PittGrub: A Frustration-Free System to Reduce Food Waste by Notifying Hungry College Students

Mark Silvis Anthony Sicilia, Alexandros Labrinidis
School of Computing and Information
University of Pittsburgh
{marksilvis, anthonysicilia}@pitt.edu, labrinid@cs.pitt.edu

#Food Waste, #Food Insecurity, #AI For Good, #Sustainability

https://pittgrub.com/kdd2018.pdf

Motivation

- Food waste is a humanitarian, environmental, and economic problem (see **Figure 2**)
- We provide local caterers and event hosts with an easy outlet for left over food:
  - Rather than contributing to waste, they can post via an application
  - We bring our hungry users to claim the leftovers (see **Figure 3**)
- But, mass notification is not a scalable system (see **Figure 4**)
- How can we constrain notifications while prioritizing specific user subsets?

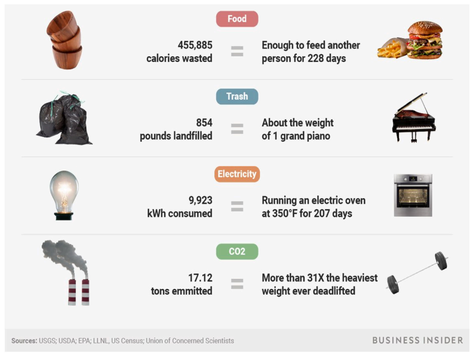

Figure 2: One year of average American's waste.

Figure 3: (Above) The general idea.

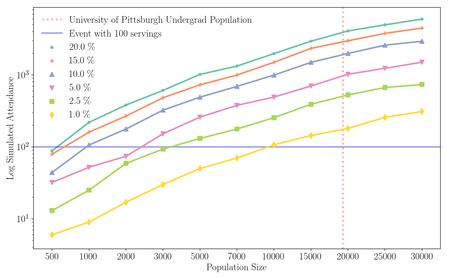

Figure 4: Simulation of user populations with varying average likelihood of attendance shows user **frustration** from over attendance is problematic if mass notification is performed.

Contributions

1) Develop **prototype** to connect users to free food, alleviating local food waste in tandem
2) Propose *smart* alternative to mass notification, selecting both ***whom*** and ***how many*** to notify.
3) Evaluate proposed method within a simulated environment.

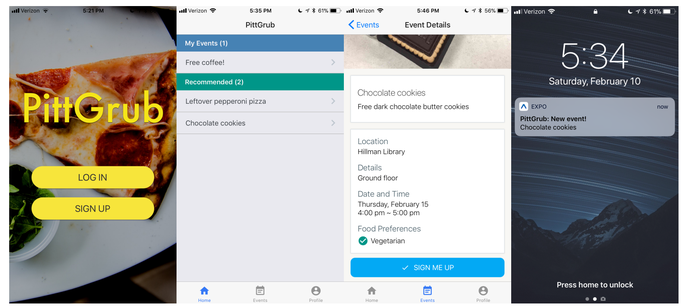

Figure 1: Screenshot's of PittGrub application to be released late August

Solution

Metrics

The ideal user selection algorithm would maximize:

- Average probability of user attendance per event
- Total notification fairness over a number of events
- Percentage of **Pantry (food-insecure)** users

To evaluate performance, we consider the following metrics (averaged to aggregate, if necessary):

- **Waste**: the # of remaining servings after an event
- **Average Probability**: the mean over an event of notified user-probabilities $\frac{1}{n_e}\sum_{i=1}^{n_e} p_{i,e}$
- **Total Fairness:** comparison of all user notification counts against max notification count over a subset of events $1 - \frac{1}{M*n}\sum_{i=1}^{n} M - x_i$
- **Percent Pantry**: the percentage of invited pantry users for an event $\frac{1}{n_e}\sum_{i=1}^{n_e} y_{i,e}$

Baselines

Algorithms designed for comparison against our proposed method:

- **Frequent-First**: maximizes *Average Probability*
- **Round-Robin**: maximizes *Total Fairness*
- **Pantry-First**: maximizes *Percent Pantry*

Proposed Method

- Frame user selection as a modified knapsack problem: $\max \ \mathbf{x}^T\mathbf{v}^T \ \text{ s.t. } \ \mathbf{x}^T\mathbf{p}^T \leq \omega S$
- $\boldsymbol{x}$ an indicator vector for user notification
- $\boldsymbol{p}$ a vector of estimated user probabilities of attendance
- $\boldsymbol{v}$ a vector of valuations for each user (see **FIPS**).
- $\boldsymbol{\omega}$ a booking factor to correct for biased probability estimates. It is learned via **booQ**.
- implementation is greedy for efficiency

FIPS: $\mathbf{v} = \mathbf{U}\mathbf{w}$

We value users in the vector $\boldsymbol{v}$ by linear valuation.

- $\boldsymbol{U}$ a matrix with rows as user features: estimated **probability** of attendance, **fairness score** given by an exponential decay scheme (based on recent invites), and binary indicator of **pantry status**.
- $\boldsymbol{w}$ a weight vector to prioritize user features

booQ:

$$Q(s_{t-1}, a_{t-1}) \leftarrow (1-\alpha)Q(s_{t-1}, a_{t-1}) + \alpha(r + \gamma \max_a Q(s_t, a))$$

To correct for biased probability estimates we use Q-Learning to adjust for an ideal booking factor to limit food-waste and user frustration. At a high level:

- A reward/punishment framework is used to teach an agent about its actions within an environment.
- The agent's state space (waste) is observed/ranked.
- Rewards are given based on change in ranking, encouraging effective adjustments to $\boldsymbol{\omega}$.

Experiments

Datasets: A simulated user dataset of 1000 users having:

- An estimated **probability** of attendance (from a truncated normal distribution)
- A true or false **Pantry** indicator for food-insecurity (respectively 0 or 1)
- A recent history to compute a **fairness score** via exponentially decaying scheme

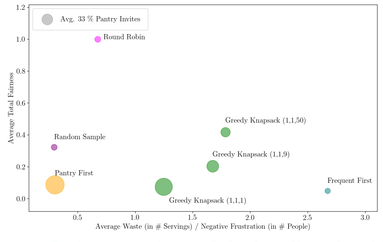

Figure 5: A comparison of the baseline algorithms with the **proposed method** (denoted by **Greedy Knapsack**). Weights in parentheses are in order (**probability**, **pantry**, **fairness**).

A simulated event data set of 100 events has:

- a number of servings from a truncated normal distribution (mean 40 and deviation of 10)

Train vs. Test: Algorithms such as **booQ** that require training are run for 1000 iterations over event/user training sets. A test set with identical parameters and underlying distributions is used for 100 iterations; test results are aggregated.

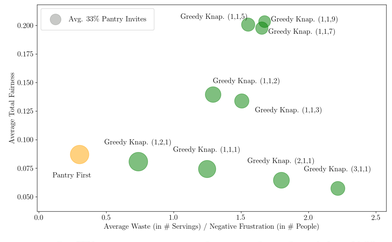

Figure 6: The **proposed method** with different weightings. Weights in parentheses are in order (**probability**, **pantry**, **fairness**). This demonstrates response to weighting and capability to prioritize different user features in notification.

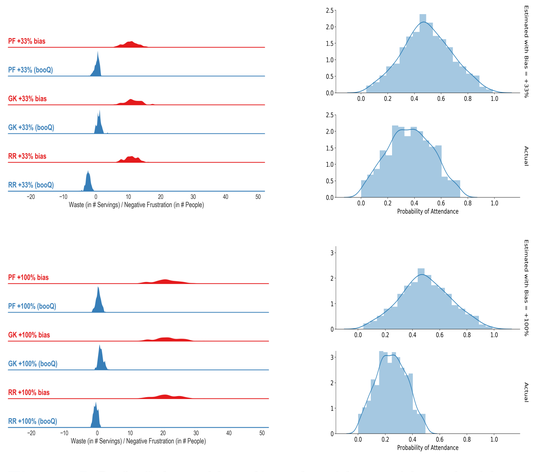

Figure 7: Left: Selected baseline algorithms with and without **booQ** in the presence of positive probability estimation bias. **booQ** learns and maintains minimal food waste. Right: Visual representation of positive bias introduced to user probability estimation.

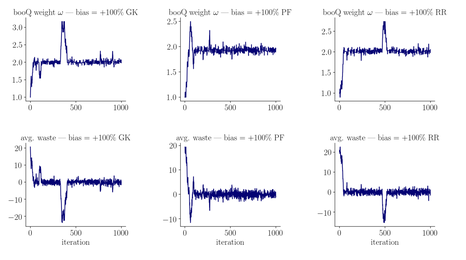

Figure 8: Displays **booQ** learning with various user selection algorithms. **booQ** is able to quickly find the optimal booking factor and recover from mistakes due to probabilistic nature of the Q-Learning algorithm.

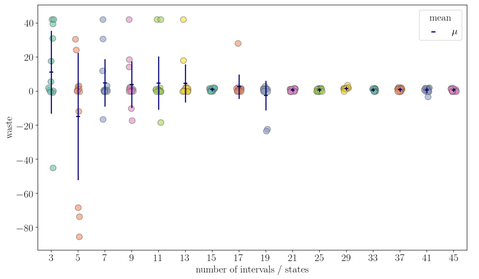

Figure 9: Analysis of **booQ** as the number of intervals or discretized states for ranks in reward framework varies. Shown are waste measurements for 10 trials of each interval count with mean and standard deviation. The modified knapsack framework is used to select users. **booQ's** consistency and resolution improve when the number of intervals is greater or equal to 21.

Acknowledgments

The authors would like to thank an early contributor to application development, David Tsui, as well as the anonymous reviewers for their helpful suggestions on additional experimentation. This work was funded in part by NSF Award CNS-1739413.