Custom recommender: Educational Events

Context

The goal is to recommend user educational events around him: public lectures, courses, trainings, such that you may find on SkillShare, EventBrite, Meetup or T&P.

Each event is a unique instance, therefore collaborative filtering if of limited use here (we can recommend only events from the past based on CF). This resembles the News use case, but Event becomes obsolete quite soon after the recommendation is made, therefore content-based filtering is the only applicable approach here.

Content attributes

It seems reasonable to use the following attributes:

1. **Topic.** Usually event services have a faceted search in the interface, so for the event we can easily use one or several topics this event belongs to.
2. **Type.** Some people have preferences to particular types of events: meetups, public lectures, roundtables, one-time seminars, course series.
3. **Tags.** In case the service provides some folksonomy around the events, this might be useful.
4. **Lecturer / Master (person).** The important part of the event is a personal communication with the person who run the event. Let's use it.
5. **Organizer (institution).** Organizer for events has equal importance as musical label for music: they provide certain quality assurance and editorial.
6. **Venue.** Event audience often is bound to some cozy venue, thus events that happen in the same venue have something in common.
7. **Location.** Current location of the user matters a lot – most of the people just not interested in events that happen on the other side of the country.
8. **Day of the week.** Each person has its own life schedule that defines appropriate event to a great extent.
9. **Time.** Same as the previous one.
10. **Description / Annotation.** Mining of textual description and links may provide lots of information about the event, especially if this is the only difference between two events.

Most of the content features can be easily represented as triples (event, feature, value). For features 1-6 it's straightforward. For location we should take major cities in, say, N mile radius from the venue. For day of the week and time we should represent them in chunks (day – days or just weekend/midweek, time – hourly chunks or just morning/day/evening). For textual annotations we should extract terms as it is usually done in the Information Retrieval.

After the triples are constructed, TFIDF should be applied to each feature set separately, resulting in 10 profiles for each item. This is necessary to later tune weight of each of the features in user-item similarity.

Another approach may be to use LSA/SVD on all features altogether, but that may not work as linear dimensionality reduction doesn't work that well for features of different modalities.

User profiles

For user profiles we are going to use information on events that user attended in the past. User profile will have 10 components (as the item profiles) and is calculated as a direct sum of events attended by user. We discard information on if the user liked the event, but for a lot of features (e.g. location/data) it is important if he was able to attend the event.

For sure, this information is not available for the new users ('cold start' problem). For such users we can create dummy profiles by aggregating profiles of the existing users from the same location and similar system usage pattern (e.g. accessed the system in the same time of the day). Also some similar users can be mined from the connected social profiles (e.g. average profile of friends).

Context-aware preferences

The context includes location, day of the week and time. They present in the features described above, their significance is represented with their weight in the distance function (e.g. cosine) and can be tuned either individually or globally via linear regression.

User feedback

To achieve proper user feedback, we need (1) good explanation (so user understands why is he getting this recommendation) and (2) tools to respond the feedback.

As with all content-based methods, it's easy to explain the recommendations providing aggregated information about most discriminative features of the recommended event and referring to the events in the past that cause user profile to be similar to the recommended event.

Hence, it is reasonable to provide two tools for user feedback: (1) negative reaction on the recommended item (e.g. 'skip' or 'close'), which provides input for tuning the algorithm (e.g. weights for different groups of features) and (2) ability to edit the history (e.g. by removing some visited, but not really liked events that lead to improper recommendations).