacts.
- Recommendation algorithms are used in Viadeo for:
  - contacts suggestions $\longrightarrow$ increase members' network & engagement
  - job offers suggestions $\longrightarrow$ help members to find a job
  - skills suggestions $\longrightarrow$ enhance profile data
  - ads targeting $\longrightarrow$ optimize revenue

## Formalisation of offline evaluation

- The evaluation of an algorithm can be done using two kind of procedures: online (based on user interactions) or offline (based on historical data).
- Offline evaluation typically consists in a procedure in 4 steps:
  1- Select a user (randomly) and isolate (randomly) one of his items
  2- Simulate recommandations for this user as if he did not have the item selected at step 1
  3- Check the rank of the selected item in the vector of recommandations
  4- Repeat for different users
- Formally, denoting $\ell$ the loss function for an algorithm $a$, the expected score of algorithm $a$ for offline evaluation procedure is given by:

$$s(a) = \sum_{i,u} p(u,i)\ell(a(u_{-i}), i)$$

- Thus if $a$ suggests the same item for every user (constant algorithm), the score can be rewritten as:

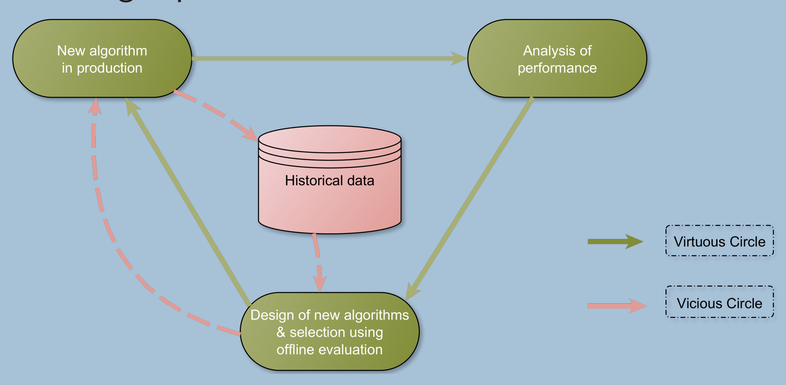
$$s(a) = \sum_{i} p(i)\ell(a(.), i)$$

So $s(a)$ directly depends on the probability selection of each item, $p(i)$.

# Bias in offline evaluation

## Origin of the bias

- A recommendation algorithm is nowadays regularly improved following a virtuous circle:
  - new algorithm put in production
  - analysis of live performance
  - improvement of the algorithm (design of new algorithms & selection using offline evaluation to quickly kill the "not promising" ones).
- However, offline evaluation depends on historical data collected, and those data have been influenced by recommendation algorithms previously put in production, creating a parallel vicious circle:




$\longrightarrow$ **Bias:** Offline evaluation tends to overestimate (resp. underestimate) the quality of an algorithm similar (resp. different) to the ones previously used.

## Illustration on experimental data

Impact of recommendation campaigns on items' probability of selection: it is more likely to select items which have been previously recommended, whereas the probability of selection decreases if an item has never been recommended.

So the offline evaluation scores of constant algorithms recommending those items evolve accordingly.




# Reducing the bias

## Suggested solution to reduce the bias

**Main idea:** decrease at step 1 of offline evaluation the probability of selecting items which have been recommended in the past. This can be done by weighting the probability selection of each item:

$$P(i|u,\omega) = \frac{\omega_i P(i|u)}{\sum_k \omega_k P(k|u)}$$

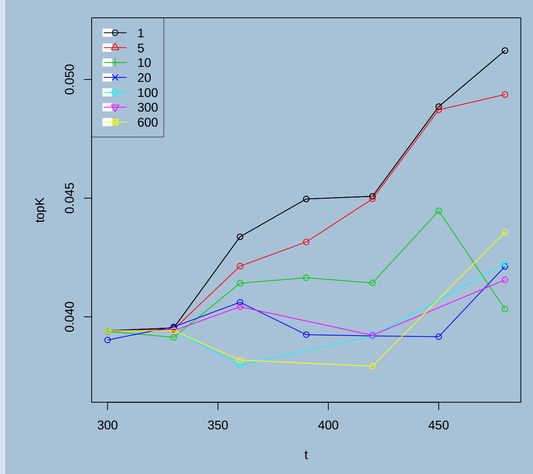
Assuming the situation at $t_0$ is bias free, we want that $P_{t_1}(i|\omega) = P_{t_0}(i)$ to ensure a fixed score within time for constant algorithms

We suggest to approximate this non-linear system by minimizing the Kullback-Leibler divergence between $P(i|\omega, t_1)$ and $P(i|t_0)$. We are thus looking for $\omega^\star$ that minimizes:

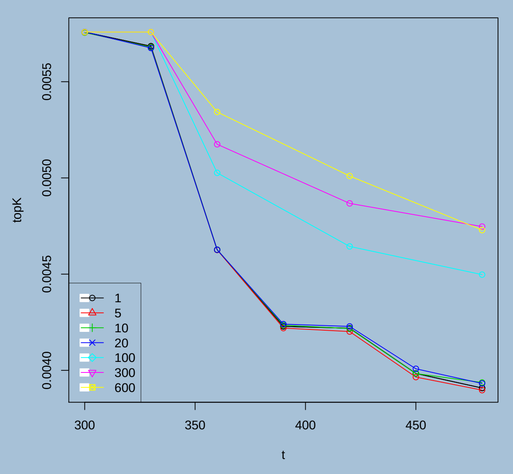
$$\mathcal{D}\big(P(i|t_0), P(i|\omega,t_1)\big) = \sum_i p(i|t_0) \log \frac{p(i|t_0)}{p(i|\omega,t_1)}$$

## Results on experimental data (using gradient descent)

Bias reduction for an algorithm similar to the one previously used:




Bias reduction for an "orthogonal" algorithm to the one previously used:




We see on this example that our method is effective to reduce the bias for constant algorithms, without needing to compute all coordinates of the gradient ($\longrightarrow$ strongly reduces the complexity of the optimization).

## Conclusion

- Offline evaluation can be biased because of previous recommendation algorithms used.
- For constant algorithms, we have shown that the bias can be reduced by weighting the probability selection of items.
- Experimental results illustrate that our approach works well to reduce the bias for constant algorithms.
- Future works include verifying that our method also works for more elaborate algorithms.

ademyttenaere@viadeoteam.com