

User Preferences Initialization and Integration in Critique-Based Mobile Recommender Systems

Quang Nhat Nguyen and Francesco Ricci

eCommerce and Tourism Research Laboratory
ITC-irst, Italy
<http://ectrl.itc.it>



- Problems and motivations
- User preferences representation
- The recommendation process
- The user preferences representation initialization
 - Space-time constraints extraction
 - History-based initialization
 - User specification-based initialization
- Discussion on the applicability of the proposed methodology
- Conclusion and future work

- Why are Recommender Systems (RS) necessary for e-commerce web sites?
 - The information overload problem
 - Users really need system support in **filtering out** irrelevant products, in **comparing** options to select the best one(s)
- In RS, the system's initial guess of the user's preferences affects:
 - ⇒ The quality of the first recommendation results
 - ⇒ The success of recommendation sessions
- In mobile RSs, building the initial representation of the user's preferences is challenging, because:
 - The trade-off between the **accuracy** of the representation and the user **cognitive effort** in making it explicit
 - The constraints of the mobile environment, and the behavioral characteristics of mobile users

- Persistent (i.e., long-term) user model
 - Initialized at registration time (typically by asking the user)
 - Periodically revised by explicitly or implicitly eliciting the user's preferences.
 - Disadvantages:
 - potentially mismatch between the current and past preferences
 - the system must collect a relatively large amount of user data
 - mining past usage data requires lot of computation resource and time
- Context-dependent user model
 - The space-time information of the user
 - The user's context-dependent preferences
 - Disadvantages:
 - the user needs to have enough knowledge about the problem domain
 - the user is willing to express his preferences
- Our proposed approach integrates both of them into the User Model.

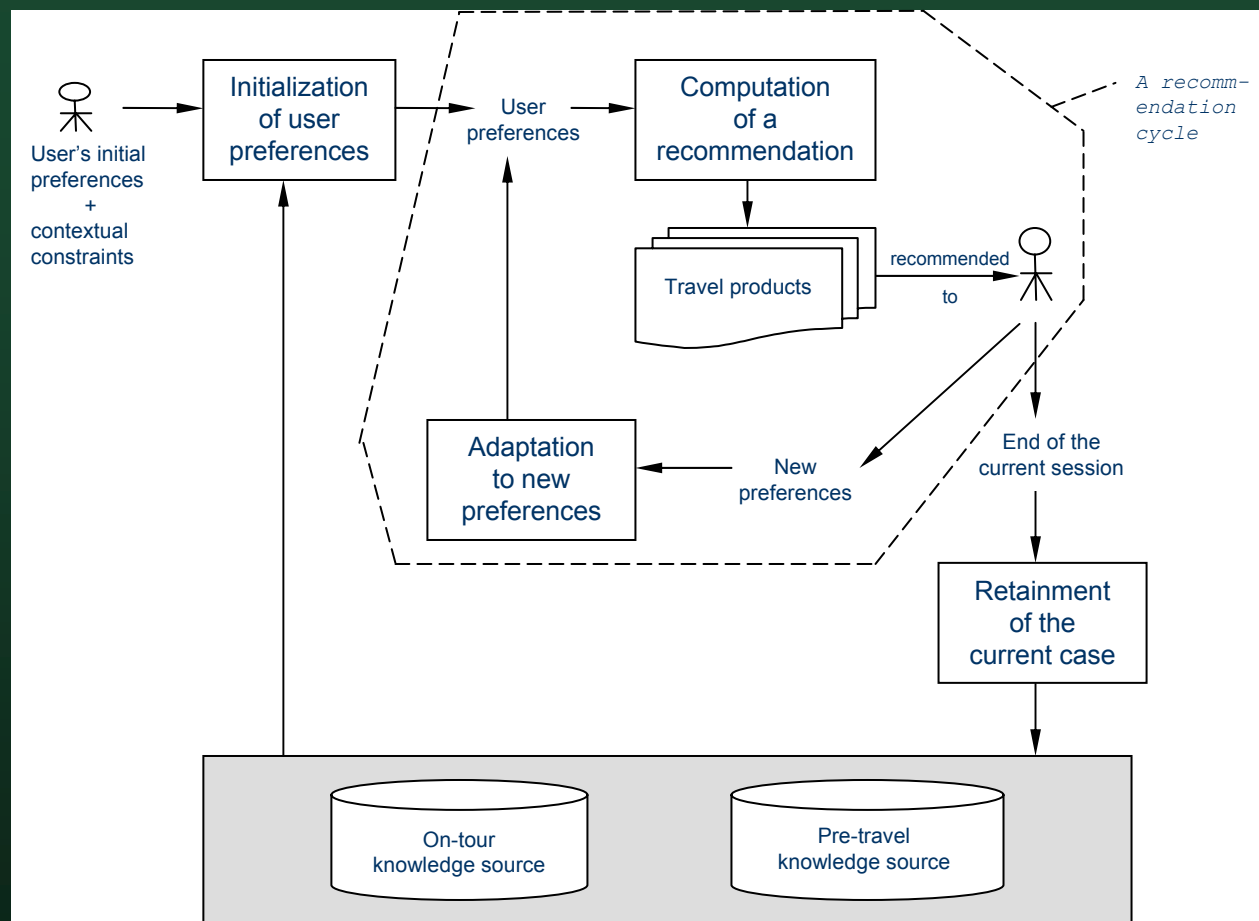
- Example: recommending travel products to travelers who
 - are at the selected destination (or on the move to), and
 - use a mobile device to interact with the system
- Travel products representation
 - A travel product is represented as a vector of feature values:
 $x = (x_1, \dots, x_n)$
 - A feature value (x_i) can be real, finite integer, nominal (symbolic), symbol-set, or text
 - E.g., $x = (\text{"LaBerta"}, \{\text{pizzeria}\}, \text{"Trento"}, 20, \{7, 1\}, \{\text{parking}, \text{smoking_room}\})$
- User's preferences are represented as a composite query that include:
 - a logical query,
 - a favorite pattern, and
 - a feature importance weights vector.

- The logical query (Q_L)
 - models conditions that must be satisfied by the recommended products
 - a conjunction of logical constraints: $Q_L = (c_1 \wedge \dots \wedge c_m)$
 - a constraint (c_j) involves only one feature; and a feature may appear in only one constraint
 - E.g., $Q_L = (x_3 = \text{"Trento"}) \wedge (x_5 \supseteq \{7,1\})$
- The favorite pattern (p)
 - models wish conditions that the recommended products should satisfy (i.e., the more the preferred): $p = (p_1, \dots, p_n)$
 - wish conditions allow trade-offs to be made
 - $p_i = ?$ means that the favorite value of i -th feature is unknown
 - E.g., $p = (?, \{\text{spaghetteria}\}, ?, ?, ?, ?)$
- The feature importance weights (w)
 - model how much each feature is important w.r.t. the others
 - a weight value (w_i) is a real number in $[0,1]$
 - E.g., $w = (0, 0.6, 0, 0.4, 0, 0)$

- On-tour support (i.e., recommendations) is provided by MobyRek
- MobyRek co-operates with NutKing (a web-base RS which supports building pre-travel plans)
- The co-operation with NutKing allows pre-travel information to be exploited when providing on-tour support
- An on-tour recommendation session
 - starts when an on-the-move traveler requests MobyRek to find some interesting travel product,
 - ends when the traveler either selects a product or quits the session,
 - evolves in cycles (in each cycle, a set of recommended products are shown to the user).
- In our model, the recommendation process consists of four logical components:

Initialization, Interaction, Adaptation, and Retainment.

The recommendation process



- Q_L is executed
- Q_L -matching products are ranked to their similarity to $\langle p, w \rangle$
- A cut-off: only the k best products are included in the recommendation result.

The recommendation process

- At a cycle, regarding the recommended products, the user can
 - **selects** one \Rightarrow a **successful** session;
 - **criticizes** one, and specifies his preference on the unsatisfactory features (e.g., "I want restaurants cheaper than this one") \Rightarrow proceeds to the **next cycle**
 - user critiques help MobyRek to adapt the previous representation
 \Rightarrow to compute a new recommendation set
 - the adaptation depends on the type of the critique expressed and on the type of the feature criticized
 - **quits** the current session \Rightarrow a **failure**.

- Storing on-tour recommendation cases

$$CB^{\text{on-tour}} = pT \times Q^0 \times SoC \times \text{Acceptance}$$

pT : a link to the pre-travel plan (if existing)

Q^0 : the system's initial representation of the user's preferences

SoC : the sequence of critiques in a recommendation

Acceptance: the final recommendation result.

Feedback on Characteristics

This restaurant's own characteristics

Parking

Admitted animals

But you would like also

Meeting room

Hotel

Music

Air conditioned

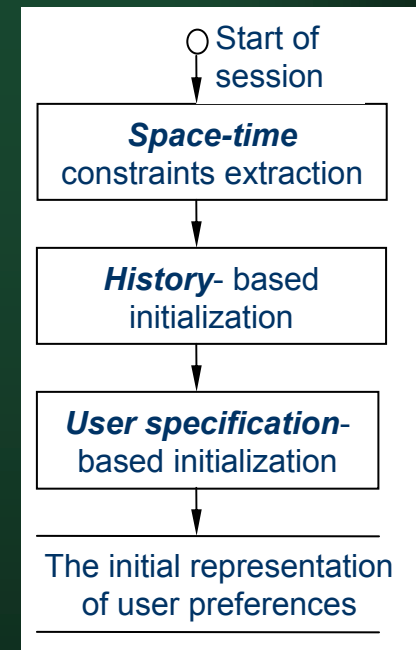
Smoking room

Back + Update search

The user preferences representation initialization

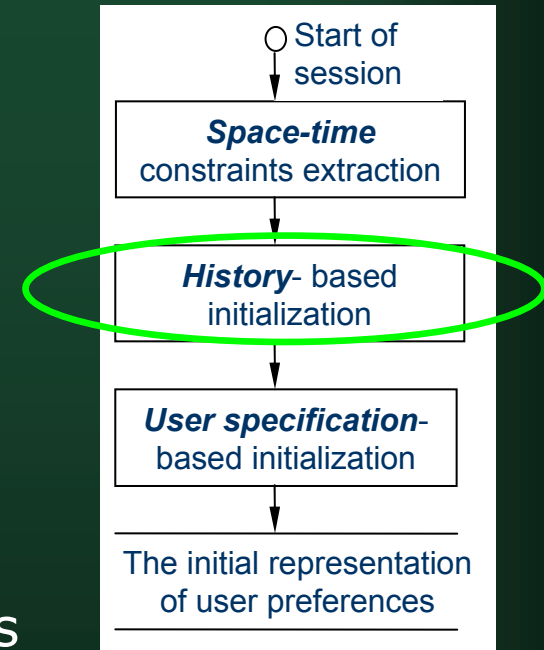
The initialization of the user preferences representation

- exploits
 - the space-time information
 - the user's pre-travel plan (of the current trip) and the pre-travel plans of other similar users
 - on-tour recommendation sessions in the past
 - the preferences explicitly specified by the user
- consists of
 - the 1st phase: to extract the **space-time** constraints
 - the 2nd phase: to build the **history**-based representation
 - the 3rd phase: to overwrite the history-based representation by the initial **preferences explicitly specified** by the user.



History-based initialization

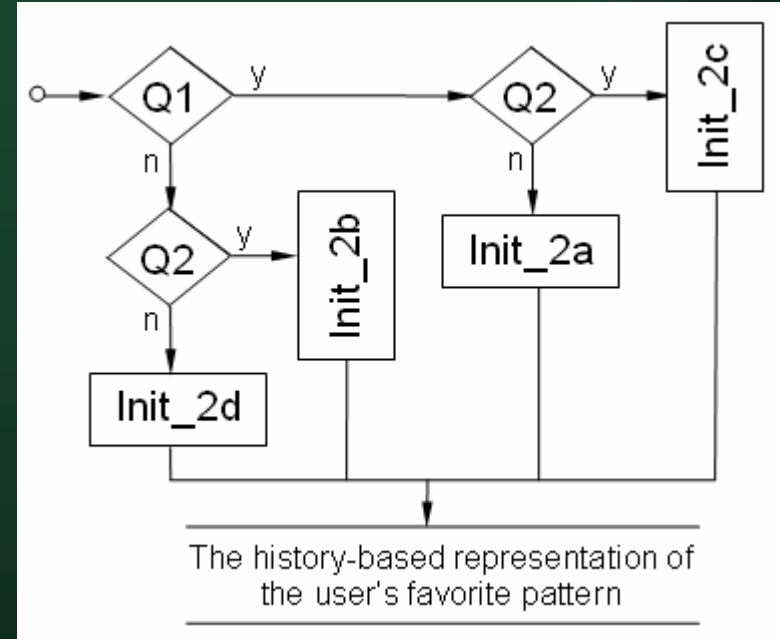
- The history-based initialization is done automatically by the system, without consulting the user
- History-based extracted preferences are considered as wish conditions, because these (old) preferences
 - are extracted from a similar session, and
 - may be incorrect regarding the current session
- Hence, history-based extracted preferences are encoded in $\langle p, w \rangle$.



- The favorite pattern initialization

There are four possible cases corresponding to

- [Q1] whether or not the user *has already built a pre-travel plan* (for the current trip), and
- [Q2] whether or not the user *has ever accepted an on-tour recommendation* (of the same product type)



- The feature importance weights initialization

- By exploiting the history of the user's interactions with MobyRek
- The more a feature has been criticized by the user the more that feature is important to him

The importance weight of feature f_i , given session s_k of user u_j

$$w_i(u_j, s_k) = \frac{1}{\lambda_k} \cdot \sum_{l=1}^{\lambda_k} \frac{Ctz(f_i, u_j, c_l)}{\alpha^{(\lambda_k - l)}}$$

c_l : a cycle; λ_k : the length of session s_k ;
 $Ctz(f_i, u_j, c_l) = 1$ if at cycle c_l user u_j criticized on feature f_i ;
 α : the importance difference between two consecutive critiques

The importance weight of feature f_i computed at **a user level**, if the number of critiques expressed by the user is large enough

$$w_i(u_j) = \frac{1}{\|S(u_j)\|} \cdot \sum_{k=1}^{\|S(u_j)\|} \frac{w_i(u_j, s_k)}{\beta^{(\|S(u_j)\| - k)}}$$

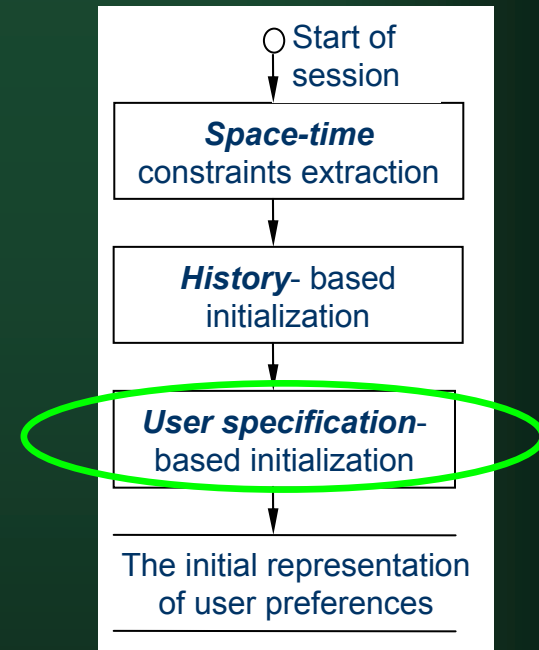
$S(u_j)$: the set of all sessions of user u_j ;
 $\beta (> 1)$: the drifting effect of old sessions

The importance weight of feature f_i computed at **the population level**, if the user is a new one.

$$w_i(U) = \frac{1}{\|U\|} \cdot \sum_{u_j \in U} w_i(u_j)$$

U : the set of all users (who have expressed at least one critique)

- This initialization phase is activated *only if* the user is willing to specify some initial preferences.
- These explicit preferences overwrite the history-based representation of the user's preferences.
- Regarding a preference explicitly specified, its type determines which of (Q_L, p, w) must be overwritten.



An example:

- The history-based representation of the user favorite pattern:

$$p^{\text{history}} = (p_1, p_2, p_3, \dots, p_n)$$

- Let's suppose the user explicitly specifies two initial wish conditions on the **1st** and the **3rd** features. After the third phase of initialization:

$$p^{\text{init}} = (p_1', p_2, p_3', \dots, p_n)$$

- Users are provided with two options for explicitly specifying initial preferences: *sample-oriented* or *feature-oriented*
- In a sample-oriented start-up
 - The user wants to find those products similar to a known one
 - This sample product serve as a starting point for the system's search process, and is used to overwrite (p)
- In a feature-oriented start-up
 - The user specifies the values of (or the constraints on) some particular product features
 - Regarding a preference expressed on a feature, its type determines which of (Q_L, p, w) should be overwritten

The image shows two screenshots of a mobile application interface. The top screenshot is titled "Start-up screen..." and asks "Would you like to specify any search conditions?". It has three radio button options: "No, use my profile", "Let me specify", and "Should be similar to". The "Should be similar to" option is selected. Below the options is a text input field containing "La Berta". The bottom screenshot is titled "Define your requirements" and shows a "Max cost: € 20" field. It has two columns of checkboxes: "Type" and "Characteristic". The "Type" column includes Pizzeria, Restaurant, Spaghetteria, Trattoria, and Typical. The "Characteristic" column includes Admitted an..., Air conditio..., Hotel, Meeting room, Outdoor tab..., Parking, and Smoking room. The "Smoking room" checkbox is checked. At the bottom of the second screenshot are "Back" and "Go" buttons.

The applicability of the proposed methodology

- The proposed initialization methodology assumes that there is *a relationship* between the preferences expressed in the past sessions and those in the current one.

- If this assumption does not hold?

- User's explicit preferences could be very different from those expressed in the past.
- E.g., Let's suppose that the user specifies two wish conditions on the 1st and 2nd features. Let's consider the two following representations:

$$\begin{aligned} p^{\text{overwritten}} &= (p_1', p_2', p_3, \dots, p_n) ; & w^{\text{overwritten}} &= (w_{12}', w_{12}', w_3', \dots, w_n') \\ p &= (p_1', p_2', ?, \dots, ?) ; & w &= (0.5, 0.5, 0, \dots, 0) \end{aligned}$$

The second representation could be better, since it avoids the misleading affects caused by those features not explicitly specified by the user.

- It is probably better
 - to discard any system guess when the user expresses some initial preferences, and
 - to rely totally on these explicit preferences.

- An important and common problem for every recommender system is the precision of *the system's initial guess of the user's preferences*
- In this paper, we have proposed an approach which
 - exploits different sources of user related knowledge
 - integrates a persistent user model with a context-dependent one; and
 - aims at providing critique-based recommendations for mobile users
- We are in the progress of implementing the proposed approach (for user preferences representation initialization) into MobyRek
 - All the GUI snapshots here illustrated are taken from an old version of MobyRek (which does not support the proposed approach of user preferences representation initialization)

- Conducting experiments to validate the appropriateness of the proposed approach regarding the quality of first-cycle recommendations
- Verifying how the exploitation of pre-travel plans contributes to the recommendation quality

Thank you for the attention!