Budget-constrained Recommendation of a Set of Alternatives for Common Use

Silvan Hungersbühler & Haukur Jónsson & Grzegorsz Lisowski & Max Rapp

17/05/2017

Outline

We seek to find good ways for a group of agents to spend a limited budget on a set of alternatives which they all have to consume and over which they have different preferences.

- Motivation: Two Scenarios
- The Knapsack Problem for Multiple Agents
- Formal Framework
- Desirable Properties of Recommendation Sets
- Budgeted Voting Rules
- Results: Rules & Property Satisfaction
- Future Work

Motivation: A problem with the News Media

News Media are vulnerable to the following vicious cycle: People are biased. They are more likely to consumn media content that confirms their biases. Thus media have an incentive to feed their consumers' bias. If they follow it, this can entrench division and erode the commonly accepted factual base. If they don't, a part of the population may come to believe that their world view is not represented by the media and become increasingly distrusting of the news.

A Common Core

Can we devise a transparent, intersubjectively justified method to select a common core of issues to which everybody should pay attention?

Scenario I: An Essential Readings News Recommendation Algorithm

Online social networks often recommend news articles to their users based on an algorithm that infers preferences from user's past behavior and demographic properties. Such a News Feed may have little overlap for people belonging to different social or political spheres.
Imagine a news aggregator that asks users for an amount of time they want to read everyday, scrapes their social network accounts and based on this information creates a list of essential readings of the desired length.

Scenario II: A Newspaper Page

A newspaper with a diverse readership has recently lost readers on one side of the political spectrum who feel that the medium does not report on the issues they care about. Responding to this criticism they would like to create a page that reflects preference data they collect from their readers. How can they fill the page in a way that is a good compromise given their voters diverse preferences over issues?

The Knapsack Problem

At a higher level of abstraction, we seek to find good ways for a group of agents to spend a limited budget on a set of alternatives which they all have to consume and over which they have different preferences.
For the single agent case, this problem is the well-known Knapsack Problem: How can I find the combination of items that fits in my knapsack which is most valuable for my camping trip?
However, if the knapsack is packed for a group with different preferences, there is no simple way to maximize utility. Thus a voting rule is needed.

Formal Framework

We have a set of *news items* $A = \{a_1, ..., a_m\}$, a set of *recommended items* $W \subseteq A$, a set of *consumers* $N = \{n_1, ..., n_n\}$, a *profile of preferences* over the set of items $\mathcal{R} \in \mathcal{L}^n$ and a *budget* $B \in \mathbb{R}_{\geq 0}$.
We derive utilities from $\mathcal{R}$ employing a utility function $u_i : \mathcal{L}^n \times N \times A \to \mathbb{N}$ assigning each news item its Borda score for the preference order of reader i. This function is extended to $u(a) = \sum_{i=1}^{n} a$, $u_i(W) = \sum_{a \in W} u_i(a)$ and $u(W) = \sum_{a \in W} u(a)$ for $W \in \mathcal{P}(A)$. Likewise each news item has a specific *cost* $C : A \to \mathbb{N}$ that extends to sets by $C(W) = \sum_{a \in W} C(a)$.

Desirable Properties: Axioms

Here should be a table with all the axioms for which we have proofs

Reference

Desirable Properties: Utility Maximization

Let $\mathcal{W}_{\mathcal{B}}$ be the set of all elements of $\mathcal{P}(A)$ s.t. $C(W) \leq B$. Utility is maximized if:

$$W = \underset{W' \in \mathcal{W}_B}{\arg\max}(u(W'))$$

Desirable Properties: θ-Minority Consistency

A recommendation set is θ-minority consistent if

$$a \in W \text{ whenever } \frac{\mathcal{N}_{a \succ b}}{N} \geq \theta \text{ for all } b \in A \setminus \{b\}$$

Desirable Properties: δ-Equality

Define the Gini-coefficient of a recommendation set as follows:

$$G(W) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |u_i(W) - u_j(W)|}{2n * u(W)}$$

For $\delta \in [0, 1]$, a recommendation set is δ-egalitarian if $G(W) \leq \delta$.

Voting Rules: Knapsack

This rule takes the voters preference orders and maximizes utility given the budget:

$$\max_{W \in \mathcal{W}_{\mathcal{B}}} u(W)$$

Voting Rules: Commitee of θ-Winners

We define a news item's θ-score as follows:

$$\theta(a) = \min_{b \in A} \frac{\mathcal{N}_{a \succ b}}{N}$$

The commitee of θ-winners is then chosen as follows:

$$W_1 = \{\arg\max_{a \in A} \theta(a)\} \cap \mathcal{W}_B$$

$$W_n = W_{n-1} \cup \{ \arg\max_{a \in A \setminus W_{n-1}} \theta(a)\} \cap \mathcal{W}_{B-C(W_{n-1})}$$

$$W = W_{|A|}$$

Voting Rules: Further Rules

Further we tested budgetized versions of the k-Plurality, k-Borda and k-Copeland rules.

Methods: Proofs & Data-Generation

We proved satisfaction or failure of satisfaction of the axioms. In addition we simulated the behaviour of the voting rules with respect to the desirable properties and the failed axioms.
For this purpose, since unfortunately good real world data was hard to find, we generated datasets randomly to our needs. We generated one dataset of small profiles with 5000 readers and 10 news items and one dataset of large profiles with 5000 readers and 100 news items. In addition, for each dataset we used various procedures to generate fully random profiles as well as profiles of several degrees of "fragmentization" from 1 to 100 clusters of preference orders.

Methods: Statistical Analysis

Greg should write something here.

Results I: Axioms

Here a table with the rules and which axioms they satisfy

Results II: Utility Maximization

Results III: *theta*-Minority Consistency

Results IV: δ-Equality

Results V: An Overall Best Rule?

Future Work

- The cost and utility functions could be used to express dependencies (some items form series of articles, cost might vary depending on reader)
- Implement slack: in many application leftover budget also has a utility
- Find, collect or mine real data (webscraping...).
- Build a ready-to-use implementation!