Chapter

Can a Recommender System induce serendipitous encounters?

Leo Iaquinta, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, Piero Molino
University of Bari
Italy

1. Introduction

Today recommenders are commonly used with various purposes, especially dealing with e-commerce and information filtering tools. Content-based recommenders rely on the concept of similarity between the bought/searched/visited item and all the items stored in a repository. It is a common belief that the user is interested in what is similar to what she has already bought/searched/visited. We believe that there are some contexts in which this assumption is wrong: it is the case of acquiring unsearched but still useful items or pieces of information. This is called serendipity. Our purpose is to stimulate users and facilitate these serendipitous encounters to happen.

This chapter presents the design and implementation of a hybrid recommender system that joins a content-based approach and serendipitous heuristics in order to mitigate the over-specialization problem with surprising suggestions.

The chapter is organized as follows: Section 2 presents background and motivation; Section 3 introduces the serendipity issue for information seeking; Section 4 covers strategies to provide serendipitous recommendations; Section 5 provides a description of our recommender system and how it discovers potentially serendipitous items in addition to content-based suggested ones; Section 6 provides the description of the experimental session carried out to evaluate the proposed ideas; finally, Section 7 draws conclusions and provides directions for future work.

2. Background and Motivation

Information overload is a common issue among the modern information society. Information Filtering (IF) is a kind of intelligent computing techniques that mitigates this problem by providing the user with the most relevant information with respect to her information needs.

Recommender systems (RSs) adopt IF techniques in order to provide customized information access for targeted domains.

They can be viewed as intelligent systems that take input directly or indirectly from users and, based on their needs, preferences and usage patterns, provide personalized advices about products or services and can help people to filter useful information.

Several definitions of RS have been given. According to (Burke, 2002): "*Recommender systems have the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options*". This definition makes it clear that user oriented guidance is critical in a RS.

Among different recommendation techniques proposed in the literature, the content-based and the collaborative filtering approaches are the most widely adopted to date. Systems implementing the content-based recommendation approach analyze a set of documents, usually textual descriptions of the items previously rated by an individual user, and build a model or profile of user interests based on the features of the objects rated by that user (Mladenic, 1999). The profile is exploited to recommend new items of interest. Collaborative recommenders differ from content-based ones in that user opinions are used instead of content. They gather ratings about objects by users and store them in a centralized or distributed database. To provide the user X with recommendations, the system computes the neighborhood of that user, i.e. the subset of users that have a taste similar to X. Similarity in taste is computed based on the similarity of ratings for objects that were rated by both users. The system then recommends objects that users in X's neighborhood indicated to like, provided that they have not yet been rated by X.

Each type of filtering methods has its own weaknesses and strengths. In particular, the content-based approach suffers from *over-specialization*. When the system can only recommend items that score highly against a user's profile, the user is limited to being recommended items similar to those already rated. Even a 'perfect' content-based technique would never find anything surprising, limiting the range of applications for which it would be useful. This shortcoming is called *serendipity problem*.

To give an example, a person with no experience with Greek cuisine would never receive a recommendation for even the greatest Greek restaurant in town.

In other words, over-specialized systems can prevent serendipitous discoveries to happen, according to Gup's theory (Gup, 1997).

It is useful to make a clear distinction between *novelty* and *serendipity*. As explained by Herlocker (Herlocker et al., 2004), novelty occurs when the system suggests to the user an unknown item that she might have autonomously discovered. A serendipitous recommendation helps the user to find a surprisingly interesting item that she might not have otherwise discovered (or it would have been really hard to discover). To provide a clear example of the difference between novelty and serendipity, consider a recommendation system that simply recommends movies that were directed by the user's favorite director. If the system recommends a movie that the user was not aware of, the movie will be novel, but probably not serendipitous. On the other hand, a recommender that recommends a movie by a new director is more likely to provide serendipitous recommendations. Recommendations that are serendipitous are by definition also novel.

Novelty is the main objective of a "classical" recommender. We agree with the theory proposed by McNee (McNee et al., 2006), that studies about improving precision and recall (or accuracy metrics in general) just do not get the point of what is useful for the user: a sensible recommendation (which is not always the most accurate one).

Our objective is to try to feed the user also with recommendations that could possibly be serendipitous. Thus, we enrich the architecture of content-based RS with a component devoted to introduce serendipity in the recommendation process.

The demonstrative scenario concerns with personalized museum tours where the serendipitous suggested items are selected exploiting the learned user profile and causing slight diversions on the personalized tour. Indeed content-base recommender module allows to infer the most interesting items for the active user and, therefore, to arrange them according the spatial layout, the user behavior and the time constraint. The resulting tour potentially suffers from over-specialization and, consequently, some items can be found no so interesting for the user. Therefore the user starts to divert from suggested path considering other items along the path with growing attention. On the other hand, also when the recommended items are actually interesting for the user, she does not move with blinkers, i.e. she does not stop from seeing artworks along the suggested path. These are opportunities for serendipitous encounters. These considerations suggest to perturb the optimal path with items that are programmatically supposed to be serendipitous for the active user.

3. Serendipity and information seeking

Horace Walpole coined the term "serendipity" in the 1754 explaining it as "*making discoveries by accident and sagacity of things which one is not on quest of*". The origin of the word "serendipity" (van Andel, 1994) is the persian fairy tale titled "The three princes of Serendip" that Cristoforo Armeno translated and published in the 1557. M. K. Stoskopf (Foster & Ford, 2003) was one of the first scientists to acknowledge the relevance that serendipity covers in scientific field, affirming that serendipitous discoveries are of significant value in the advancement of science and often presents the found for important intellectual leaps of understanding.

The history of science is full of serendipitous discoveries: the (re-)discovery of the Americas by Columbus, the Gelignite by Nobel, the Penicillin by Fleming, etc.

We agree with Roberts (Roberts, 1989) when he stresses that serendipitous encounters depend on the characteristic of the information seeker, her open minded attitude, her wide culture and her curiosity.

The idea of serendipity has a link with de Bono's "lateral thinking" (De Bono, 1990) which consists not to think in a selective and sequential way, but accepting accidental aspects, that seem not to have relevance or simply are not sought for. This kind of behavior surely helps the awareness of serendipitous events.

The subjective nature of serendipity is certainly quite a problem when trying to conceptualize, analyze and implement it. As Foster & Ford said: "*Serendipity is a difficult concept to research since it is by definition not particularly susceptible to systematic control and prediction. […] Despite the difficulties surrounding what is still a relatively fuzzy sensiting concept, serendipity would appear to be an important component of the complex phenomenon that is information seeking*" (Foster & Ford, 2003). Even though we agree with van Andel (van Andel, 1994) that we cannot program serendipity because of its nature, we share the concern of Campos and de Figueiredo (Campos & de Figueiredo, 2001a) of programming for serendipity. They also tried to suggest a formal definition of serendipity (Campos & de Figueiredo, 2001b) identifying different categories for serendipitous encounters.

By the way, the problem of programming for serendipity has not been deeply studied and there are really few theoretical and experimental studies.

The noble objective of allowing users expand their knowledge and preserve the opportunity of making serendipitous discoveries even in the digital libraries could push the development of useful tools that can facilitate important intellectual leaps of understanding.

Like Toms explains (Toms G., 2000), there are three kind of information searching:

- seeking information about a well-defined object;
- seeking information about an object that cannot be fully described, but that will be recognized at first sight;
- acquiring information in an accidental, incidental, or serendipitous manner.

It is easy to realize that serendipitous happenings are quite useless for the first two ways of acquisition, but are extremely important for the third kind.

To accomplish the goal of implementing a serendipity–inducing module for a content-based recommender, the appropriate metaphor in a real-world situation could be one of a person visiting a museum (or going for shopping in a virtual space) who, while walking around, would find something completely new that she has never expected to find, that is definitely interesting for her.

4. Strategies to induce serendipity

Introducing serendipity in the recommendation process requires an operational strategy. Among different approaches which have been proposed, Toms suggests four strategies, from simplistic to more complex ones (Toms G., 2000):

1) Role of chance or 'blind luck', implemented via a random information node generator.
2) Pasteur principle ("chance favors the prepared mind"), implemented via a user profile.
3) Anomalies and exceptions, partially implemented via poor similarity measures.
4) Reasoning by analogy, whose implementation is currently unknown.

In this chapter we propose an architecture for content-based RSs that implements the "Anomalies and exceptions" approach, in order to provide serendipitous recommendations alongside classical ones.

The basic assumption is that serendipity cannot happen if the user already knows what is recommended to her, because a serendipitous happening is by definition something new. Thus the lower is the probability that user knows an item, the higher is the probability that a specific item could result in a serendipitous recommendation. The probability that user knows something semantically near to what the system is confident she knows is higher than the probability of something semantically far. If we evaluate semantic distance with a similarity metric, like internal product which takes into account the item description to build a vector and compares it to other item vectors, it results that it is more probable to get a serendipitous recommendation providing the user with something less similar to her profile.

According to this idea, items should not be recommended if they are too similar to something the user has already seen, such as different news article describing the same event.

Therefore, some content-based RSs, such as DailyLearner (Billsus & Pazzani, 2000), filter out items not only if they are too different from the user preferences, but also if they are too similar to something the user has seen before. Following this principle, the basic idea underlying the proposed architecture is to ground the search for potentially "serendipitous" items on the similarity between the item descriptions and the user profile, as described in the next section.

5. Inducing serendipity in a content-based recommender

The starting point to provide serendipitous recommendations consists in a content-based recommender system developed at the University of Bari (Degemmis et al., 2007; Semeraro et al., 2007). The system is capable of providing recommendations for items in several domains (e.g., movies, music, books), provided that descriptions of items are available as text metadata (e.g. plot summaries, reviews, short abstracts). In the following, we will refer to documents as textual metadata about items to be recommended.

Fig. 1 **Errore. L'origine riferimento non è stata trovata.**shows the general architecture of the system evolved to provide also serendipitous suggestion within the museum scenario.

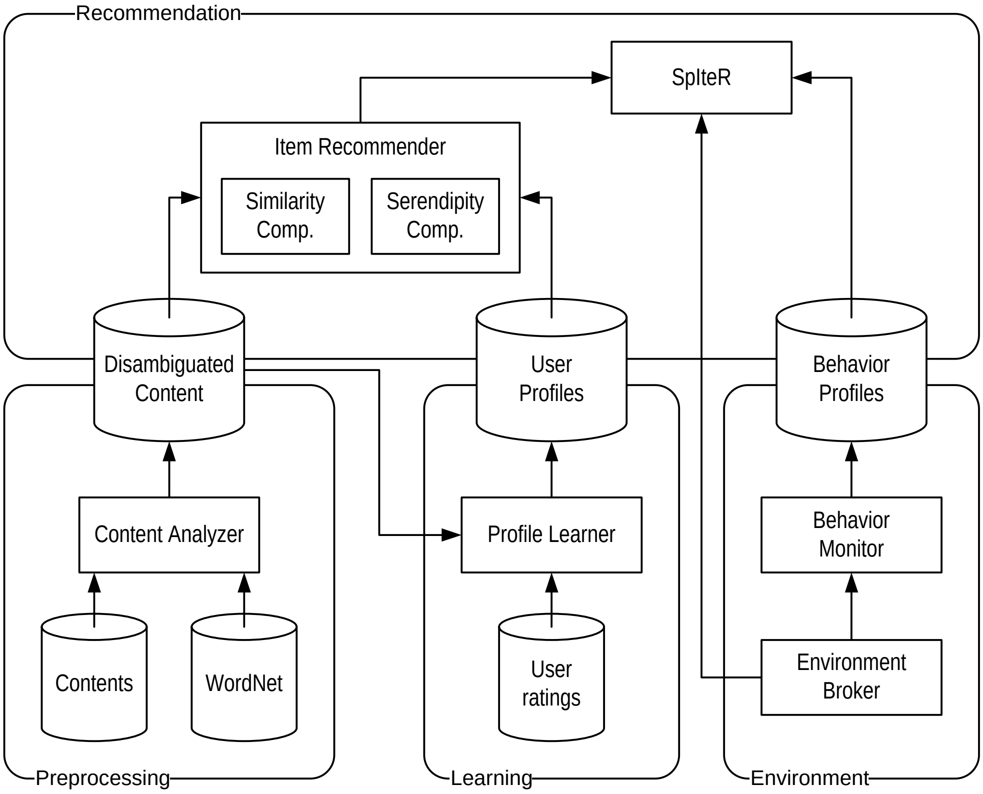


Fig. 1. General system architecture

The recommendation process is performed in several steps, each of which is handled by a separate component of the architecture shown in **Errore. L'origine riferimento non è stata trovata.**. First, given a collection of documents, a preprocessing step is performed by the *Content Analyzer*, which uses the WordNet lexical database to perform *Word Sense Disambiguation* (WSD) to identify correct senses, corresponding to concepts identified from words in the text.

Subsequently, a learning step is performed by the *Profile Learner* on the training set of documents, to generate a probabilistic model of the user interests. This model is the user profile including those concepts that turn out to be most indicative of the user's preferences.

Then, the *Item Recommender* component implements a naïve Bayes text categorization algorithm to classify documents as interesting or not for a specific user by exploiting the probabilistic model learned from training examples. In addition, the *Item Recommender* contains a sub-module implementing heuristics to provide *serendipity computation*.

Finally, the *SpIteR* (Spatial Item Recommender) module rearranges the suggested items in a personalized tour using information about environment and user behavior.

5.1. Content Analyzer

It allows introducing semantics in the recommendation process by analyzing documents in order to identify *relevant concepts* representing the content. This process selects, among all the possible meanings (senses) of each polysemous word, the correct one according to the context in which the word occurs. In this way, documents are represented using concepts instead of keywords, in an attempt to overcome the problems due to natural language ambiguity. The final outcome of the preprocessing step is a repository of disambiguated documents. This semantic indexing is strongly based on natural language processing techniques and heavily relies on linguistic knowledge stored in the WordNet lexical ontology (Miller, 1995).

The core of the Content Analyzer is a procedure for Word Sense Disambiguation (WSD), called JIGSAW (Basile et al., 2007). WSD is the task of determining which of the senses of an ambiguous word is invoked in a particular use of that word. The set of all possible senses for a word is called *sense inventory* that, in our system, is obtained from WordNet. The basic building block for WordNet is the *synset* (*syn*onym *set*), which contains a group of synonymous words that represents a concept. Since it is not the focus of the chapter, the procedure is not described here. What we would like to underline here is that the WSD procedure allows to obtain a synset-based vector space representation, called bag-of-synsets (BOS), that is an extension of the classical bag-of-words (BOW) model. In the BOS model a synset vector, rather than a word vector, corresponds to a document. Our idea is that BOS-indexed documents can be used in a content-based information filtering scenario for learning accurate, sense-based user profiles, as discussed in the following section.

5.2. Profile Learner

It implements a supervised learning technique for learning a probabilistic model of the interests of the active user from disambiguated documents rated according to her interests. This model represents the semantic profile, which includes those concepts that turn out to be most indicative of the user preferences.

We consider the problem of learning user profiles as a binary Text Categorization task (Sebastiani, 2002), since each document has to be classified as interesting or not with respect to the user preferences. Therefore, the set of categories is restricted to POS, that represents the positive class (user-likes), and NEG the negative one (user-dislikes). The induced probabilistic model is used to estimate the *a-posteriori* probability, $P(X|d)$, of document d belonging to class X.

The algorithm adopted for inferring user profiles follows a Naïve Bayes text learning approach, widely used in content-based recommenders (Mladenic, 1999). More details are reported in (Semeraro et al., 2007). What we would like to point out here is that the final outcome of the learning process is a text classifier able to categorizes a specified item in two classes: POS (for the item the user should like) and NEG (for the item the user should not like). Given a new document d, the model computes the *a-posteriori* classification scores $P(\mathrm{POS}|d)$ and $P(\mathrm{NEG}|d)$ by using probabilities of synsets contained in the user profile and estimated in the training step. The classifier is inferred by exploiting items labeled with ratings from 0 to 5 (items rated from 0 to 2 are used as training examples for the class NEG, while items rated from 3 to 5 are used as training examples for POS).

5.3. Item Recommender

It exploits the user profile to suggest relevant documents, by matching concepts contained in the semantic profile against those contained in documents to be recommended. The module devoted to discover potentially serendipitous items has been included in this component, in addition to the module which is responsible for the similarity computation between items and profiles.

In order to integrate Toms' ”poor similarity“ within the recommender, a set of heuristics has been included in the module for *serendipity computation*.

The module devoted to compare items with profiles (*Similarity Computation*) produces a list of items ranked according to the a-posteriori probability for the class POS. That list will contain on the top the most similar items to the user profile, i.e. the items high classification score for the class POS. On the other hand, the items for which the a-posteriori probability for the class NEG is higher, will ranked lower in the list.

The items on which the system is more uncertain are the ones for which difference between the two classification scores for POS and NEG tends to zero. We could reasonably assume that those items are not known by the user, since the system was not able to clearly classify them as relevant or not. Therefore, one of the heuristics included in the serendipity module takes into account the absolute value of the difference of the probability of an item to belong to the two classes: $|P(\mathrm{POS}|d)\text{-}P(\mathrm{NEG}|d)|$. The items for which the lowest difference $|P(\mathrm{POS}|d)\text{-}\underline{P}(\mathrm{NEG}|d)|$ is observed are the most uncertainly categorized, thus it might result to be the most serendipitous ones. Beside the most serendipitous function, serendipitous items can be also pseudo-ranked, i.e. the top most items are randomly arranged to avoid that the repeated requesting obtains the same result for the same un-updated user profile.

5.3. SpIteR

It dynamically arranges the suggested items to make the user experience more enthralling. Indeed, the Item Recommender is able to provide a static ordered list of items according to the user assessed interests, but it does not rely on the user interaction with environment. Besides, if the suggested tour simply consists of the enumeration of ranked items, the path is too tortuous and with repetitive passages that make the user disoriented, especially under a time constraint. Finally, different users interact with environment in different manner, e.g. they travel with different speed, they spend different time to admire artworks, they divert from the suggested tour. Consequently, the suggested personalized tour must be dynamically updated and optimized according to contextual information on user interaction with environment.

The tour suggestion task requires knowledge about item layout. We propose to basically represent items as nodes of an Euclidean graph. Thus the museum tour is quite similar to the classical *Traveling Salesperson Problem* (TSP). TSP aims to find the shortest cycle visiting each node exactly once (Lawler et al., 1985). TSP is a well known combinatorial optimization problem and has been studies extensively in many variants. The problem is NP-hard, as shown by Papadimitriou (Papadimitriou, 1977), even if the weight of each edge satisfies the triangle inequality (in addition to nonnegative and symmetry properties) like in Euclidean graphs.

The TSP combinatorial complexity makes difficult to achieve a methodology to solve efficiently TSP of realistic size to global optimality. A number of different methods have been proposed for obtaining either optimal or near optimal solutions, ranging from classic methodologies based on linear programming (Wong, 1980) and branch-and-bound (Little et al., 1963; Bellmore & Malone, 1971) to artificial intelligence methods such as neural networks (Bhide et al., 1993), tabu search (Glover, 1990), and genetic algorithms (Goldberg & Lingle, 1985).

SpIteR does not uses only one method, but here some aspects of genetic one are shortly presented. Genetic algorithms (GAs) are search algorithms that work via the process of natural selection. They begin with a sample set of potential solutions which then evolves toward a set of more optimal solutions. A solution (i.e., a *chromosome* for GAs) for the museum tour is a sequence of suggested items. GAs require that a potential solution can be break into discrete parts, namely *genes*, that can vary independently. It's necessary to evaluate how “good” a potential solution is with respect to other potential solutions. The *fitness function* is responsible for performing this evaluation and the returned *fitness value* reflects how optimal the solution is: the higher the number, the better the solution. In the museum tour scenario, the fitness function relies on a user-sensitive time constraint, the user behavior (i.e., speed and stay times), the user learned preferences and the item layout.

Further information about museum tour items can be exploited to obtain the problem solution. Usually, few items are placed in rooms and each room is connected with some other rooms. A sample layout of rooms is shown in Fig. 2 with the schematic representation of Vatican Pinacoteca. Rooms provide a simpler perspective from combinatorial complexity point of view, but, while tour visits each item at most once (like in *k*-TPS), each room can be visited more times[1].

[1] a user should anyway cross each room access at most twice

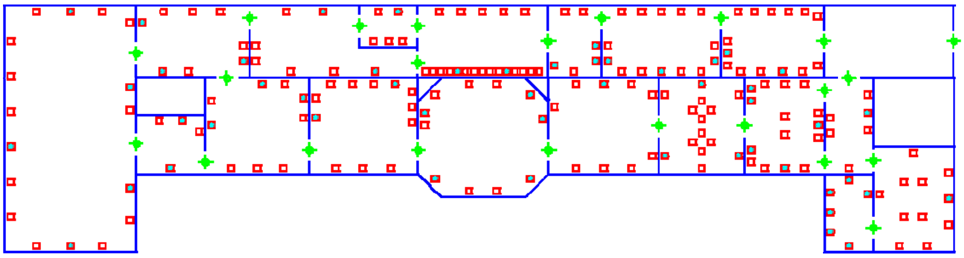


Fig. 2. Schematic representation of Vatican Pinacoteca layout

User profile can be exploited to further reduce nodes involved in the TSP for museum scenario. Indeed the suggested tour should consist of recommended items, i.e., the *k* most interesting items for the active user. The question about *k* value selection arises. Intuitively, the *k* value depends on how long should be the personalized tour, e.g., the preferred tour duration and the user behavior must be counted. Thus the estimate of *k* value mainly deals with the overall time constraint on expected time spent by the user to admire the supposed most interesting museum items. That estimate is allowed to be rough (i.e., the actual time spent to reach items can be disregarded) since the initial goal is to cut back the supposed less interesting nodes. Fig. 3 shows a sample tour consisting of the *k* most interesting items sorted according to the ranking provided by the Item Recommender: bend lines are only a graphical expedient to avoid hidden segments.

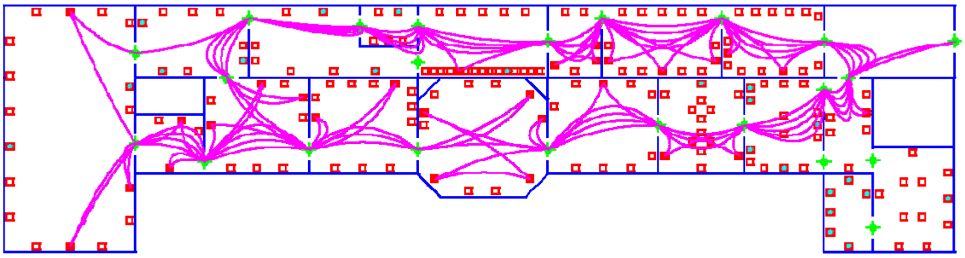


Fig. 3. A sample tour consisting of the ranked *k* most interesting items

Speed and stay times are parameters related to user behavior. At the beginning, they are estimated on the basis of a stereotypical user profile (Shapira et al., 1997) and then updated according to data collected during the tour from actual user behavior by the *Behavior Monitor* module. Hence the *k* value can change during the tour: that is not an issue, since the tour is constantly monitored to detect significant deviation from proposed one. When the user behavior requires too many significant updates to the behavior profile or the user skips a recommended item or she stays in front of un-recommended items, the tour is planned again taking into account the previous user behavior and the actual viewed items.

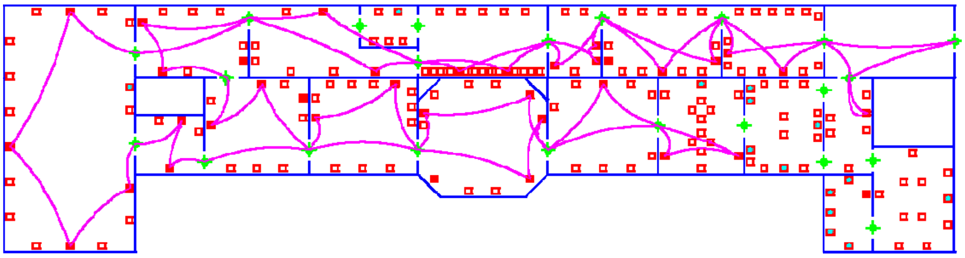


Fig. 4. The optimized version of personalized tour in Fig. 3

Once the personalized tour is achieved starting from the *k* most interesting items, as shown in Fig. 4, serendipitous disturbs are applied. Indeed, the ranked list of serendipitous items is obtained from the *Item Recommender* module and the previous personalized tour is augmented with some serendipitous items along the path. The resulting solution most likely has a worse fitness value and then a further optimization step is performed. However, the further optimization step should cut away exactly the disturbing serendipitous items, since they compete with items that are more similar with the user tastes. Therefore serendipitous items should be differently weighed from the fitness function, for instance changing their stay time. Indeed, the supposed serendipitous items should turn out not so serendipitous and the user should reduce her stay time in front of such items. Once again, this is not a problem since the user behavior is constantly monitored and tour dynamically updated. Fig. 5 shows an "good enough" personalized tour consisting of the most interesting items and the most serendipitous ones. It is amazing to note that some selected serendipitous items are placed in rooms otherwise unvisited.

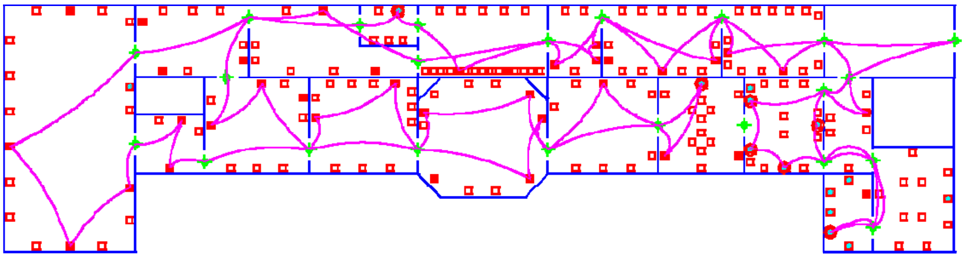
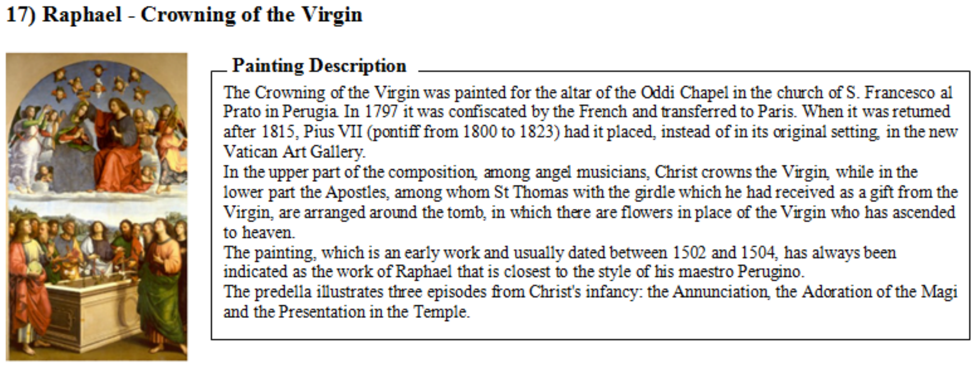


Fig. 5. The optimized version of personalized tour whit serendipitous items

6. Experimental Session

The goal of the experimental evaluation was twofold: to investigate different strategy for providing serendipitous recommendations and to evaluate the serendipity augmenting effects on personalized tours.

The experimentations we conducted was based over a corpus of 45 paintings chosen from the collection of the Vatican picture-gallery. The dataset was collected using screenscraping bots, which captured information from the official website[2] of the Vatican picture-gallery. In particular, for each element in the dataset an image of the artefact and three textual metadata (title, artist, and description) were collected. **Errore. L'origine riferimento non è stata trovata.** shows a sample of the resulting item. All artworks were laid in the schematic environment model of the Pinacoteca in order to deal with the spatial layout influence on recommending.



17) Raphael - Crowning of the Virgin

Painting Description

The Crowning of the Virgin was painted for the altar of the Oddi Chapel in the church of S. Francesco al Prato in Perugia. In 1797 it was confiscated by the French and transferred to Paris. When it was returned after 1815, Pius VII (pontiff from 1800 to 1823) had it placed, instead of in its original setting, in the new Vatican Art Gallery.
In the upper part of the composition, among angel musicians, Christ crowns the Virgin, while in the lower part the Apostles, among whom St Thomas with the girdle which he had received as a gift from the Virgin, are arranged around the tomb, in which there are flowers in place of the Virgin who has ascended to heaven.
The painting, which is an early work and usually dated between 1502 and 1504, has always been indicated as the work of Raphael that is closest to the style of his maestro Perugino.
The predella illustrates three episodes from Christ's infancy: the Annunciation, the Adoration of the Magi and the Presentation in the Temple.

Fig. 6. Sample of dataset item

We involved 30 users who voluntarily took part in the experiments. The average age of the users was in the middle of twenties. None of the users was an art critic or expert. Users were requested to express their preferences for collection items as a numerical vote on a 5-point scale (1=strongly dislike, 5=strongly like).

The ratings have been used for *K*-fold cross validation (Kohavi, 1995) that gave back an average degree of precision, recall and *F*-measure (Herlocker et al., 2004). We simulated the user interaction with the system using a small part of the ratings of each user for the training of the classifier. Then five iterations were performed, in which a serendipitous item was selected by the module, rated with the ratings already expressed and added to the training set. The ratings for 5 serendipitous items proposed to each one of the 30 users were collected. The whole process has been done with the most serendipitous function and for the random over most serendipitous function with a numeric threshold of 5%, 10% and 15% of the database.

Investigating different strategy to provide serendipitous recommendation, the rating interpretation issue comes out. Indeed, from a pragmatic point of view, ratings must be homogenous with other ratings in order to allow their subsequent exploitation in the profile learning step, but user rating motivations affect the meaning evaluation of finding unknown and possibly interesting things, and not simply interesting ones. For instance, a poorly rating for serendipitous suggested items should come from the experience of the user (the user already knows the item), from her lack of interest (the user already knows the item and

[2] http://mv.vatican.va/3_EN/pages/PIN/PIN_Main.html

is not interested in it), from her lack of interest in finding new things (the user does not know the item and has no interest in knowing something new), from the conscious expression of dislike (the user did not know the item before, now she knows it but she does not like it or is not interested in it) or from a serendipitous encounter (before-unknown item that results to be interesting for the user).

The results of the experimentation (Table 1) showed that the average percentage of POS items (ratings better than 3) were 40,67% for the most serendipitous function, 42,67% for the random over most serendipitous function with a threshold of 5% of the database size, 46,67% with a 10% threshold and 48,67% with a 15% threshold.

The results present a trend: with a larger threshold of randomness there are more good ratings. This could be interpreted as follows: the randomness of the selection over the most serendipitous item helps improving the ratings. So the best function would be the one with a more wide threshold of randomness. But, as the average POS ratings increase and better ratings means more similar items, we can hypothesize that suggested items are more semantically near to user tastes and knowledge so it is less probable that they are unknown. In this case the best function would be the most serendipitous one.

	Average POS
Most serendipitous	40.57%
Random over most serendipitous (5% threshold)	42.57%
Random over most serendipitous (10% threshold)	46.67%
Random over most serendipitous (15% threshold)	48.67%

Table 1. Four functions average results

In order to evaluate the serendipity augmenting effects on personalized tours, the learned profiles were used to obtain personalized tours with different time constraints and different serendipitous disturbs. Five time constraints were chosen so that tours consisted approximately of 10, 15, 20, 25, 30 items. Serendipitous items ranged from 0 to 7.

Table 2 reports the average of sums and means of POS values of tours. Fig. 7 helps in the data interpretation: the serendipitous item augmenting causes the exploiting of items less similar to the user tastes and this effect is particularly evident when there are too many serendipitous items. On the other hand, there is also a decrease when many items are selected according to the user profile, since they are progressively less interesting. When there are many items, the serendipitous item augmenting seems to have no effects over POS mean, but probably this comes from the not very large dataset used.

	T_1		T_2		T_3		T_4		T_5	
S0	7.18	0.711	10.69	0.714	14.02	0.705	17.21	0.679	19.94	0.671
S1	7.15	0.708	10.61	0.709	14.00	0.704	17.20	0.679	19.89	0.670
S2	7.12	0.705	10.59	0.708	13.98	0.702	17.20	0.679	19.88	0.669
S3	7.08	0.701	10.60	0.708	13.96	0.702	17.19	0.679	19.87	0.669
S4	7.03	0.696	10.58	0.707	13.96	0.701	17.19	0.678	19.87	0.669
S5	6.88	0.681	10.52	0.703	13.95	0.701	17.17	0.678	19.85	0.668
S6	6.54	0.647	10.42	0.696	13.90	0.698	17.11	0.676	19.75	0.665
S7	6.17	0.611	10.19	0.681	13.76	0.692	16.99	0.671	19.64	0.661
Items	*10.10*		*14.97*		*19.90*		*25.33*		*29.70*	

Table 2. Sums and means of POS values of tours

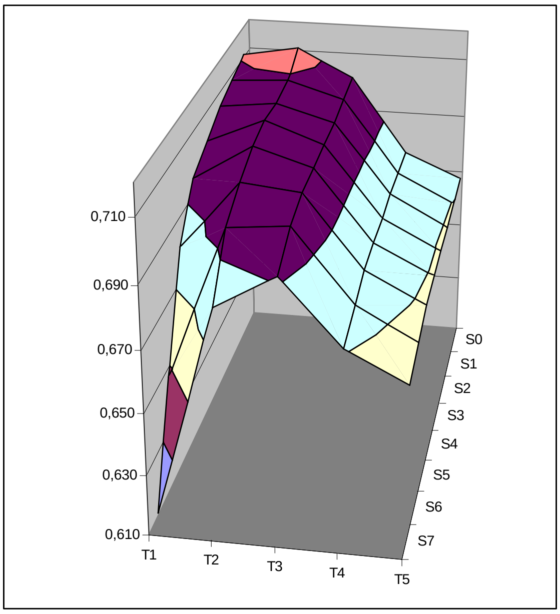


Fig. 7. Means of POS values of tours

Table 3 reports percentages of walking time over the tour. Data show that, increasing the time constraint, less time is spent to walk. Indeed, if few items are selected, they are scattered around (proportionally) many rooms and the user visits room with very few and even no one suggest item. The serendipitous item augmenting seems to increase the walking time. This result is quite amazing according to the selection serendipitous item strategy, i.e., items that are along to a previously optimized path. Actually, the walking time percentage mainly increases because serendipitous items are introduced as new genes of a "good enough" chromosome (solution). However, the augmented chromosome tends to evolve toward the previous one. Thus the new genes should be promoted with a benefit over the fitness function: the reduction in their supposed stay time. This approach is simple and intuitive, but it makes difficult the interpretation of expected walking time percentage. Indeed the variation on walking time becomes from path variations, but the total tour time is also changed on account of the technical issue about the GA fitness function.

	T_1	T_2	T_3	T_4	T_5	
S0	39.9%	34.0%	34.6%	31.6%	30.2%	**34.1%**
S1	42.6%	36.3%	36.0%	32.8%	31.3%	**35.8%**
S2	45.0%	38.1%	37.4%	34.0%	32.2%	**37.4%**
S3	49.7%	40.1%	38.3%	34.5%	33.5%	**39.2%**
S4	52.7%	42.0%	39.9%	36.3%	34.6%	**41.1%**
S5	56.0%	45.5%	41.9%	37.8%	35.9%	**43.4%**
S6	60.0%	47.5%	43.7%	39.7%	37.2%	**45.6%**
S7	65.2%	51.7%	45.6%	41.7%	39.0%	**48.6%**

Table 3. Percentages of walking time

Moreover, the effects of serendipitous items on expected walking time are analyzed with respect to the starting optimized tours (S0), i.e. the previously discussed drawback is partially cut off. Table 4 and Fig. 8 show that few disturbs cause a quite uniform increase of the walking time: the ground becomes from the slight deviations on S0 tour. On the other hand, growing the number of serendipitous items, the deviations are amplified. This is more evident for the shortest S0 tours, since many serendipitous items can encourage the "exploration" of rooms untouched by S0, about Fig. 5.

	T_1	T_2	T_3	T_4	T_5
S1	106%	106%	104%	103%	103%
S2	112%	112%	108%	107%	107%
S3	124%	119%	112%	110%	111%
S4	131%	126%	117%	115%	115%
S5	141%	136%	123%	121%	120%
S6	150%	143%	130%	127%	125%
S7	164%	155%	135%	134%	131%

Table 4. Increment of walking time for tours with serendipitous items

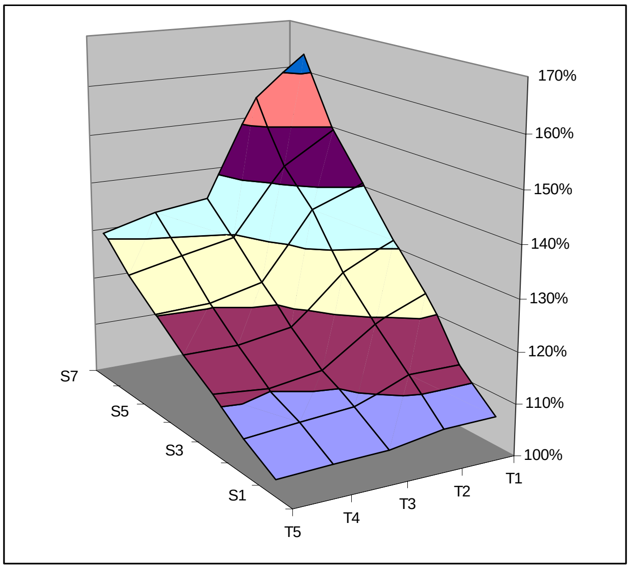


Fig. 8. Serendipitous effects on walking time

7. Conclusions and Future work

This chapter presents a beginning effort to apply some ideas about serendipity to information retrieval and information filtering systems, especially in recommenders, to mitigate the over-specialization issue. The museum scenario is particularly interesting because items are arranged in a physical space and users interact with the environment. Thus disregarding context facets makes useless recommendations. Similar remarks are still valid in domains (different from cultural heritage fruition) in witch a physical or virtual space is involved and it represents a pragmatic justification to explain (supposed) serendipitous recommendations.

As future work, we expect to carry out more extensive experimentation with more users and wider item collections. We plan also to gather user feedback and feeling by questionnaires focused on qualitative evaluation of the recommendations and the idea of getting suggestions that should surprise them. That is really important for the need to understand the effectiveness of the module in finding unknown items rather the ones that result best rated. Experimentation with users with different cultural levels and with different information seeking tasks are also important to find out which kind of user would like most serendipitous recommendations and to whom they are more useful.

We expect also to implement the other suggestions given by Toms (Toms G., 2000) and to develop further the heuristic proposed (maybe padding a parameter factor that multiplies the probabilities in order to balance better between categories) or also introduce new heuristics and make an experimental comparison.

8. References

Basile, P.; Degemmis, M.; Gentile, A. L.; Lops, P. & Semeraro, G. (2007). UNIBA: JIGSAW algorithm for Word Sense Disambiguation, *Proceedings of the 4th ACL 2007 International Worshop on Semantic Evaluations (SemEval-2007)*, pp. 398-401, Prague, June 2007, Association for Computational Linguistics (ACL)

Bellmore, M. & Malone, J. C. (1971). Pathology of traveling-salesman subtour elimination algorithms. *Operations Research*, 19, 2, (March-April 1971), pp. 278-307, ISSN 0030-364X

Bhide, S.; John, N. & Kabuka, M. R. (1993). A Boolean neural network approach for the traveling salesman problem. *IEEE Transactions on Computers*, 42, 10, (October 1993), pp. 1271-1278, ISSN: 0018-9340

Billsus, D. & Pazzani, M. J. (2000). User Modeling for Adaptive News Access. *User Modeling & User-Adapted Interaction*, 10, 2-3, (June 2000), pp. 147-180, ISSN: 0924-1868

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12, 4, (November 2002), pp. 331-370, ISSN: 0924-1868

Campos, J. & de Figueiredo, A. (2001a). Searching the Unsearchable: Inducing Serendipitous Insights, *Proceedings of the Workshop Program at the 4th International Conference on Case-Based Reasoning (ICCBR 2001)*, pp. 159-164

Campos, J. & de Figueiredo, A. (2001b). The Serendipity Equations, *Proceedings of the Workshop Program at the 4th International Conference on Case-Based Reasoning (ICCBR 2001)*, pp. 121-124

De Bono, E. (1990). *Lateral Thinking: A Textbook of Creativity*, Penguin Books, ISBN: 9780140137798, London

Degemmis, M.; Lops, P. & Semeraro, G. (2007). A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research (UMUAI)*, 17, 3, pp. 217-255, ISSN: 0924-1868 (Print) 1573-1391 (Online)

Foster, A. & Ford, N. (2003). Serendipity and information seeking: an empirical study. *Journal of Documentation*, 59, 3, pp. 321-340, ISSN: 0022-0418

Glover, F. (1990). Artificial Intelligence, Heuristic Frameworks and Tabu Search. *Managerial and Decision Economics*, 11, 5, (November 2006), pp. 365-378, ISSN: 0143-6570 (printed) 1099-1468 (electronic)

Goldberg, D. E. & Lingle, R. J. (1985). Alleles, loci, and the traveling salesman problem, *Proceedings of the International Conference on Genetic Algorithms*, pp. 154-159, London

Gup, T. (1997). Technology and the end of serendipity. *The Chronicle of Higher Education*, 44, (November 21), pp. A52

Herlocker, J.; Konstan, J.; Terveen, L. & Riedl, J. (2004). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22, 1, pp. 5-53, ISSN: 1046-8188

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 95)*, pp. 1137-1145, Montréal, Morgan Kaufmann

Lawler, E. L.; Lenstra, J. K.; Rinnooy Kan, A. & Shimoys, D. (1985). *The Traveling Salesman Problem*, John Wiley & Sons, ISBN: 0471904139

Little, J.; Murty, K.; Sweeney, D. & Karel, C. (1963). An algorithm for the traveling salesman problem. *Operations Research*, 11, 6, (November-December 1963), pp. 972-989, ISSN 0030-364X

McNee, S. M.; Riedl, J. & Konstan, J. A. (2006). Being accurate is not enough: how accuracy metrics have hurt recommender systems, *Proceedings of the 2006 Conference on Human Factors in Computing Systems (CHI 2006)*, pp. 1097-1101, Montréal, ACM

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38, 11, pp. 39-41, ISSN:0001-0782

Mladenic, D. (1999). Text-learning and related intelligent agents: a survey. *IEEE Intelligent Systems*, 14, 4, (July 1999), pp. 44-54, ISSN:1541-1672

Papadimitriou, C. H. (1977). The Euclidean Traveling Salesman Problem is NP-Complete. *Theoretical Computer Science*, 4, 3, (June 1977), pp. 237-244, ISSN: 0304-3975

Roberts, R. M. (1989). *Serendipity: Accidental Discoveries in Science*, John Wiley & Sons, Inc, ISBN: 0471602035, New York

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34, 1, (March 2002), pp. 1-47, ISSN: 0360-0300

Semeraro, G.; Degemmis, M.; Lops, P. & Basile, P. (2007). Combining Learning and Word Sense Disambiguation for Intelligent User Profiling, *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pp. 2856-2861, Hyderabad - India, January 6-12, 2007

Shapira, B.; Shoval, P. & Hanani, U. (1997). Stereotypes in information filtering systems. *Information Processing and Management*, 33, 3, (May 1997), pp. 273-287, ISSN: 0306-4573

Toms G., E. (2000). Serendipitous Information Retrieval, *Proceedings of DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries*, Zurich - Switzerland

van Andel, P. (1994). Anatomy of the Unsought Finding. Serendipity: Orgin, History, Domains, Traditions, Appearances, Patterns and Programmability. *The British Journal for the Philosophy of Science*, 45, 2, pp. 631-648, ISSN 1464-3537 (Online) ISSN 0007-0882 (Print)

Wong, R. (1980). Integer programming formulations of the traveling salesman problem, *Proceedings of the IEEE International Conference of Circuits and Computers*, pp. 149-152