Recommender system in Event Based Social Networks

Marco Pietro Abrate

Supervisor: Amin Kaboli

Project type: semester project (12 credits)

21/10/2019

Contents

1 Event-Based Social Network (EBSN)

An EBSN does not only contain online social interactions as in other conventional social networks, but also includes valuable offline social interactions captured in offline activities. Communities in EBSNs are more cohesive than those in other type of social networks. In the context of information flow, an interesting problem is event recommendation. [2]

2 Context-aware recommendation in EBSN

The sheer volume of events available in EBSN often undermines the users' ability to choose the event that best fit their interests. Recommender systems (RS) appear as a natural solution for this problem. Differently from classic recommendation scenarios, the event recommendation problem is intrinsically cold-start. Indeed, events published in EBSNs are typically short-lived and, by definition, are always in the future, having little or no trace of historical attendance. To overcome this problem, several contextual signals from the EBSN are exploited: répondez s'il vous plaît (RSVP), group membership, geographical location, time preference. The assumption is that these signals have a positive influence on the users' decision about attending an event.

3 Problem setup

The event recommendation problem can be stated as follow: given a target user and a set of contextual signals, which of the available events are more likely to be attended (i.e. RSVP'ed "yes") by this user? This scenario is simulated by splitting the available RSVP data into two partitions, before and after timestamp θ. The (user, events) pairs occurring before θ form the training partition, while those that occur after θ form the test set. The task is to find a scoring function of the form:

$$\hat{s} : U \times E \times T \times G \times C \times D \to \mathbb{R} \tag{1}$$

where U is the set of users, E is the set of events, T are the users' temporal preferences, G is the set of groups, C is the set of textual content of events and D are the users' geographic distance preferences. Thus, for a given user $u \in U$ and the contextual signals $t \in T$, $g \in G$, $c \in C$, $d \in D$, the top-n recommendations can be computed by:

$$\text{top-}n(u, t, g, c, d) := \underset{e \in E \setminus E_u}{\arg\overset{n}{\max}}\, \hat{s}(u, e, t, g, c, d) \tag{2}$$

where n denotes the number of events to be recommended and E_u is the set of events that user u has attended in the past. In top-n recommendation settings, such as this, the goal is to minimize a ranking loss function of the form:

$$l : P(E) \times \mathbb{R}^E \to \mathbb{R} \tag{3}$$

which quantifies the misfit between the recommended list (generated by $\hat{s} \in \mathbb{R}^E$) and the actual list (a subset of $P(E)$) on the test users whose actual events are unknown during training.

4 Contextual models

4.1 Content-aware

Content-based filtering appears as a promising approach. In particular, besides its usual capacity of retrieving events with similar descriptions, it also captures recurrent events since these tend to have similar or nearly identical content. Each user is represented as a TF-IDF vector of the words extracted from the past events the user attended. Moreover, each event is weighted by its time distance to the recommendation instant by means of a time value of money decay function [3]. Formally, the profile of user $u \in U$ is defined as:

$$\overrightarrow{u} := \sum_{e \in E_u} \frac{1}{(1+\alpha)^{\tau(e)}} \times \overrightarrow{e} \tag{4}$$

where $\overrightarrow{e}$ is the TF-IDF representation of event e, α is a time decay factor and $\tau(\alpha)$ return the number of days from the RSVP to e until the moment of recommendation. Finally, the candidate events are ranked based on their cosine similarity with respect to the target user:

$$\hat{s}_C(u,e) = cos(\overrightarrow{u}, \overrightarrow{e}) \tag{5}$$

4.2 Social-aware

In some EBSNs, such as Meetup, users may join groups and interact with other users both online and face-to-face. In our case, groups will be created automatically based on the real life connections of the user. An other approach would be to use a density based clustering algorithm to cluster users in groups based on their past attendance to events. In any case, the best way to find groups can only be corroborated by doing experiments and trying out different techniques. After group membership of each user has been found, two different models can be implemented to recommend events: group frequency and multi-relational model.

Group frequency model: The intuition of this method is that the likelihood of the target user $u \in U$ attending event $e \in E$ depends on the number of events this user attended with the users that are part of the group that e belongs to. In other words, the more events a user attends in a group, the higher the probability that this user will continue to attend events with the members of this group. Formally, assuming that g_e represents the group associated with the candidate event e, the relevance score of this event for the target user $u \in U$ is:

$$\hat{s}_{S_1}(u,e) := \frac{|E_{u,g_e}|}{|E_u|} \tag{6}$$

where E_u denotes the set of all events that user $u \in U$ attended and E_{u,g_e} denotes the set of events attended by u that were created by group $g_e \in G$.

Multi-relational model: With this method, which is more complex than the previous one, one might discover that users affiliated to the same or similar groups are prone to attend the same events created by these groups. The user-group relation is especially important because it enables the recommendation of cold-start events (with zero RSVP). This method is based on the Multi-Relational Factorization with Bayesian Personalized Ranking (MRBPR) [1]. the idea is to reconstruct the target relation $R_{UE} \subseteq U \times E$ by considering a joint factorization approach that includes the auxiliary relations $R_{UG} \subseteq U \times G$ and $R_{GE} \subseteq G \times E$. The parameters of the model $\Theta = \{U, E, G\}$ are the latent matrices associated with each considered entity. The objective function is then expressed as follows:

$$\begin{aligned}\arg\min_{\Theta} \quad & \alpha L(R_{UE}, \mathbf{U}\ \mathbf{E}^T) && (7)\\ & + \beta L(R_{UG}, \mathbf{U}\ \mathbf{G}^T) && (8)\\ & + \gamma L(R_{GE}, \mathbf{G}\ \mathbf{E}^T) && (9)\\ & + \lambda_U ||\mathbf{U}||^2 + \lambda_E ||\mathbf{E}||^2 + \lambda_G ||\mathbf{G}||^2 && (10)\end{aligned}$$

where L is a loss function measuring the reconstruction error of the relation under construction, α, β, γ are weights for the losses and λ_U, λ_G, λ_E are regularization parameters. Once the model is learned, the relevance score of an event $e \in E$ for the target user $u \in U$ is:

$$\hat{s}_{S_2}(u, e) = \sum_{f=1}^{k} \overrightarrow{u}_f \overrightarrow{e}_f \tag{11}$$

where $\overrightarrow{u}$ and $\overrightarrow{e}$ are the latent vectors of user u and event e, respectively.

4.3 Location-aware

The mobility pattern of users within a city may vary from user to user. The idea is to use a kernel-based density estimation approach to model the mobility patterns of individual users as distributions of geographic distances between the attended events. The relevance of a new event for a user is based on the likelihood of this event being located in any of the regions the user attended events in the past. Formally, let L_u be the sample of latitude-longitude (lat-long) coordinates of the events the target user has attended in the past:

$$L_u := \bigcup_{e \in E_u} l_e \tag{12}$$

where l_e is the lat-long coordinate of event $e \in E$. A kernel density function $\hat{f}$ is defined over L_u as follows:

$$\hat{f}(l) := \frac{1}{|L_u|} \sum_{l' \in L_u} K_{\mathbf{H}}(l - l') \tag{13}$$

where l is a lat-long coordinate, $K_{\mathbf{H}}(.)$ is the bivariate Gaussian kernel,

$$K_{\mathbf{H}}(\mathbf{x}) := \frac{1}{\sqrt{2\pi|\mathbf{H}|}} \epsilon^{-\frac{\mathbf{x}\mathbf{x}^T}{2\sqrt{\mathbf{H}}}} \tag{14}$$

The geographical preferences of a given user are then represented by the sum of all Gaussian distributions centered at l_e, for all $e \in E_u$:

$$\hat{s}_D(u, e) := \hat{f}(l_e) \tag{15}$$

4.4 Time-aware

Another important factor is when the event occurs. This information is captured by assuming that users that attended events in the past at certain days of the week and at certain hours of the day will likely attend events with a similar temporal profile in the future. Formally, each event e is represented as a 24×7-dimensional vector $\overrightarrow{e}$ in the space of all possible days of the week and hours of the day, with an entry set to 1 whenever the event happened at that particular day and hour. Each user is represented as:

$$\overrightarrow{u} := \frac{1}{|E_u|} \sum_{e \in E_u} \overrightarrow{e} \tag{16}$$

The temporal score for user u and event e can now be expressed as the cosine between their vector representation:

$$\hat{s}_T(u, e) := cos(\overrightarrow{u}, \overrightarrow{e}) \tag{17}$$

5 Rank events with contextual features

The aforementioned recommenders are used as features for learning to rank events. In particular, let $D := \{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, y_n)\}$ be the training set where $\mathbf{x}_i := \{s_1(u, e), ..., s_m(u, e), |U_e|\}$ is a feature vector containing the normalized scores generated for a given pair of user $u \in U$ and event $e \in E$ by each of the m context-aware recommenders, plus the number of RSVPs $|U_e|$ the event e has received in the past. Also let $y_i = \{0, 1\}$ denote whether user u attended event e (1) or not (0). The goal is to learn a function $h(\mathbf{x})$ such that for any pair of training instances $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$ the following implication holds:

$$h(\mathbf{x}_i) > h(\mathbf{x}_j) \leftrightarrow y_i > y_j \tag{18}$$

References

[1] Artus Krohn-Grimberghe et al. “Multi-relational matrix factorization using bayesian personalized ranking for social network data”. In: *Proceedings of the fifth ACM international conference on Web search and data mining.* ACM. 2012, pp. 173–182.

[2] Xingjie Liu et al. “Event-based social networks: linking the online and offline social worlds”. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM. 2012, pp. 1032–1040.

[3] Thomas Sandholm and Hang Ung. “Real-time, location-aware collaborative filtering of web content”. In: *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation.* ACM. 2011, pp. 14–18.