A Recommender System for Crowdsourcing Food Rescue Platforms

Zheyuan Ryan Shi
ryanshi@cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Leah Lizarondo
412 Food Rescue
Pittsburgh, PA, USA

Fei Fang
feif@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

ABSTRACT

The challenges of food waste and insecurity arise in wealthy and developing nations alike, impacting millions of livelihoods. The ongoing pandemic only exacerbates the problem. A major force to combat food waste and insecurity, food rescue (FR) organizations match food donations to the non-profits that serve low-resource communities. Since they rely on external volunteers to pick up and deliver the food, some FRs use web-based mobile applications to reach the right set of volunteers. In this paper, we propose the first machine learning based model to improve volunteer engagement in the food waste and security domain. We (1) develop a recommender system to send push notifications to the most likely volunteers for each given rescue, (2) leverage a mathematical programming based approach to diversify our recommendations, and (3) propose an online algorithm to dynamically select the volunteers to notify without the knowledge of future rescues. Our recommendation system improves the hit ratio from 44% achieved by the previous method to 73%. A pilot study of our method is scheduled to take place in the near future.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; Computational advertising; • **Applied computing** → *Economics*.

KEYWORDS

Food rescue, food waste, food security, recommender system, online planning, AI for social good

ACM Reference Format:
Zheyuan Ryan Shi, Leah Lizarondo, and Fei Fang. 2021. A Recommender System for Crowdsourcing Food Rescue Platforms. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3442381.3449787

1 INTRODUCTION

Recall the last time you saw several full shelves of bread with an expiry date in two days at a grocery store, or the last time you saw a homeless person downtown asking for a meal. It is not a coincidence if both scenarios seem familiar to you. The simultaneous food waste and food insecurity are a serious problem shared by many parts of the world [11, 19]. Unfortunately, the ongoing COVID-19 pandemic is only making things worse [22]. Even after the pandemic hits its peak, the increased struggle with basic food security will not subside quickly on our long way back to normal. Thus, now more than ever, there is an urgent call for action to address the food security and food waste problem. In this paper, we leverage our AI expertise to answer this call with our collaborators.

Food rescue organizations (FR) are a major non-profit force in fighting food waste and insecurity. They match fresh, unexpired food from donors to organizations serving low-resource communities, thereby facilitating food redistribution.[1] FRs rely on volunteers to transport the food. To engage with the volunteers, FRs use a web-based application on cell phones to post information of upcoming rescues. Please refer to Section 3 for a detailed description of the food rescue operation. This "crowdsourcing" paradigm has proved to be successful in engaging with the general public to address food waste and insecurity [41].

However, relying on external volunteers to deliver the food comes with inherent uncertainty. What if no volunteer will claim the rescue? This uncertainty is prevalent in FR operations and it has serious consequences such as lost faith in the program from the donor and recipient organizations. Since the primary way in which FRs contact volunteers is through push notifications, to improve the claim rate one would certainly want to send push notifications to volunteers who are likely to claim the given rescue. Currently, when a rescue is published, the mobile app sends notifications to volunteers who are within a certain radius of the donor. Although being close to the donor is clearly a positive factor, this is far from a perfect solution as its hit ratio is only 44%, which means it misses the "correct" volunteer more than half of the time. On the other hand, we also want to avoid sending push notifications to every volunteer all the time, because that would easily drive them away from the platform or prompt them to disable notifications altogether [14]. A customized push notifications, with good justifications, would better engage the user. Thus, it is crucial to send the push notifications to a selected set of volunteers who are likely to claim the rescue.

In this paper, we propose the first machine learning based model to select the right set of volunteers to notify in the food waste and security domain. By treating each rescue trip as a "user" and each volunteer as an "item", we study this problem from a recommender system perspective. Recommender systems have received a lot of interest from the WWW community in the past years [12, 21, 26, 45]. Our task is relevant to this literature but also brings several new challenges. We state these challenges and our approaches to address them as follows.

[1] We would like to emphasize that only fresh and unexpired food can be donated through FRs.

- First, since each rescue only happens once, we stay in the "cold start" phase of the recommender system forever, rendering collaborative filtering-based methods unsuitable. We leverage a sophisticated set of contextual features, an adaptive under-sampling technique, and a neural architecture to develop a content-based recommender system. We show that our model outperforms a number of baselines, and improves the hit ratio of recommending the correct volunteer to 73%. This is a 66% improvement from the current practice which has a 44% hit ratio.
- Second, not being able to recommend diverse items is a serious issue in the recommender systems literature. It is particularly concerning in our application because the "items" are human volunteers who contribute their time to the cause. We leverage a mathematical programming based approach by imposing diversity constraints on the output of the recommender system. This ensures that each volunteer receives only a limited number of push notifications every day.
- Third, most literature on recommender systems assumes an offline environment that has a static dataset. However, food donations arrive sequentially and thus the FR must accordingly make decisions without the knowledge of future rescues. We identify an important arrival pattern of the food rescue trips based on our experience in the domain. Relying on this insight, we develop an online planning algorithm which sequentially selects the volunteers to notify, while still satisfying the diversity constraint we imposed earlier. We show that our online algorithm achieves a hit ratio that is only 10% worse than the hypothetical offline mode where we assume knowledge of all the rescues at the beginning of the day.

Food rescue organizations have made their presence in most major cities in the US and beyond. In the US alone, there are already over 50 cities where FRs are providing basic necessities to the communities, affecting over a million people. We are working with 412 Food Rescue (412FR), a large food rescue organization in Pittsburgh. [2] Since its incorporation in 2015, 412FR has served over 1,000 nonprofit partners and has grown a network of over 15,000 volunteers in the Greater Pittsburgh Region. The models we describe in this paper are in the process of being deployed at 412FR. Furthermore, we believe that the problem we tackle is not limited to this particular application: it can be adapted to many domains with a crowdsourcing type of operation that relies on volunteers to perform the task.

2 RELATED WORK

There is a growing literature on using AI or related tools to study the food rescue operation. Some formulate a vehicle routing problem to match the donation with recipients [20, 31], while others tackle the problem from a fair allocation and market design perspective [5, 27, 35]. Because the demand and supply or food are external to the FR, some works are focused on forecasting the future food supply [32, 34]. While all these works provide useful insights into the FR operation, the existing literature largely misses the volunteer side of the process. Among the few pieces of work that explicitly consider the volunteer crowdsourcing aspect of food rescue, Lee et al. developed a participatory democracy framework to allow volunteers and other stakeholders to decide on the matching of donations and recipients, which is orthogonal to our work [24]. Shi et al. developed a machine learning model to predict whether a rescue trip will be claimed and an optimization model to find the best intervention scheme [38]. Our work complements theirs and both can be used simultaneously by FRs. Yet we advance from them in two aspects. First, compared to their predictive model as a decision aid for downstream human interventions, our recommender system directly improves the upstream notification process which can reduce the need for the costly human intervention. Second, compared to their prescriptive model which sets system-level notification parameters, our recommender system is rescue-specific, thereby leveraging more information to make better decisions. Finally, Manshadi and Rodilitz design online volunteer notification algorithms in a similar setting [28]. Compared to their work, we take a pure data-centric approach and make few modeling assumptions about the volunteer and rescue patterns, with the sole purpose of finding the most likely volunteers in the real-world use case of this system.

The literature on recommender systems is vast and we will only discuss two topics relevant to our work – cold start and diversity. Cold start refers to the scenario where there is no previous label on a new user or new item [37]. An active approach to deal with cold start would be to interact with the user to request labels, which is often framed as a bandit problem [10, 18, 25, 36, 39, 44]. This is clearly not applicable to our setting, since each rescue is a one-shot business and the stake is too high for real-world trials. Thus, we turn to passive approaches which make do with the data we have. Content-based approaches are a natural candidate for this challenge. Collaborative filtering is not designed to perfectly handle cold start, though there have been methods to enhance it with side information towards addressing this problem [8, 9, 16]. In this paper, we propose a content-based model for the following reasons. First, in our food rescue setting, cold start is not just a short unpleasant period at the beginning which might imply secondary concern. Instead, we stay in the cold start phase forever because every rescue (user) is new. Thus, handling it perfectly with content-based models is of utmost consideration. Second, we have identified a good set of interaction features based on our experience in the food rescue operation. Third, there is no collaborative filtering-based system currently in place that holds us back from using a content-based approach. Recent advances in leveraging neural architecture in recommender systems serve as a starting point for us to build our model [21].

Our problem is also related to the diversity of recommender systems, which concerns both individual diversity and aggregate diversity. Individual diversity refers to recommending diverse items within the recommended list for each user [43, 45]. Aggregate diversity refers to recommending diverse items between the recommended lists for different users [1, 7, 15, 30, 33]. In our problem, we hope to avoid sending push notifications to a small subset of volunteers (items) all the time. Thus, our problems concerns the aggregate diversity. At the technical level, our method of diversifying the recommendations can be considered as a variant of the integer programming approach by Adomavicius and Kwon [1].

[2]https://412foodrescue.org/

However, our problem has an additional subtlety that makes it more challenging, as we discuss below.

In food rescue, each rescue trip arrives sequentially and thus we need to make the recommendation decision in an online fashion. This brings additional challenge to our effort to diversify the recommendations. This problem resembles the well-studied online adwords matching problem [3, 17, 29], and could fit under the more general online linear programming framework [4]. However, these works typically only guarantee asymptotic results or require prior knowledge of the number of rescues on any given day, which makes them impractical in our setting. There is also a literature on budget pacing in online advertisement [2, 23, 42]. Our application domain and the central problem are different from online advertisement, but our online planning for push notification budget can be considered as a novel way of pacing.

3 ANATOMY OF FOOD RESCUE OPERATIONS

Food rescue organizations serve as an intermediary between the food donors and recipient organizations. Donors, typically grocery stores and restaurants, would call the FR when they have food items that they want to donate. After receiving the call, the FR dispatcher matches this donation with some recipient organization, typically some non-profit organization that serves a low-resource community. Once this matching is done, the dispatcher posts this matching on the FR's mobile app. Hereafter, the food rescue process becomes visible to the volunteers. As shown in Figure 1, a volunteer, who has the FR's mobile app installed on the phone, will then receive a push notification about the rescue. If they choose to claim it on the app, the app would provide them with the detailed information instructing them where to pick up the donation and where to deliver. The volunteer then goes out to complete the rescue trip.

Of course, the workflow described above is an ideal scenario. In reality, occasionally, some rescue trip stays unclaimed on the mobile app for a long time. FR dispatchers want to prevent this situation as much as possible, since each rescue comes with a deadline which is bounded by the nature of the food and the operation hours of the donor and recipient. Unclaimed rescues discourage the donors and recipients from participating in the program in the future. FRs have two ways to address this problem. First, it sends push notifications to possible volunteers to advertise the rescue. Second, the dispatchers might individually call some regular volunteers to ask for help. In a previous work, Shi et al. focus on the latter approach in order to help the dispatcher's decision-making [38]. We focus on the former by directly finding the best set of volunteers to send push notifications to.

4 DATA

To develop a recommender system, we need both positive and negative labeled examples. A positive example means that a particular volunteer (item) claims a particular rescue (user); a negative example means otherwise. In this section, we detail our data acquisition, labeling, and feature engineering process.

4.1 Positive Labels

We obtained the rescue database from 412FR, covering the period from March 2018 to March 2020. The database keeps the log of each rescue. For most rescues, the database logs its timestamps from being drafted by the dispatcher, to being published on the mobile app, to being claimed and completed by a volunteer. For these rescues, we simply take the rescue plus the volunteer who claimed it as a positive data point. However, the food rescue operation is not always so neat. Occasionally, the dispatcher knows ahead of time that some volunteer would do the job, so they directly assign the volunteer for a particular rescue and bypass the app notification stage. In this case, we take this direct assignment as a positive example as well. Sometimes a volunteer might claim a rescue and then drop it, causing some rescue to have multiple volunteers in the log. In this case, we create our labels based on the last volunteer.

4.2 Negative Labels

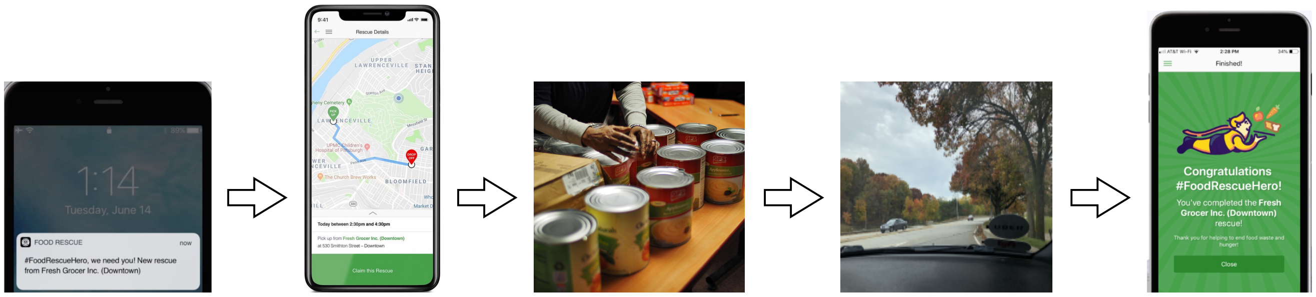
A negative example means that a particular volunteer did not claim a particular rescue. Since almost all rescues have only one volunteer who claimed the rescue, obviously most of our data points will have negative labels. However, not all of these negative data points are necessarily true, because perhaps a volunteer would have claimed some rescue if someone else had not claimed it 10 minutes in advance. Thus, we use the following ways to construct a selected negative dataset. First, in the time period covered by our database, 412 Food Rescue used a mobile app push notification scheme which notifies volunteers within 5 miles when the rescue is first available and then notifies all volunteers 15 minutes later if the rescue has not been claimed. Thus, if a rescue is claimed within 15 minutes, we only treat the volunteers who were within 5 miles and did not opt out of push notifications as negative examples.

We also incorporate another data source to strengthen our negative sampling. In addition to mobile app notifications, the dispatcher at 412FR also manually call some regular volunteers to ask for help with a specific rescue. This usually happens when some rescue has been available for over an hour yet nobody has claimed it. We obtained the call history from 412FR, from which we identify the volunteers they reached out to within the time frame of each rescue. If these volunteers did not claim the rescues in the end, we treat them as negative examples. Compared to the negative examples derived from push notifications, we have more confidence in this set of negative examples, since declining on a phone call is a stronger indicator than ignoring a push notification.

4.3 Feature Engineering

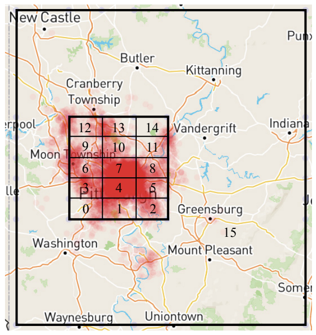
Based on our extended collaboration with 412FR, we carefully identify a selected set of useful features that are relevant in the food rescue operation.

First, the experience of food rescue dispatcher indicates that if a volunteer has completed a rescue at or near a donor or recipient, they are more likely to do a rescue trip again in the neighborhood. As shown in Figure 2, we divide the Greater Pittsburgh Region into 16 cells. We evenly divide a central rectangular region into a 3×5 grid, and label them grid cells 0 through 14. Then, we label the entire map outside the rectangular region cell 15. The rationale is that in the outer suburbs there are fewer donors, recipients, and volunteers, and furthermore volunteers who in suburbs are more willing to do long-distance, i.e. inter-cell, rescue than volunteers in downtown. For each rescue trip and each volunteer, we calculate

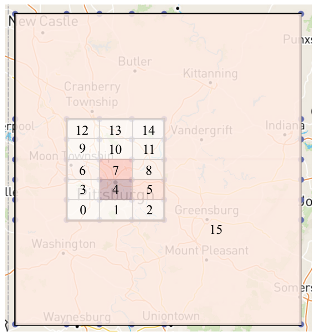

1:14
Tuesday, June 14
FOOD RESCUE
now
#FoodRescueHero, we need you! New rescue from Fresh Grocer Inc. (Downtown)
Rescue Details
Today between 2:30pm and 4:30pm
Claim this Rescue
Finished!
Congratulations #FoodRescueHero!
Close

(1) Receive push notification (2) Claim on mobile app (3) Pick up from donor (4) Deliver to recipient (5) Success!

Figure 1: The workflow of a food rescue operation from the volunteer's perspective.



(a) Distribution of donor organizations. Darker colors mean more frequent donations. We plot the donor locations with random perturbations.



(b) Density of recipient organizations. Darker colors mean more recipient organizations in the grid.

Figure 2: We divide the Pittsburgh area into 16 grid cells, with cells 0–14 covering downtown Pittsburgh and its neighborhoods, and cell 15 containing the rest of the region.

the number of rescues the volunteer has done in the rescue donor's cell, in the rescue recipient's cell, and across all cells. These counts are only up to the date of the given rescue, so that we could prevent data leakage. We also tried to include as features the volunteer's historical rescues in each cell, not just the donor's and recipient's cell. However, they did not contribute any predictive power and thus we leave them out of the final model.

Closely related to this is the distance between the volunteer and the donor. It is unlikely that a volunteer would drive 30 miles to pick up a donation, as we show in Figure 3a. We measure the distance using the straight line distance based on geographic coordinates. Although the actual traveling distance might be a better indicator, we observe that the straight line distance already serves our purpose.

Aside from the geographical information, the length of time between volunteer's registration on the platform and the rescue is also an important factor, as suggested by our collaborators at 412FR. We plot the histogram of this variable in Figure 3b. Immediately after registration, the volunteer is eager to claim a rescue to get a feel of the food rescue experience. If a volunteer has stuck with the program for an extended period and remains active, it is likely that they are a regular and dependable one as well, which is substantiated with the upward trend and plateau in Figure 3b around days 300–600. Thus, we include this feature in our prediction model.

Weather information is also an important factor in the prediction. Presumably rainy and snowy days would see a lower volunteer activity in general. However, the impact of inclement weather would fall disproportionately on volunteers who do not have a car or live in suburban areas. We use the Climate Data Online (CDO) service provided by the National Oceanic and Atmospheric Administration to access the weather information.[3] The CDO dataset contains weather information at the discretization level of days and weather station. There are multiple weather stations in the Pittsburgh area and for each rescue we select the data for the date of rescue and the station that is closest to the donor organization. As shown in Figure 3c, on wet days, relatively more volunteers who claim the rescue reside in downtown Pittsburgh (cell 4 and 7). Whereas on dry days, a lot more volunteers who live in the outer suburbs of Pittsburgh (cell 15) are active. In fact, we also saw a significant difference in the average distance between volunteer and donor for dry days (5.94 miles) and rainy days (5.22 miles), with a t-test p-value 3×10^{-8}.

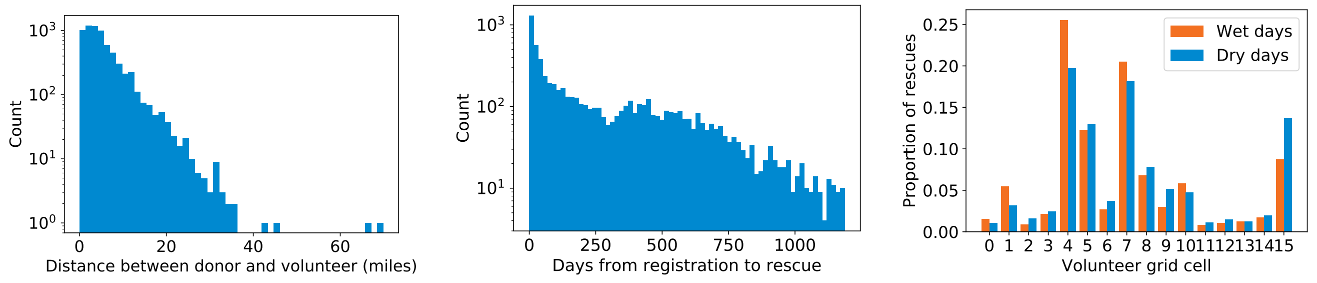
We also explored a number of other features but did not incorporate them into our final model. These features include the rescue's time of day and day of week, the volunteer's availability, whether the volunteer uploaded an avatar to their profile or not, whether the volunteer is located in the same grid as the donor or recipient, and so on. Although these are intuitive factors, we did not find them improve the predictive power of our model and hence left them out.

5 RECOMMENDER SYSTEM

We build a neural network-based recommender system. We first detail our network architecture, and then discuss our approaches to address the unique challenges in the food rescue domain.

We show the neural network architecture in Table 1. The input to the neural network is the feature vector of a rescue-volunteer pair. The feature vector passes through four dense layers. Each layer is followed by a ReLU activation function, except for the last layer where we output a single number which is then converted

[3]https://www.ncdc.noaa.gov/cdo-web/



(a) Histogram of rescues, based on the distance between the donor and the volunteer who claimed the rescue. **(b) Histogram of rescues, based on the length of time between the rescue and the registration of the volunteer who claimed the rescue.** **(c) Histograms of rescues under wet and dry weather, based on the location of the volunteer who claimed the rescue.**

Figure 3: Data analysis results.

Layer	Operation	Hidden Units
1	Dense (ReLU)	384
2	Dense (ReLU)	2048
3	Dense (ReLU)	512
4	Dense (Logistic)	16

Table 1: Neural network architecture

to a number between 0 and 1 by the logistic function. This output represents the likelihood that this volunteer will claim this rescue trip. We use the cross entropy loss to train the neural network. To output a list of k volunteers to whom we send push notifications for a particular rescue at prediction time, we pass the feature vectors of the rescue-volunteer pairs for all volunteers on a fixed rescue through the network and rank the output to take the top k of them.

5.1 Negative Sampling

As mentioned earlier, there is an extremely high label imbalance in our dataset. From March 2018 to March 2020, there are 6757 rescues available for the training. Each rescue typically has only one volunteer who claimed it, and there are 9212 registered active volunteers in the Pittsburgh area. This means, theoretically, the ratio between negative and positive examples is over 9000 : 1. Using the method introduced in Section 4.2, we can obtain a selected set of negative examples D_n derived from push notifications and another set of negative examples D_c derived from dispatcher calls. The set D_c is slightly smaller than the positive examples D_p, while $|D_n| : |D_p| \approx 700 : 1$. When training the neural network, we always use all the examples from D_p and D_c. However, we randomly sample a subset of examples from D_n at each episode of the training. By doing this, we ensure that the negative examples from D_n do not dominate the training set, and at the same time the "more certain" negative examples from D_c gets emphasized more than D_n. This whole procedure leads to an overall ratio between negative and positive samples around 3 : 1 in each single batch.

5.2 Diversity and Online Planning

Recommender systems in general suffer from the diversity issue, where "hot" items get recommended to all the users. In commercial applications, this might lead to the "rich gets richer" phenomenon on superstar items and the missed revenue opportunity on the less popular items. All these are valid. However, as we have emphasized several times in this paper, the "items" on the other side of our recommender system are humans. The aforementioned consequences of the lack of diversity is only going to be more problematic in our case. If a popular volunteer received push notifications for every single rescue throughout the day, they would possibly get annoyed and mute the notifications. On the other hand, for volunteers who are already not very active, if our system never sent them push notifications, they would probably just forget about the platform and would be unlikely to return. Therefore, it is crucial that we properly handle the diversity issue.

We distinguish between two notions of diversity: individual diversity and aggregate diversity. The former means that each user (rescue) gets recommended a diverse set of items (volunteers). The latter means that the recommended items (volunteers) across different users (rescues) combined cover a large portion of the item space. Our human-centric approach determines that we focus on aggregate diversity here. In fact, we focus on a slightly different metric: how many times each volunteer gets recommended for a rescue every day. We wish to put a cap on this metric, which is directly linked to the user experience of each volunteer.

To this end, we can formulate the following mathematical program for a given day of food rescue operation.

$$
\begin{aligned}
(\Pi) \quad \max_{x} \quad & \sum_{i \in R} \sum_{j \in V} p_{ij} x_{ij} \\
s.t. \quad & \sum_{j \in V} x_{ij} \le k, \quad \forall i \in R \\
& \sum_{i \in R} x_{ij} \le b, \quad \forall j \in V \\
& x_{ij} \in \{0,1\}, \quad \forall i \in R, \forall j \in V
\end{aligned}
$$

Let V denote the set of volunteers. On a particular day, we have a set of rescues R. The binary decision variable x_{ij} is equal to one if we decide to send push notification to volunteer j about rescue i. The first constraint indicates that for each rescue we will notify the top k volunteers, as introduced at the beginning of Section 5.

Algorithm 1: Online Planning for Optimizing Push Notifications

```
  input: A trained neural network predictor
1 while a new rescue $i$ arrives do
2     Flush $X_i$
3     for $dayToSample = 1, 2, \ldots H$ do
4         Sample the set of rescues $R$ on the dayToSample that
            occured from the time of the current rescue $i$ till
            the end of the day.
5         Compute predicted claim probabilities $p_{ij}$ and $p_{i'j}$
            for all $i \in R$, for all $j \in V$.
6         Solve the following optimization problem:
```

$$
\begin{aligned}
(\Pi_i) \quad \max_x \quad & \sum_{j \in V} \left(p_{ij} x_{ij} + \sum_{i' \in R} p_{i'j} x_{i'j} \right) \\
s.t. \quad & \sum_{j \in V} x_{i'j} \le k, \quad \forall i' \in R \\
& \sum_{j \in V} x_{ij} \le k \\
& x_{ij} + \sum_{i' \in R} x_{i'j} \le b_j, \quad \forall j \in V \\
& x_{ij} \in \{0, 1\}, \quad \forall i \in R, \forall j \in V
\end{aligned}
$$

```
7         Keep in $X_i$ the optimal solution $x^*_{ij}$ for the current
            rescue only.
8     Sum $X_i$ over the sampled histories, find the top $k$
        volunteers.
9     Send push notifications to them. Update the remaining
        budget $b_j$ for each volunteer $j$.
```

The second constraint is our diversity constraint, which makes sure that each volunteer receives at most b push notifications a day. The p_{ij} in the objective is the output from our trained neural network, representing the predicted likelihood that volunteer j is going to claim rescue i.

While this optimization problem Π is a valid method to improve diversity in generic recommender systems, it does not solve the problem in our setting. The reason is that donations, and hence food rescue trips, arrive in our system sequentially throughout the day, and the dispatcher must also act in real-time. It is unacceptable to wait till the end of the day, run the optimization problem above, and then send the push notifications. Therefore, we need an online algorithm.

An intuitive approach is to resort to the literature on online linear programming [1]. Indeed, we could imaging solving π where at each time step, a new rescue is revealed with a new column in the x matrix and p matrix. However, we do not know how many rescues there will be at the beginning of the day. This is a major obstacle in applying the established algorithms with theoretical guarantees. Instead, the daily rescue pattern is hardly adversarial in nature and thus we propose a simple heuristic, as shown in Algorithm 1.

In Algorithm 1, when a food rescue arrives in the system, we sample the historical rescue data for trajectories. Typically, we would sample the rescues on the same weekday a week ago, two weeks ago, and so on. The underlying idea is that the same weekdays might have similar rescue patterns. This is because most donations that come to 412FR come from grocery stores or large companies/universities. Grocery stores often perform inventory counts on a weekly basis. Companies and universities often hold weekly events, with catered food. For each sampled day, we only take the trajectory from the time of the current rescue to the end of the day. Then, for each trajectory along with the current rescue, we obtain the neural network's predicted claim probabilities and solve the optimization problem Π_i. Π_i is similar to Π except that each volunteer now has their own remaining budget of push notification. Note that now everything in Π_i is observed and known, whereas in Π the future rescues are unknown at the decision-making time. We keep only the part of the optimal solution that concerns the current rescue and discard the rest. Later, on Line 8 in Algorithm 1, for each volunteer, we sum over its value in the optimal solution across all the sampled trajectories. We take the top k volunteers as voted by these solutions, who become the ones we will send push notifications to for this current rescue.

We note that the optimization problem Π and Algorithm 1 are extremely flexible to account for many additional considerations. For example, we could use personal budget b_j in Π and add additional constraints to represent the volunteer's push notification preferences. We could also add weights to the objective function to emphasize the importance of a particular rescue.

6 EXPERIMENTS

6.1 Recommender System

We use a training set containing rescues from March 2018 to October 2019, which is 80% of the entire dataset. We use the remaining 1373 rescues from November 2019 to March 2020 as the test set. We conducted all of our experiments on an Intel i7-7700K 4.20GHz CPU with 64GB RAM.

First, we only consider the prediction part of our algorithm. We compare our neural network recommender system with several competitive baselines that are commonly used, including random forest (RF), gradient boosted decision trees (GBDT), and stacking model (SM). To determine the hyper-parameters of the baseline models and the neural network model, we separate a validation set which consists of the last 1/8 of our training set and then run a grid search according to the performance on the validation set. For experiments on the baselines, we use the same negative sampling method on D_c and D_p as described in Section 5.1. As for negative examples from the app notifications D_n, since the baselines are not gradient descent-based methods, we sample them in two schemes such that the ratio between the positive and negative examples is roughly $1:1$ and $1:20$, respectively. We consider the latter because that is roughly the number of negative examples that the neural network approach has seen throughout the training, in order to ensure a fair comparison.

We show the results of all these algorithms on the test set, averaged over 5 runs, in Table 2. We consider the hit ratio at k (HR@k) and the normalized discounted cumulative gain at k (NDCG@k) in Table 2. However, we note that our main metric of interest is the hit ratio, because when sending push notifications, we do not care about the particular order in which each volunteer ranks on the

Model	HR@k (SD)	NDCG@k (SD)
NN	**0.7269 (0.0310)**	**0.1898 (0.0147)**
RF(1:1)	0.5989 (0.0395)	0.1319 (0.0303)
RF(1:20)	0.6035 (0.0511)	0.127 (0.0053)
GBDT(1:1)	0.6235 (0.0549)	0.1613 (0.0098)
GBDT(1:20)	0.5394 (0.0152)	0.1023 (0.0086)
SM(1:1)	0.4996 (0.0005)	0.1332 (0.0002)
SM(1:20)	0.5219 (0.0125)	0.0948 (0.0030)
Default	0.4392 (N/A)	N/A (N/A)

Table 2: The performance of the neural network based recommender system and several baselines. All experiments are repeated five times with the mean and standard deviation shown in the table.

list. Also because of this, HR@k is our primary metric during the grid search on hyper-parameters for all the predictive models. We choose the value of k to be 964, since this is the average number of push notifications sent per rescue under the current notification scheme. The current distance-based notification scheme has a hit ratio of 0.4392. All baselines show better performance than the current method, with random forest and GBDT being better than the stacking model. However, the neural network based prediction model outperforms all the baselines.

The hit ratio of the neural network model is a 66% improvement over that of the current distance-based method. This means that we would be able to reach the would-be volunteer in approximately 900 more rescues every year. Each of these rescues has a donor and a recipient organization that serves tens or hundreds of people behind it. A smooth food rescue experience would not only provide basic food necessities to these people, but also encourage these organizations to keep up the engagement in a sustainable way.

6.2 Diversity and Online Planning

As mentioned in Section 5.2, recommender systems, in general, suffer from the diversity issue. This problem also exists in our model. In Figure 4, we plot the histogram of the number of push notifications received by each volunteer for the test set rescues. The neural network based recommender system, shown in yellow in Figure 4, exhibits an alarming bimodal distribution: most volunteers either receive almost no push notifications, or receive push notifications for almost every single rescue. We remark that although the number of volunteers in the rightmost bin (446 out of 9312) is much smaller than that in the leftmost bin (7458 out of 9312), the former is much more concerning. This is because they are typically the most "active" volunteers who have contributed the most to the food rescue program. In fact, these 446 volunteers contain 39 of the top 50 most frequent volunteers, and 51 of the top 100. If they left the platform due to too many notifications, which is likely to happen should the proposed recommender system get deployed, the loss to 412FR would be disproportionately high. On the other hand, the default distance-based notification scheme does not suffer from this issue, as shown in red in Figure 4. Although the majority of the volunteers still receive few push notifications, the notification frequency for each volunteer is capped at roughly once every two rescues.

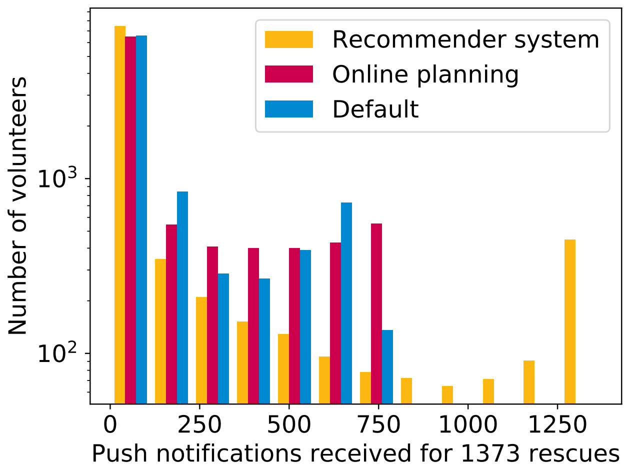


Figure 4: Histograms of the number of push notifications received by each volunteer over all the 1373 rescues in the test set. The online planning algorithm has a budget of 6 notifications per day.

Figure 4 serves as a stern warning against the premature deployment of machine learning algorithms in the real world. That a certain model outperforms the current practice by 66% in some important metric (here, the hit ratio) does not mean it would not cause other problems.

We use our Algorithm 1 to improve the diversity of volunteer recommendations. As a preliminary and straightforward comparison, we ran our online planning Algorithm 1 with budget $b = 6$ push notifications per day using the rescues seven days ago as the sampled history. We plot its notification histogram in yellow in Figure 4. It is easy to see that the online planning algorithm achieves a push notification distribution much more similar to the default scheme, than the recommender system alone. It completely avoids sending push notifications about every single rescue to any particular volunteer.

Indeed, the effect of Algorithm 1 on recommendation diversity depends on the budget parameter b. In Figure 5, we plot the notification distributions for different choices of the budget value, and compare them against those of the recommender system and the default notification scheme. As the budget increases, the distribution of push notifications from Algorithm 1 approaches that of the recommender system. We note that the position of the rightmost peak of each histogram should not be interpreted as an indicator of the total number of push notifications sent. In all of these experiments, we limit the number of notifications for each rescue at $k = 964$. Except for when the budget is extremely small, the algorithm always notify exactly 964 volunteers for each rescue. The diversity goal here is to make the histogram occupy as little space as possible on the right side of the figure.

Much as we demonstrate the improvement of recommendation diversity, we would also like to ensure that the recommendation accuracy of our algorithm does not drop too much. The budget parameter b captures the inherent trade-off between diversity and

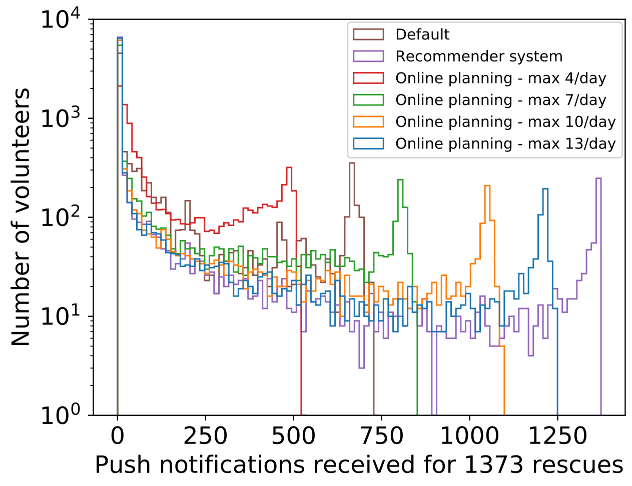


Figure 5: Histograms of the number of push notifications received by each volunteer over all the 1373 rescues in the test set, compared across different budget values.

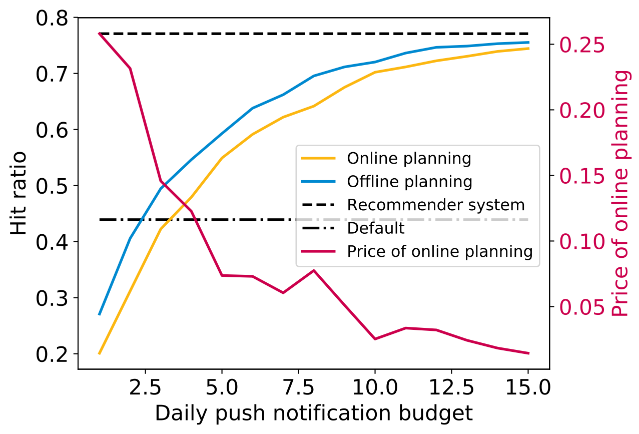


Figure 6: Hit ratio of the online planing algorithm. Price of online planning, computed as $1 - \frac{HR_{\text{online}}}{HR_{\text{offline}}}$, is shown on the right axis.

accuracy. As we show in Figure 6, the yellow curve represents the hit ratio of Algorithm 1. Algorithm 1 outperforms the existing notification scheme when the budget is more than four notifications per day, which is a relatively trivial amount. When the budget rises to 10 notifications per day or more, the hit ratio is very close to the bare bone recommender system.

In order to further evaluate the quality of *online* planning in Algorithm 1, we also solve an offline version of the problem, where we solve the mathematical program Π separately for each day, assuming full information about the rescues on that day. We show the hit ratio of the recommendation decision from this offline version in blue in Figure 6. Since having full information is always better, the blue curve always lies above the yellow curve representing the online planning. However, the difference is not big. We term the difference as the "price of online planning", which is computed as $1 - \frac{HR_{\text{online}}}{HR_{\text{offline}}}$. In fact, Figure 6 shows that the price of online planning is decreasing as the budget grows, and is consistently smaller than 0.1 when the algorithm is of potential deployment interest (performing better than the current practice). This validates our earlier claim in Section 5.2 that the rescues on the same weekday of the previous week are a reasonably good indicator of the rescues on the present day.

7 CONCLUSION AND FUTURE DIRECTIONS

A critical goal in the food rescue operation is to be able to reach the "right" volunteers in time. Working with 412 Food Rescue, we developed a machine learning model to recommend the most probable volunteers to send push notifications to for each given rescue. Our machine learning model improved the hit ratio of the current notification scheme by 66%. The food rescue operation features two main challenges: the recommendation diversity is of utmost importance to ensure volunteer experience, and the recommendation must be made in an online fashion. We proposed a novel algorithm to dynamically recommend volunteers for rescues in real time, while diversifying the recommendations and still managing to keep the hit ratio well above the current practice.

The problem of low hit ratio is a real problem that needs to be addressed. This is also a problem natural for a data-driven approach, and we do have the relevant data available. There is an existing approach to this problem (distance-based notification), so our technological intervention does not introduce new initiatives. Instead, we amplify the existing initiative. Lots of previous endeavors have shown that this amplification approach is more likely to achieve deployment and sustainable impact [40]. In fact, our technological intervention does not replace, reduce, or attempt to dictate any human employee's job at the FR.

There are two immediate future directions that are useful in the food rescue operation. First, our Algorithm 1 features a classical predict-then-optimize framework where the learning objective and the optimization objective are not perfectly aligned. It would be interesting to consider the recent literature on data-driven optimization [6, 13] to further improve the results shown in Figure 6. Of course, the online nature of our problem brings an additional interesting challenge that has never been addressed in that literature. Second, recommendation is necessarily limited by the counterfactuals. In Section 4.2, we proposed several approaches to select the most credible negative examples. It would be interesting to identify credible positive examples beyond the explicitly labeled ones, which are rather scarce in the dataset.

A pilot study of the model and algorithm described in this paper is scheduled to take place in the near future.

ACKNOWLEDGMENTS

We sincerely thank our collaborators at 412 Food Rescue for contributing their time, expertise, experience, and dataset. They are the unsung heroes behind this work. We thank 412 Food Rescue and other food rescue organizations worldwide for supporting our communities at this critical time. This work is supported in part by NSF grant IIS-1850477.

REFERENCES

[1] Gediminas Adomavicius and YoungOk Kwon. 2014. Optimization-based approaches for maximizing aggregate recommendation diversity. *INFORMS Journal on Computing* 26, 2 (2014), 351–369.

[2] Deepak Agarwal, Souvik Ghosh, Kai Wei, and Siyu You. 2014. Budget pacing for targeted online advertisements at linkedin. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1613–1619.

[3] Gagan Aggarwal, Ashish Goel, and Rajeev Motwani. 2006. Truthful auctions for pricing search keywords. In *Proceedings of the 7th ACM conference on Electronic commerce*. 1–7.

[4] Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. 2014. A dynamic near-optimal algorithm for online linear programming. *Operations Research* 62, 4 (2014), 876–890.

[5] Martin Aleksandrov, Haris Aziz, Serge Gaspers, and Toby Walsh. 2015. Online fair division: analysing a food bank problem. In *Proceedings of the 24th International Conference on Artificial Intelligence*. 2540–2546.

[6] Dimitris Bertsimas and Nathan Kallus. 2020. From predictive to prescriptive analytics. *Management Science* 66, 3 (2020), 1025–1044.

[7] Erik Brynjolfsson, Yu Hu, and Duncan Simester. 2011. Goodbye pareto principle, hello long tail: The effect of search costs on the concentration of product sales. *Management Science* 57, 8 (2011), 1373–1386.

[8] T. Chen, Linpeng Tang, Q. Liu, Diyi Yang, Saining Xie, Xuezhi Cao, C. Wu, E. Yao, Zhengyang Liu, Z. Jiang, C. Chen, Weihao Kong, and Yingrui Yu. 2012. Combining Factorization Model and Additive Forest for Collaborative Followee Recommendation.

[9] Tianqi Chen, Weinan Zhang, Qiuxia Lu, Kailong Chen, Zhao Zheng, and Yong Yu. 2012. SVDFeature: a toolkit for feature-based collaborative filtering. *The Journal of Machine Learning Research* 13, 1 (2012), 3619–3622.

[10] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 815–824.

[11] Alisha Coleman-Jensen, Matthew P Rabbitt, Christian A Gregory, and Anita Singh. 2020. Household Food Security in the United States in 2019. *USDA-ERS Economic Research Report* (2020).

[12] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*. 278–288.

[13] Adam N Elmachtoub and Paul Grigas. 2017. Smart" predict, then optimize". *arXiv preprint arXiv:1710.08005* (2017).

[14] Adrienne Porter Felt, Serge Egelman, and David Wagner. 2012. I've got 99 problems, but vibration ain't one: a survey of smartphone users' concerns. In *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*. ACM, 33–44.

[15] Daniel Fleder and Kartik Hosanagar. 2009. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management science* 55, 5 (2009), 697–712.

[16] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. 2010. Learning attribute-to-feature mappings for cold-start recommendations. In *2010 IEEE International Conference on Data Mining*. IEEE, 176–185.

[17] Gagan Goel and Aranyak Mehta. 2008. Online budgeted matching in random input models with applications to Adwords.. In *SODA*, Vol. 8. 982–991.

[18] Mark P Graus and Martijn C Willemsen. 2015. Improving the user experience during cold start through choice-based preference elicitation. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 273–276.

[19] Dana Gunders and Jonathan Bloom. 2017. Wasted: How America is losing up to 40 percent of its food from farm to fork to landfill. (2017).

[20] Canan Gunes, Willem-Jan van Hoeve, and Sridhar Tayur. 2010. Vehicle routing for food rescue programs: A comparison of different approaches. In *CPAIOR*.

[21] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[22] David Laborde, Will Martin, Johan Swinnen, and Rob Vos. 2020. COVID-19 risks to global food security. *Science* 369, 6503 (2020), 500–502.

[23] Kuang-Chih Lee, Ali Jalali, and Ali Dasdan. 2013. Real time bid optimization with smooth budget delivery in online advertising. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*. 1–9.

[24] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, and Ariel D. Procaccia. 2019. WeBuildAI: Participatory Framework for Algorithmic Governance. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 181 (Nov. 2019), 35 pages. https://doi.org/10.1145/3359283

[25] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. 2016. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 539–548.

[26] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*. 689–698.

[27] Taylor Lundy, Alexander Wei, Hu Fu, Scott Duke Kominers, and Kevin Leyton-Brown. 2019. Allocation for social good: auditing mechanisms for utility maximization. In *ACM EC*.

[28] Vahideh Manshadi and Scott Rodilitz. 2020. Online Policies for Efficient Volunteer Crowdsourcing. *arXiv preprint arXiv:2002.08474* (2020).

[29] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. 2007. Adwords and generalized online matching. *Journal of the ACM (JACM)* 54, 5 (2007), 22–es.

[30] Ibrahim Muter and Tevfik Aytekin. 2017. Incorporating aggregate diversity in recommender systems using scalable optimization approaches. *INFORMS Journal on Computing* 29, 3 (2017), 405–421.

[31] Divya Jayakumar Nair, Hanna Grzybowska, David Rey, and Vinayak Dixit. 2016. Food rescue and delivery: Heuristic algorithm for periodic unpaired pickup and delivery vehicle routing problem. *Transportation Research Record* 2548, 1 (2016), 81–89.

[32] Divya J Nair, Taha Hossein Rashidi, and Vinayak V Dixit. 2017. Estimating surplus food supply for food rescue and delivery operations. (2017).

[33] Katja Niemann and Martin Wolpers. 2013. A new collaborative filtering approach for increasing the aggregate diversity of recommender systems. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 955–963.

[34] Caleb Phillips, Rhonda Hoenigman, and Becky Higbee. 2011. Food redistribution as optimization. *arXiv preprint arXiv:1108.5768* (2011).

[35] Canice Prendergast. 2016. The Allocation of Food to Food Banks. *EAI Endorsed Trans. Serious Games* 3, 10 (2016), e4.

[36] Neil Rubens, Mehdi Elahi, Masashi Sugiyama, and Dain Kaplan. 2015. Active learning in recommender systems. In *Recommender systems handbook*. Springer, 809–846.

[37] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. 253–260.

[38] Zheyuan Ryan Shi, Yiwen Yuan, Kimberly Lo, Leah Lizarondo, and Fei Fang. 2020. Improving Efficiency of Volunteer-Based Food Rescue Operations. In *AAAI*. 13369–13375.

[39] Liang Tang, Yexi Jiang, Lei Li, and Tao Li. 2014. Ensemble contextual bandits for personalized recommendation. In *Proceedings of the 8th ACM Conference on Recommender Systems*. 73–80.

[40] Kentaro Toyama. 2015. *Geek heresy: Rescuing social change from the cult of technology*. PublicAffairs.

[41] Megan D Wolfson and Catherine Greeno. 2018. Savoring surplus: effects of food rescue on recipients. *Journal of Hunger & Environmental Nutrition* (2018).

[42] Jian Xu, Kuang-chih Lee, Wentong Li, Hang Qi, and Quan Lu. 2015. Smart pacing for effective online ad campaign optimization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2217–2226.

[43] Mi Zhang and Neil Hurley. 2008. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*. 123–130.

[44] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive collaborative filtering. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 1411–1420.

[45] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*. 22–32.