February 22, 2019

Location, time, and preference-aware multi-user scheduling

William Li

Sophia Yang

Location, time, and preference-aware multi-user scheduling

ABSTRACT

This disclosure describes techniques to determine an optimal point-of-interest for multiple users to visit together. The optimality criteria include various user preferences, schedules, and routines, that are obtained and used with user permission. A point of interest can include a location of entertainment, business or leisure, e.g., a restaurant, a movie, an event, or other outing.

KEYWORDS

- Location recommendation
- Group schedule
- Group meeting
- Location-aware scheduling
- Preference-aware scheduling
- Multi-user scheduling
- Point of interest

BACKGROUND

A group of individuals with varied interests and schedules may want to meet at a mutually convenient time and location, e.g., at a restaurant, a movie theater, an event, etc. It is of interest to automatically determine a suitable location based on user schedules and preferences of the multiple individuals. For example, if the group wants to meet at a restaurant, an optimal time and location is one that accounts for the cuisine preferences, food allergies, the restaurant preferences, preferred time of eating, etc., of individual members of the group.

DESCRIPTION

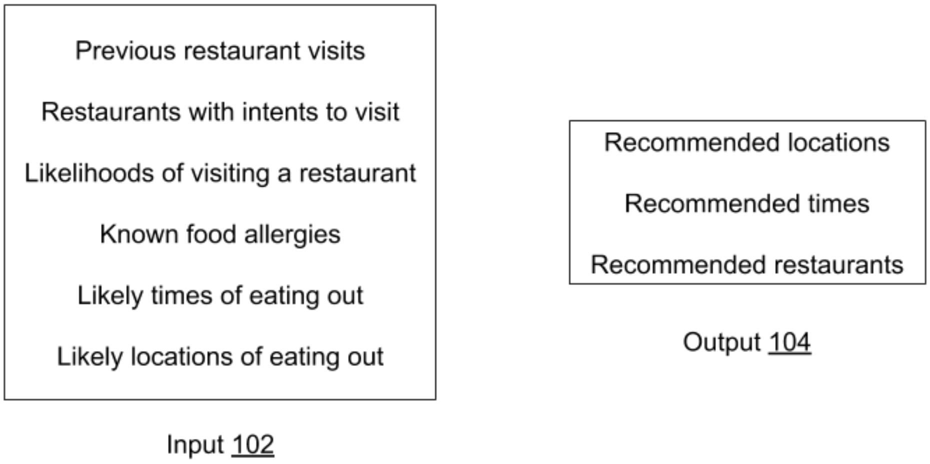

Previous restaurant visits
Restaurants with intents to visit
Likelihoods of visiting a restaurant
Known food allergies
Likely times of eating out
Likely locations of eating out
Input 102
Recommended locations
Recommended times
Recommended restaurants
Output 104

Fig. 1: Examples of input and output for restaurant as an illustrative point of interest

Using a restaurant as an illustrative point of interest (Fig. 1), the techniques of this disclosure obtain, with user permission, one or more of the following as inputs (102):

- a user's previous restaurant visits, and dates and histograms thereof, derived, e.g., from a semantically annotated location history from mobile devices, when the users permit use of such information;
- a user’s list of restaurants with intent-to-visit, inferred, e.g., from map-application bookmarks, lists in note-keeping applications, mentions in emails, chats, etc., if the user permits use of such information;
- a user's likelihood to visit a given restaurant within a given geographic area, based on, e.g., a score assigned by the user to the restaurant at a restaurant-review website or application, when the user permits use of such information;
- a user’s known food allergies or food preferences, based on, e.g., information available to a virtual assistant, when the user permits use of such information;

- a user’s likely times of eating out, based on, e.g., location history, commute routines, calendar information, eating routines, etc., when the user permits use of such information;
- a user’s likely locations for eating out, based on, e.g., location history, calendar information, etc., when the user permits use of such information; etc.

One or more of the aforementioned input values are obtained for respective members of the group of users, based on permissions specified by individual users. An output list of top-*N* recommended times, locations and restaurant recommendations (104) is determined that includes restaurants that are likely most acceptable to the group members.

The inputs are used variously to determine the outputs. For example, if a user indicates that they’d prefer variety, the list and histogram of recently-visited restaurants are used to suggest a restaurant that the user has not visited recently. On the contrary, if a user prefers familiarity of location, the list and histogram of recently-visited restaurants are used to suggest a restaurant that the user has visited recently.

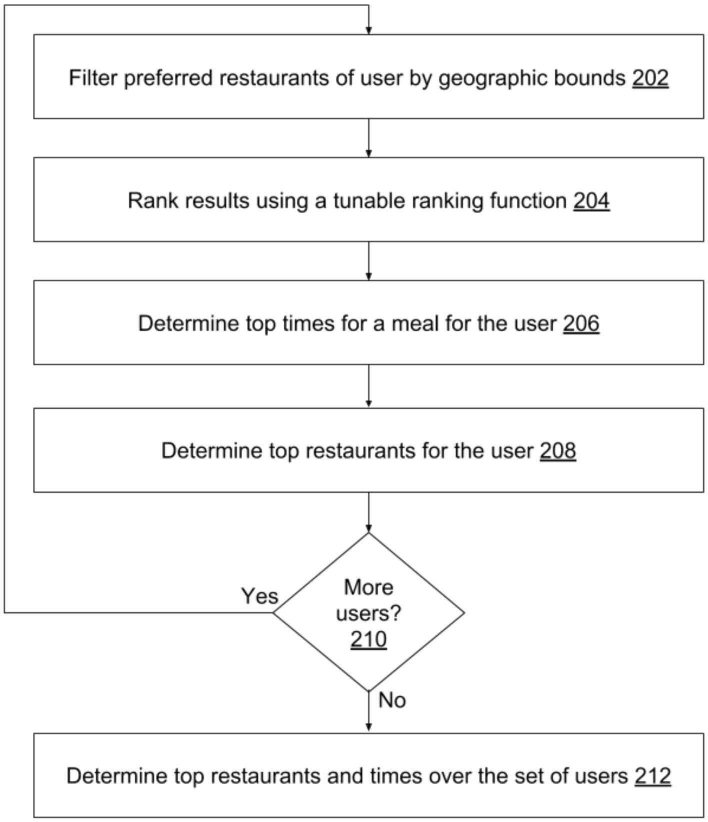
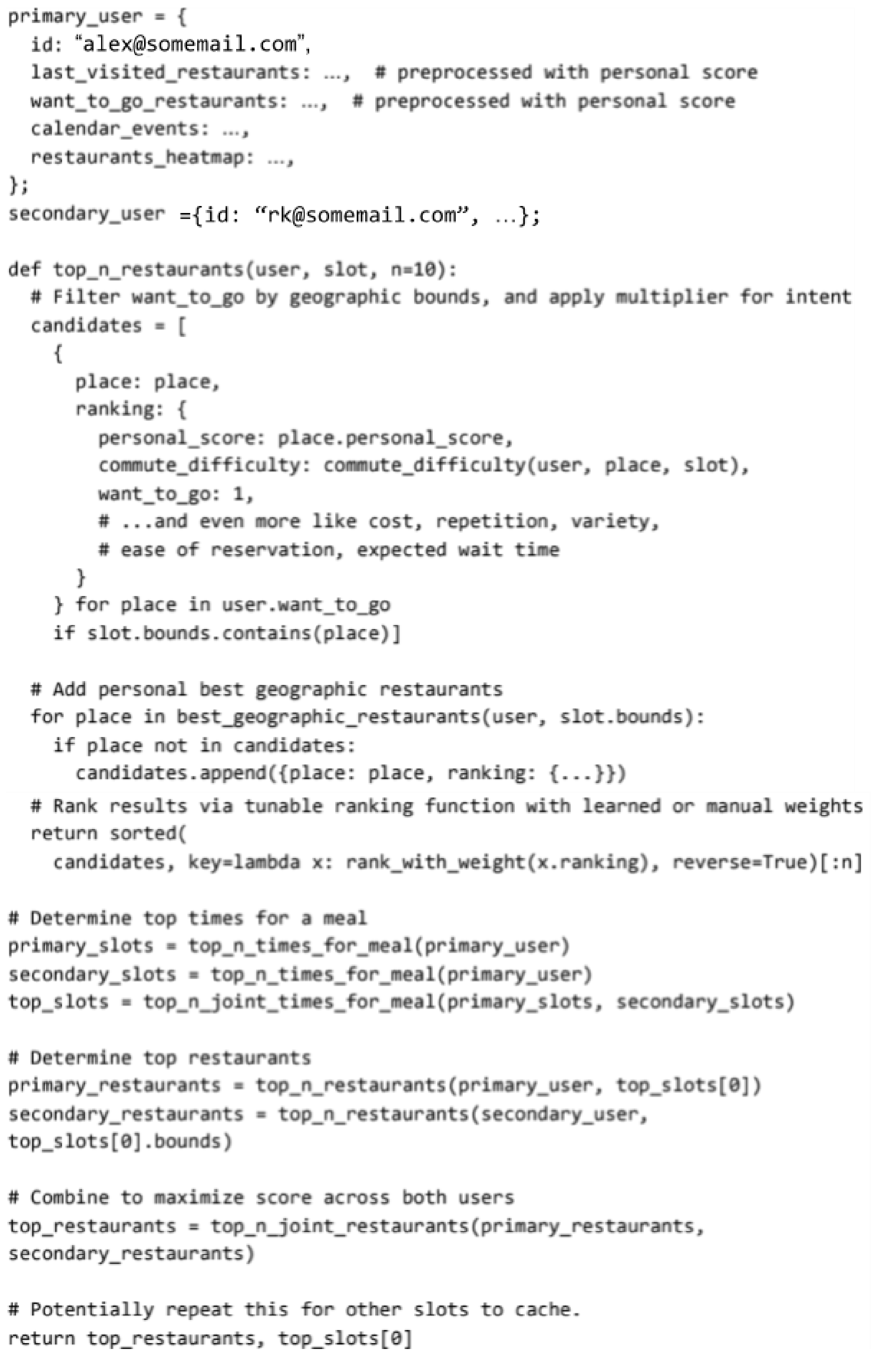

Filter preferred restaurants of user by geographic bounds 202
Rank results using a tunable ranking function 204
Determine top times for a meal for the user 206
Determine top restaurants for the user 208
More users? 210
Yes
No
Determine top restaurants and times over the set of users 212

Fig. 2: Determining an optimal time, location, and restaurant for a group of users

Fig. 2 illustrates an example process to determine an optimal time, location, and restaurant for a group of users, per techniques of this disclosure. The preferred restaurants of a user are filtered based on geographic bounds (202), with multipliers applied to account for user intent. The filtered results are ranked using a tunable ranking function (204) that has machine-learned or manually-set weights. Parameters that contribute to the rank of a restaurant can include, e.g., the personal score given by the user to the restaurant; the personal best score within a geography; the difficulty of commuting to the restaurant; cost of food and beverages at the restaurant; repetition or variety of the menu; the ease of reservation; the expected wait time; etc. The ranking parameters are used with user permission. The top times for a meal for the user are determined (206). The top restaurants for the user are determined (208). A similar process is repeated for all users within the group (210). The optimal times and restaurant choices are combined over users to determine a list of top-*N* restaurants and times that are suitable for the group of users to meet (212). The process can be implemented as part of a map application, a virtual assistant, a calendar application, etc.



```
primary_user = {
  id: "alex@somemail.com",
  last_visited_restaurants: …,  # preprocessed with personal score
  want_to_go_restaurants: …,  # preprocessed with personal score
  calendar_events: …,
  restaurants_heatmap: …,
};
secondary_user ={id: "rk@somemail.com", …};

def top_n_restaurants(user, slot, n=10):
  # Filter want_to_go by geographic bounds, and apply multiplier for intent
  candidates = [
    {
      place: place,
      ranking: {
        personal_score: place.personal_score,
        commute_difficulty: commute_difficulty(user, place, slot),
        want_to_go: 1,
        # ...and even more like cost, repetition, variety,
        # ease of reservation, expected wait time
      }
    } for place in user.want_to_go
    if slot.bounds.contains(place)]

  # Add personal best geographic restaurants
  for place in best_geographic_restaurants(user, slot.bounds):
    if place not in candidates:
      candidates.append({place: place, ranking: {...}})
  # Rank results via tunable ranking function with learned or manual weights
  return sorted(
    candidates, key=lambda x: rank_with_weight(x.ranking), reverse=True)[:n]

# Determine top times for a meal
primary_slots = top_n_times_for_meal(primary_user)
secondary_slots = top_n_times_for_meal(primary_user)
top_slots = top_n_joint_times_for_meal(primary_slots, secondary_slots)

# Determine top restaurants
primary_restaurants = top_n_restaurants(primary_user, top_slots[0])
secondary_restaurants = top_n_restaurants(secondary_user,
top_slots[0].bounds)

# Combine to maximize score across both users
top_restaurants = top_n_joint_restaurants(primary_restaurants,
secondary_restaurants)

# Potentially repeat this for other slots to cache.
return top_restaurants, top_slots[0]
```

Fig. 3: Pseudocode to determine optimal time, location, and restaurant for two users

Fig. 3 illustrates pseudocode to determine optimal time, location, and restaurant per the techniques of this disclosure, when the number of users is two.

In this manner, the techniques combine a user’s location history with knowledge of the user’s immediate future plans or intentions to make a place recommendation. Compared to a naive approach of recommending a restaurant (or other business), e.g., via a 5-star rating system, the advantages of the techniques include the following:

- Cuisine and food preferences of participants are taken into consideration in making a recommendation.
- A mutually acceptable time is found based on the predicted calendars of participants.
- A mutually acceptable location for the meal is found based on predicted commute times of participants.
- A participant does not need to manually maintain a list of places to visit, or manually enter their work or home schedule.

Alternatively, the user can specify time, location, or restaurant preferences. Factors fewer in number than the aforementioned set, whether inferred or not, or additional factors can be utilized as permitted by the users to make mutually acceptable recommendations.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user’s social network, social actions or activities, profession, a user’s preferences, or a user’s current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user’s identity may be treated so that no personally identifiable information can be determined for the user, or a user’s geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

This disclosure describes techniques to determine an optimal point-of-interest for multiple users to visit together. The optimality criteria include various user preferences, schedules, and routines, that are obtained and used with user permission. A point of interest can include a location of entertainment, business or leisure, e.g., a restaurant, a movie, an event, or other outing.