Value-sensitive product (content) recommendation

Aditya Kiran

June 16, 2021

1 Abstract

2 Introduction

3 Formulation

For each article A_i in our database we know its 3 properties:

1. **v_i^1 is a vector of its graph-representation**:
   Let (V^1, E^1) be the coviewership graph. Here V^1 represents the set of graph-embeddings of all nodes (articles) v^1, and E^1 represents set of all edges connecting those nodes. d_{ij}^1's be the distance between v_i^1 and v_j^1.

2. **v_i^2 is a vector of its sentence embedding**:
   Let (V^2, E^2) be the similarity graph. As above, d_{ij}^2's be the distance between v_i^2 and v_j^2.

3. **c_i is the value associated with that article**

So, each article A_i is represented by this triad $\{v_i^1, v_i^2, c_i\}$. Although, this is a very general formulation. For the sake of simplicity, for now we will just assume that each the article is characterized by its sentence embedding and its value.

Problem Statement:

Now, the problem statement is:
Given that a user is reading an article A_i, what article must be recommended next so that we

1. maximize the chances of them reading more articles
2. maximize the ad-monetary value.

Mathematically, given A_i, we seek A_j such that:

$$j = \arg\max_J \phi(A_i, A_J),$$

where ϕ is the utility function.

How do we choose the utility function?

Each article A_J is attributed with its sentence-embedding and its value. So we need to design a utility function ϕ that maps its attributes to a real value:

$$\phi : \mathbb{R}^D \times \mathbb{R}^D \times \mathbb{R} \ \to \mathbb{R}$$

- ϕ must be directly proportional to the article's value

$$\phi(A_i, A_J) \propto c_J$$

- ϕ must be inversely proportional to the distance of the article A_J from the current article A_i

$$\phi(A_i, A_J) \propto \frac{1}{|d_i - d_J|} = \frac{1}{d_{iJ}}$$

Hence, we choose

$$\phi(A_p, A_q) := \frac{(c_q)^{r_1}}{(d_{pq})^{r_2}}.$$

Here r_1 and r_2 are the parameters that control the contribution of the value and the distance, on the utility term ϕ respectively.

Note: The special case of $\{r_1 = 0, r_2 = 1\}$ reduces the method back to the BAU.

Assumptions made:

1. The value of an article $\approx$ its revenue/impressions from the past
2. Ignore all articles with zero revenue or impressions
3. Each article is characterized by the 512-vector Spacy embedding of its post_title+meta_title+summary

Formulation for Infinite scroll: Quite often, we recommend more than just one article, either in the form of infinite-scroll or the '*Next Page*' option.

So a more general form of the problem is: Given an article A_i, we seek (N-1) articles $\{A_{a_1}, A_{a_2}, \cdots A_{a_{N-1}}\}$ such that:

$$\{a_1 \cdots a_{N-1}\} = \underset{\{b_1 \cdots b_{N-1}\}}{\arg\max} \sum_{k=1}^{N-1} \gamma^{k-1} \phi(A_i, A_{b_k}),$$

where $0 < \gamma < 1$ is called the discount factor.

Finally, after we know the 'optimal' recommendations, we evaluate the discounted value as

$$\mathcal{V}_{disc} := \sum_{k=1}^{N-1} \gamma^{k-1} c_{a_k}$$

We use the discounted-value instead of actual value to account for the fact that the likelihood of a user reading an article decreases with the scroll depth.

4 Results

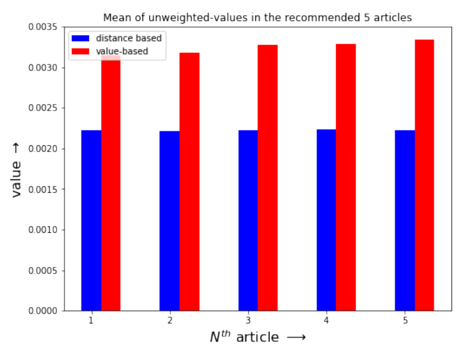
Simulation results for infinite scroll:

We used $r_1 = 0.8, r_2 = 15, \gamma = 0.09, N = 6$, and ran 30000 random simulations.

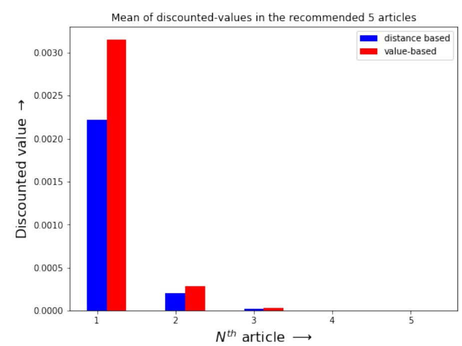
- We observe a lift of 41.96% in the discounted value

$$\left(\frac{\mathcal{V}_{disc}^{new} - \mathcal{V}_{disc}^{BAU}}{\mathcal{V}_{disc}^{BAU}}\right) \times 100 = 41.96\%$$

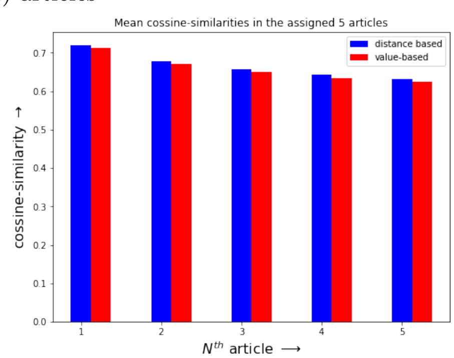
- Mean cosine-similarity between the recommended articles and the article being read:
  - Purely distance-based: 0.665
  - Value-distance based: 0.658



(a) Mean of unweighted-values in the recommended $(N-1)$ articles



(b) Mean of the discounted-values



(c) Mean cosine-similarities

Figure 1: Conglomerated results of the recommended articles after doing the random simulations.

```
Rand-expt #29991/30000
Current article being read:-
  ['Choosing the best mattress for combination sleepers']

Recommended using distance-based:-
1. [disc_val=0.00162, cossim=0.7820] The 11 Best Mattresses Made for Side Sleepers
2. [disc_val=0.00028, cossim=0.7618] 7 Best Mattresses for Back Sleepers
3. [disc_val=0.00005, cossim=0.7568] The 9 Best Hybrid Mattresses for Side Sleepers
4. [disc_val=0.00001, cossim=0.7447] What are some of the best mattress toppers for side sleepers?
5. [disc_val=0.00000, cossim=0.7383] Side sleepers: 8 of the best mattresses

Recommended using value-based:-
1. [disc_val=0.00565, cossim=0.7568] The 9 Best Hybrid Mattresses for Side Sleepers
2. [disc_val=0.00073, cossim=0.7447] What are some of the best mattress toppers for side sleepers?
3. [disc_val=0.00001, cossim=0.7820] The 11 Best Mattresses Made for Side Sleepers
4. [disc_val=0.00000, cossim=0.7618] 7 Best Mattresses for Back Sleepers
5. [disc_val=0.00000, cossim=0.7383] Side sleepers: 8 of the best mattresses

 Value: Old= 0.00195     Value-new= 0.00640      lift= 228.66 %
Mean cosine_similarity: Old= 0.757       new= 0.757
-------------------------------------------------------------------------------------
```

Figure 2: Example recommendations showing a lift in the value for the same set of articles.

5 Conclusion

As seen from Fig. (1c), by achieving an almost equal cosine similarity score as the BAU, we are achieving a higher value (Fig. (1b)) using the value-based method.

Hence we think that it is worth implementing this value-based recommendation method into a live A/B test with the BAU.