Augmenting Points of Interest Recommendations with Music

Marius Kaminskas*   Francesco Ricci†

Abstract
Recommender systems are personalized information search and decision support tools that can selectively retrieve from large information sources personalized set of items (e.g. news, CD descriptions, or services) that suit the particular preferences of a user. Recommender systems are normally limited in providing recommendations for just one type of information items. But, recently the notion of cross-domain recommender systems has been introduced to denote applications that can reuse knowledge about the users, which is derived in one domain, to provide recommendations in another different domain. A particular kind of cross domain recommendation task consists of selecting simultaneously two items in two different domains and recommending them together because they fit the user preferences and also they fit well together. In this work we show that given a personalized recommendation of points of interests (POIs), the user satisfaction for the itinerary composed by these POIs can be increased by enriching the itinerary presentation with music tracks that match the user's profile and are coherent with the POIs. We present the results of an online experiment where alternative approaches for matching POIs and music, based on tagging and text matching, have been tested with users.

Keywords: recommender system, mobile service, travel planning, consumer behavior

1 Introduction.

Recommender Systems (RS) are personalized information search and decision support tools that can selectively retrieve from large information sources, such as content management systems, or databases or the web, a well selected set of information items (e.g. news, CD descriptions, or services) that suit the particular preferences and constraints of a user [5, 8]. These preferences could be stored in various formats, forming the user's profile, and are either compared with descriptions of items (content-based approach) or with the profiles of other users (collaborative approach).

Recommender systems are generally limited to providing recommendations for just one type of information items, e.g., either movies, or CDs, or travels. But recently, there have been some attempts to combine information from different domains, introducing the notion of cross-domain recommender systems. These systems can reuse knowledge about the users, which is derived in one domain, e.g., CDs, to provide recommendations in another different domain, e.g., movies [6]. Overall, most of the current research concerning cross-domain recommender systems is focused on generating cross-domain user profiles thus overcoming data scarcity problem and achieving more precise user-to-user similarity measures. The target recommended items, however, typically belong to one type of objects.

In this paper we focus on a particular kind of cross-domain recommendation task, not yet well studied in the literature, where we try to select simultaneously two items in two different domains and to recommend them together because they fit the user preferences and they fit well together. We argue that this type of suggestions can be very useful in tourism information systems, and in more general multimedia recommender systems. Consider for instance a person who is sightseeing a city. The tourist may use a cross-domain recommender system running on her mobile device. This system offers the user a walking itinerary and, while the user is visiting the suggested points of interest (POIs), it plays a soundtrack which matches the visited POIs and enhances the experience. For example, a user visiting a Gothic cathedral might hear a classical composition by J.S. Bach, or during a visit to a lively market square the user might be offered an amusing folk-rock tune.

In this paper we show that user satisfaction for an itinerary can be increased by matching music to the points of interest in the itinerary. Furthermore, we hypothesize that simultaneously searching for and matching POIs and music tracks in order to offer the best combinations may provide as good results as independently selecting a personalized set of POIs and then matching music tracks to the POIs. In other words, not only the music can be matched to a pre-selected itinerary, but also the discovered relations between music and a place can influence the itinerary selection process and to a certain extent substitute the POI personalization.

The main research challenge we have faced is related to the complex relations between the two domains: location and music. Some previous attempts of relating music to physical surroundings of the user include research on context-aware systems that use the information about the user's surroundings (time, weather etc.) to infer user's music preference [15, 16, 17]. Other works

*Free University of Bozen-Bolzano, Italy.
†Free University of Bozen-Bolzano, Italy.

describe the procedure of projecting different domains to a common space, e.g. projecting text and music to a mood taxonomy. However, it is still unclear how (and which) properties of one domain relate to the properties of the other [13, 9].

In our approach, we have explored the usage of tags and textual descriptions as a way for matching items from different domains. The matching techniques are described in detail in Section 3. We have implemented three ways of combining music and POIs in a single recommendation session:

1. ranking the POIs and subsequently assigning a music track to each POI, without matching POI and music;
2. ranking the POIs and subsequently assigning the best matching music track to each POI;
3. ranking pairs of POIs and music tracks based on the matching score (similarity of music and POI) and then offering to the user the top-ranked pairs.

For evaluating the effectiveness of our approach we have designed a web application that is described in detail in Section 2.

In summary, in this paper we address a new and not yet fully developed research area in multimedia recommender systems. We present a novel approach for providing POI recommendations (text and images) together with music recommendations. We will show that such joined recommendations can increase the user satisfaction with recommended POI itineraries.

The rest of this paper is structured as follows. In Section 2 we describe the web application that we developed in order to test our research hypothesis and the main architecture of the cross-domain recommendation process. In Section 3 we illustrate the details of the POI and music personalization and matching procedures. In Section 4 we describe the outcome of the user study that we conducted, and finally in Section 5 we draw conclusions and point out some future work.

2 General Approach and Web Application.

In order to perform the validation of our approach we have implemented a web application which is aimed at offering the user two itinerary recommendations for the city of Bolzano (Italy) and its surroundings. Each itinerary consists of a sightseeing route and a soundtrack. An itinerary is composed of three points of interest (POIs) inside and around the city (monuments, buildings, nature parks etc.). Each POI has a profile (more on this later) which is stored in the system. The application also stores a database of music tracks belonging to two genres - classical and rock.

For evaluation purposes the system computes two recommended itineraries using two (randomly selected) among three different modes (as we mentioned in the Introduction and as will be described in detail in Section 3). The users are not aware of the different modes used in the recommendation process they are just offered with two itineraries. After viewing the two recommended itineraries the users are asked to choose the preferred one. This will allow us to evaluate the impact of matching music and location on the quality of recommendations.

2.1 The Evaluation Application. We will now describe in detail the main stages of the user interaction with the system that was designed to test our research hypothesis. The human/computer interaction consists of three main parts: filling out the profile information (registration), viewing two recommended itineraries and providing a feedback, i.e., choosing a preferred itinerary.

Before using the system for the first time the users must register. The registration form requests some profile information. The users must enter their age, gender, country, preferred music genre, sightseeing interests and the level of knowledge about Bolzano city and its surroundings.

Then the user can start requesting the itinerary. She should enter her ephemeral preferences and then two itineraries are calculated. The user's ephemeral preferences are entered in the starting page. The user is asked to choose a few tags that describe the POIs that she would like to see. The tags are chosen from a finite set of tags that was previously used to tag both POIs and music tracks. POIs and music tracks were tagged by the users in the data collection phase that is described later.

After selecting the preferences for POIs, the user is given a first recommended itinerary. The suggested sequence of POIs is displayed, one POI after the other, and the accompanying music tracks are played. Having viewed the first itinerary, the user is offered a second itinerary that satisfies the same ephemeral preferences but is computed with a different version of the recommendation algorithm.

Finally, after viewing the two itineraries the user is asked to state a preference by choosing the itinerary that she liked the best. The user is shown a summary description of the two itineraries and is offered the option to replay any of the two, in case the user wants to review the recommendations before making the choice (see Figure 1). The user's choice is stored in the database and used for evaluating the performance of different recommendation algorithms.

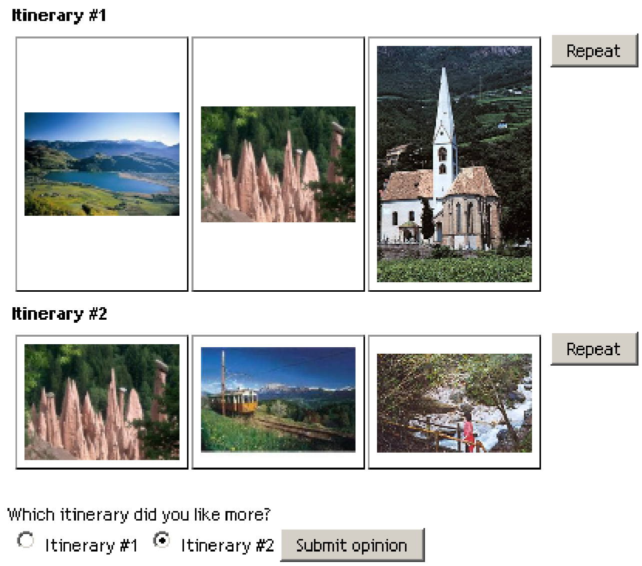

Figure 1: The evaluation page

2.2 The Recommender System Architecture. Figure 2 shows the general logical architecture of the recommender system and the technologies that we used. The system uses profile information for standard item personalization as well as music-to-POI similarity for combining music and POI in a single recommendation. The main components are: the *user profile*, the *POI profile*, the *music profile* and the recommendation algorithm that consists of *POI ranking, music filtering* and *music-to-POI similarity* computation. We will now describe these components in detail.

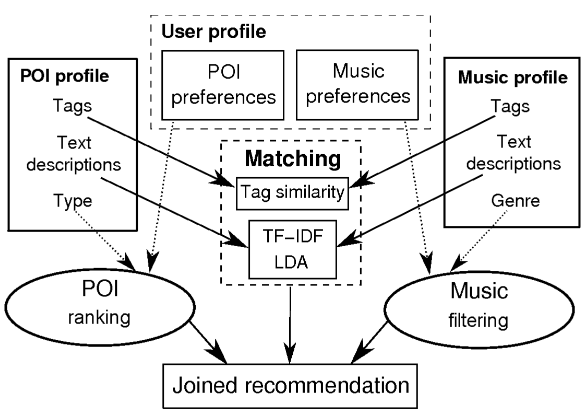

Figure 2: Model of the system

The user profile contains the basic music and sightseeing preferences of the user. The *sightseeing preferences* are used for ranking the POIs. The user can choose any of the following general categories as her preferences: art, architecture, history and nature. These four categories roughly correspond to the types of POIs stored in our database. The objects include castles, museums, monuments, churches and nature objects. Another part of the user profile contains his/her *preferred music genre*. In our application we used just two genres: classical and rock. Hence a user preferring classical (rock) will receive only recommendations with soundtracks from the classical (rock) repertoire.

Each music track is described by a music profile. The music profile contains: the title of the track, the description of the track, the genre of the track and a set of tags describing the music item. Table 1 shows an example of such a music profile. The music database, which was used in the experiments, consists of a set of manually selected music tracks (around 20 tracks per genre). The descriptions of classical music tracks have been taken from a dedicated web site (Classical Notes [11]), while the descriptions of rock music tracks were taken from Wikipedia [4]. These descriptions are used for measuring the text-to-text similarity between music tracks and POIs. The tags describing each music item have been assigned to them manually (see later). Like text descriptions, these tags are used for measuring similarity between music tracks and POIs.

Finally, the POIs are described by a POI profile containing: the name of the point of interest, the description the POI, a set of tags describing the POI, and the categories of the POI. The type of a POI is the combination of any of the four categories: art, architecture, history, nature. Table 1 gives an example of POI profile. The points of interest have been manually selected as their number around the city is not large (up to 60 POIs). The descriptions of the POIs have been taken from the tourism web-sites [2, 1, 3] and are used for measuring the text-to-text similarity between POIs and music tracks.

The categories of each POI have been manually assigned to them based on the characteristics of the POI, e.g., a lake would represent *nature*, a modern art museum is *art*, castle ruins are *architecture and history*, a fully furnished castle is *art, architecture and history*, etc.

The tags describing each point of interest have been assigned to them manually in the same way as for the music items - through a separate tagging application that is described later. The tags describing both music tracks and POIs are taken from the same set of adjectives (see table 2).

Table 1: Example of music and POI profiles

Music profile	**POI profile**
Name: Wagner - Tristan und Isolde	*Name*: Roncolo Castle
Genre: Classical	*Type*: Art, Architecture, History
Description: Tristan und Isolde (Tristan and Isolde) is an opera in three acts by Richard Wagner to a German libretto by the composer, based largely on the romance by Gottfried von Strazburg...	*Description*: First mentioned in 1237, Runkelstein castle is situated in a picturesque position on a massive mound of porphyry rock at the entrance to the Sarntal valley. It has become famous for its superb secular fresco cycles...
Tags: Beautiful(2), Big(1), Bright(1), Calm(1), Clear(1), Cold(1), Colorful(1), Dark(2), Fast(1), Gentle(1), Heavy(2), Melodic(4), Modern(1), Mysterious(2), Old(1), Pleasant(1), Powerful(1), Sad(3), Silent(2), Slow(2), Warm(2)	*Tags*: Beautiful(1), Big(2), Calm(1), Cold(1), Dark(1), Dull(1), Gentle(1), Happy(1), Mysterious(5), Narrow(1), Old(3), Pleasant(1), Powerful(2), Scary(1), Silent(1), Slow(1), Ugly(1)

Since in this work we are dealing with items belonging to different domains, one of the main difficulties we encountered was to identify a way to compute similarity between these items. The key task was to find a common representation of items that would allow to match the two. After reviewing different possibilities we decided to use, in addition to text-based descriptions of the items, a common set of tags such that they can provide a link between music and location items. We have chosen the tags in the "List of Adjectives in American English" [14]. 38 adjectives were selected in such a way that each adjective could describe both location and a music track. Table 2 lists the adjectives used for tagging.

Table 2: The set of tags for describing music and POI

Sad	Happy	Wide	Narrow
Scary	Amusing	Flat	Uneven
Angry	Calm	Dark	Bright
Embarrassing	Pleasant	Dull	Colorful
Mysterious	Clear	Powerful	Feeble
Heavy	Light	Ugly	Beautiful
Fast	Slow	Cold	Warm
Big	Small	Harsh	Gentle
Loud	Silent	Noisy	Melodic
Old	Modern		

As we mentioned already, the POIs and music items were tagged manually. In order to efficiently tag all the items in our database we have implemented a simple web application that allowed the users to tag POIs and music tracks. The application was used by around 30 people. On average, 11 tags were assigned to each POI and 13 tags - to each music track. Some more statistical information on the tags are provided later in Section 4.

The tags assigned to POIs and music tracks by the users have not been post-processed. As a result, profiles of some items may contain contradicting tags. In the example in table 1 we see tags "Bright" and "Dark" assigned to the same item. This is natural, since different users often have different perception of the same music track or POI. We observe that such contradictions may be resolved, e.g., by checking the frequency of the contradicting tags and removing the tag which was chosen by less users, or by simply discarding contradicting tags that have the frequency lower than a certain threshold.

The tagging process was conducted some months before the itinerary recommendation. The user was offered a randomly selected POI item from our database of POIs (containing 57 POIs in total). After tagging the POI, the user was given a randomly selected music track from or database (containing 21 classical and 20 rock tracks) which was followed by another random POI. The sequence continued until the user decided to logout or tagged all the items in our database. The users could choose the genre of music tracks they want to tag. This was done to avoid forcing the users to tag music they do not like.

3 The Recommendation Process.

In this section the recommendation algorithms are described in detail. In this work we focus on three approaches of combining music and POIs which we will describe later. We start with describing music filtering and POI ranking since they are used in all approaches.

The goal of both music filtering and POI ranking steps is to filter the full set of POIs and music tracks before the two types of items can be paired by one of the POI and music recommendation algorithms. In fact, in all the three algorithms the recommendation process starts with the input of the user's ephemeral preferences, which are described by the tags entered by the user, and specifying the general type of POIs that she would like to see (see Section 2.1). Based on these ephemeral preferences and the user's profile data the ranking of the POIs and the filtering of music tracks is initially done separately.

Since we are not interested in the best quality single-domain recommendations, the music filtering approach is rather simple: filtering the music tracks by the user's preferred genre. POI ranking is more complex, and consists of measuring the overlap between the type of POI and the user's interests as well as tags entered by the user and tags stored in POI profile.

The POI ranking algorithm takes as an input the user's profile data, user's ephemeral preferences, the set of all POIs and the number of POIs to return (N). The result of the algorithm is the top N ranked POIs. The user's sightseeing preferences are read from the profile data and stored as $userInterests$. This information is represented as a subset of the set {art, architecture, history, nature}, e.g. a user's interests might be {history, nature}. The user's ephemeral preferences are stored as $userTags$ and may include any of the mentioned repertory of tags. For instance, a user's ephemeral preferences might be {Bright, Calm, Clear}. We then loop through the set of all POIs in order to compute the score of each POI. Only those POIs, that have at least one common element with the user's sightseeing preferences contained in $userInterests$ are considered for recommendation. The tags describing a POI are read from the POI profile and stored as $POItags$. This information is represented as a bag of words where tags may be repeated (if assigned to a POI by different users), e.g. $POItags$ might store {sad, sad, sad, cold}. The algorithm then computes the intersection of the sets $userTags$ and $POItags$. The cardinality of the intersection is stored as the score of POI. Note that we take into account the repeating tags in POI profile, i.e. an intersection of the POI tags {sad, sad, sad, cold} with user's ephemeral preferences {sad, cold, dark} would give an intersection of 4 elements: {sad, sad, sad, cold}. This is done under the assumption that repeating tags are more important since more than one user associated a POI with these tags. Therefore, the presence of such important tags should be reflected in the POI score. Furthermore, the algorithm checks the POI profile for tags assigned to the POI by the current user. If such tags exist and they are present in the intersection described above, the POI score is increased by the number of such tags. This is done in order to boost the score of those POIs that the user considered as relevant. After looping through all the POIs in the database, the algorithm sorts the ranked list according to the score of POIs and returns the top N POIs.

The output of both music filtering and POI ranking processes, i.e., a set of music tracks and a ranked list of POIs, is then combined in different ways, depending on the approach of combining music and POIs. Let us now detail the three approaches in turn.

- **POI rank.** In this approach, we assign to each of the top N POIs (from the ranked list of POIs) one music track chosen randomly from the set of tracks belonging to the preferred genre of the user. In this way the selected music is liked by the user but is not adapted to the POI. The order of the recommended POIs is then determined by POI rank.

- **POI rank + music matching.** In this approach, after obtaining the ranked list of POIs, we compute the POI-to-music similarity between the selected POIs and the filtered music items. We then assign the best-matching music track to each of the top N POIs. The music-to-POI similarity computation is described in Section 3.1. Here, the order of recommended POIs remains identical to the first approach. Only the recommended music is different.

- **Music similarity rank.** In this approach, we do not simply assign music to the existing ranking of POIs, but rather we combine the POI rank with the music-to-POI similarity score to produce a new ranking for the pairs of POIs and music tracks. The similarity score computation is described later in Section 3.1.

  Based on the music-to-POI similarity score, as well as the POI ranking score, we generate a new ranked list of POI-music pairs. We then recommend the top N pairs. The POI-music pair score is computed using this formula:

  $$score = (0.7 * POIpers) + (0.3 * similarity) \quad (3.1)$$

Here $POIpers$ is the POI score obtained from the POI ranking algorithm; and *similarity* is the music-to-POI similarity.

We observe that the weights in formula 3.1 have been determined after some tests. We initially assigned more weight to the *similarity* parameter, since testing the music-to-POI similarity effectiveness is one of the major goals of this research. Subsequently, more weight was shifted toward the $POIpers$ parameter and the results were compared.

3.1 Computing music-to-POI Similarity. In this section the algorithm for matching a music item with a point of interest is described in detail. The music-to-POI similarity scores are stored in a matrix. This matrix is computed off-line since it depends only on static item profiles. The similarity matrix S is a P x M matrix where P is the number of POIs in the database and M is the number of music tracks in the database. Each element of the matrix S_{ij} represents the similarity of music track j to POI i . Each similarity value is computed based on the profiles of the POI and the music track as follows:

$$similarity = (0.2 * tagOverlap) + (0.8 * textSc) \tag{3.2}$$

here *tagOverlap* is the cardinality of the intersection of the tags in the POI profile the tags in the music profile. *textSc* is computed as follows:

(3.3)
$$textSc = \begin{cases} ldaSc, & \text{if } tfSc < 0.1 \\ (0.2 * ldaSc) + (0.8 * tfSc), & \text{if } tfSc \geqslant 0.1 \end{cases}$$

Due to lack of time, we were not able to test the effect of different weights in this formula. Our aim was to give more weight to $tfSc$ in the second condition (the reasons for this are discussed later).

In the above formula *ldaSc* is the similarity of text descriptions in item profiles calculated using the Latent Dirichlet Allocation *LDA* method [7]. Using the *LDA* method, each document in a corpus can be represented as a distribution of a certain number of topics. In our work, we have used all the descriptions of POIs and music tracks as the document corpus. The number of topics is the main input parameter of *LDA* algorithm. In order to choose the number of topics we ran experiments with *LDA* on a corpus of documents for which relations were clear. Our tests showed that the algorithm is able to recognize similar documents better when the topic number was set to 100. However, it must be said that more thorough testing needs to be performed in order to fully evaluate the impact of this parameter on the results.

Using the *LDA* method, each document can be represented as a vector of topic probabilities:

$$d = (p_1, p_2, ..., p_{100})$$

where p_i is the probability that the i^{th} topic is found in document d ($p_i \in [0, 1]$). The similarity between two text documents is then computed as the cosine distance between the two document vectors $d_1 = (p_1^1, ..., p_{100}^1)$ and $d_2 = (p_1^2, ..., p_{100}^2)$.

Besides, $tfSc$ is the similarity of text descriptions in item profiles computed using the classical *TF-IDF* method [18]. In our work, we used all the descriptions of music and POIs as the *TF-IDF* document corpus. Each document in the corpus can be represented by a term vector:

$$d = (t_1, t_2, ..., t_n)$$

where t_i is the weight of i^{th} term in document d and n is the number of distinct *significant* terms found in the corpus. A term is considered *significant* if it has a sufficiently high (greater than 0.01) weight in at least one document of the corpus. Using the significance threshold of 0.01 allowed us to decrease the size of term vector from 8333 (the number of *all* distinct terms in the corpus) to 5595.

The similarity between two text documents is then computed as the cosine distance between the two document vectors $d_1 = (t_1^1, ..., t_n^1)$ and $d_2 = (t_1^2, ..., t_n^2)$.

Experiments have shown that the two text processing algorithms (*LDA* and *TF-IDF*) produce different results depending on the type of text documents. *TF-IDF* can give good results when two documents are linked by highly weighted keywords. But, such situation is not common in the dataset used in this research due to different nature of the described objects. POIs and music descriptions typically lack direct links between themselves. An example of such link can be found in the descriptions of "Roncolo Castle" and the classical music composition "Tristan und Isolde" by Richard Wagner. Since Roncolo castle has some medieval frescoes depicting scenes from "Tristan and Isolde" story, the *TF-IDF* algorithm scores "Tristan und Isolde" high against "Roncolo Castle".

However, in most of the cases, the text descriptions of items lack such overlapping keywords. In such cases *LDA* algorithm performs better than *TF-IDF*. Therefore, the way to compute *textSc* in 3.3 formula is chosen based on the threshold of $tfSc$ value. The threshold was chosen based on the results achieved using *TF-IDF* algorithm with our data. Due to different vocabularies of music and POI description texts, the

TF-IDF similarity values for our data typically range from 0 to 0.06. With current data, only one POI-music pair scores more than 0.1 (the previously mentioned "*Roncolo Castle*" and "*Tristan und Isolde*"). Based on these observations we assume that if *TF-IDF* similarity is greater than 0.1, it is likely that linking keywords have been identified. Otherwise, $tfSc$ will not have an influence on the result.

The weights of 3.2 have been established empirically. Due to lack of time, only limited number of experiments with different weights could be performed. We started the tests by assigning more weight to *textSc*. The results showed no significant difference in similarity results when shifting more weight to *tagOverlap*. Therefore, the initially assigned weights have been left as our choice.

4 Evaluation.

In order to evaluate the different versions of the recommendation algorithm, as we mentioned already, we included two itineraries in a single recommendation session and asked the users to choose the itinerary they liked the better (see Section 2 for details on system usage).

Each itinerary was computed using one, randomly selected, version of the recommendation algorithm out of the three different versions (see Section 3 for details on the three versions). Therefore, a single recommendation (and evaluation) session allowed us to compare two of the three versions. While using the system, the users were unaware of the different versions of itinerary computation.

The users participated in the evaluation on a voluntary base. Invitations to participate have been sent to research colleagues as well as readers of the user-modeling mailing list (um@unito.it).

A total of 53 users used the system. 13 users used the system 3 times and more. 40 users used the system once or twice. In total, 115 evaluation sessions were performed. We note that a few users were also previously tagging the items. We believe that this could not cause a bias effect in the evaluation since the tagging was done more than 1 month before the evaluation. Moreover, the tagging process involved viewing random sequences of items and the users were not required to match any items.

The goal of the evaluation was to compare the user satisfaction with different recommendation techniques and to analyze the effect of music matching on the quality of POI recommendations.

We now illustrate the user choices when evaluating the different versions of the recommendation algorithm. We compared the number of times each recommendation algorithm was preferred by the users in all the evaluations. Furthermore, we compared, pair by pair, the number of time a version was preferred to another.

Figure 3 shows the selection probability of the three algorithms considering all session data as well as for session data where only classical and only rock music were preferred by the users. Selection probability is estimated as the ratio of times an itinerary with a soundtrack generated by one method was selected over the total number of times it was offered in one of the two suggested itineraries in a recommendation session. In this figure, *POI rank*, *POI rank + matching*, and *music similarity rank* are the different versions of recommendation algorithm.

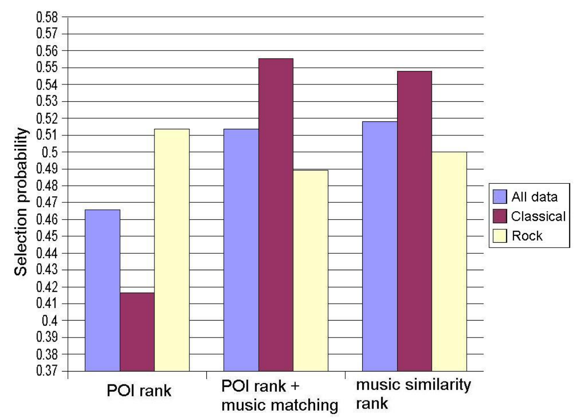

Figure 3: Selection probability of different algorithms

These results, for all session data, show that both *POI rank + matching* and *music similarity rank* approaches are preferred more often than the recommendations where music is assigned to POI without matching. This data supports our main hypothesis that the process of music matching to POI is beneficial and produces more satisfying recommendations. We note that even in the simpler algorithm, *POI rank*, the played music was selected among tracks of the user's preferred genre. So the played music was among the preferred type in all the three cases. Moreover, the testers were not aware that we were measuring their reaction to the matching of music to POIs, they were just supposed to select the itinerary they preferred the best.

The results do not show a clear preference of the users for one of the two approaches: *music similarity rank* and *POI rank + matching*. In fact, these approaches have similar selection frequency. While in the *POI rank + matching* approach the recommended itineraries are primarily composed by ranking the POIs and only then assigning music track to each POI, the *music similarity rank* approach focuses rather on recommending pairs where POI and music are highly matching to each other. Therefore, we conclude that selecting highly matching pairs of POI and music may, to some extent, compensate the lack of a precise POI personalization.

The results for session data collected separately for each music genre show that classical music listeners more clearly preferred the advanced recommendation techniques. It is clear in these results that when rock music was the preferred genre of the user no clearly preferred approach can be identified and our research hypothesis cannot be proved. This result confirms that even the simpler approach, i.e., assigning a generic track in the user's preferred tracks can be suitable under certain conditions, and that this baseline can also be used for this task.

But, these results suggest that the effectiveness of the two proposed approaches for matching music with POI depends on the genre of music preferred by the system users, and on the data acquired from the users that prefer classical music. Hence, we have further analyzed the user choices for algorithm pairs involving classical music. Table 3 shows the user selection frequency for those evaluation sessions, where classical music was chosen by the users as the preferred genre.

Table 3: Selection frequency when choosing between different approaches (with classical music)

	POI rank	POI rank + music matching	music similarity rank
POI rank *VS.* POI rank + music matching	7	**9**	
POI rank *VS.* music similarity rank	8		**12**
POI rank + music matching *VS.* music similarity rank		**6**	**5**

These results show an advantage of the advanced recommendation techniques over music and POI combination without matching; in more sessions the advanced techniques were those used in the preferred itinerary. Note especially that the *music similarity rank* approach is preferred more often than combination of music and POI without matching. *Chi-squared* test for pairs where *POI rank* itineraries were offered to the users together with *POI rank + matching* or with *music similarity rank* itineraries showed approximately 60% support to our hypothesis that both advanced recommendation techniques perform better than music and POI combination without matching.

However, the amount of evaluation data submitted by classical music listeners is lower than the data involving rock music. Out of 115 evaluation sessions 47 involved itineraries with classical music soundtracks. This can be explained by lower popularity of classical music among the users of our system. The insufficient number of evaluations does not allow us to draw any conclusion about user preferences when comparing pairs of approaches ("*POI rank + matching*" vs. "*POI rank*" and "*POI rank + matching*" vs. "*music similarity rank*" both have small number of evaluations). Therefore, it remains the task of future work to perform a more extensive evaluation.

The dependence of the algorithm performance on the music genre could be explained by the fact that classical music is much more diverse (in tone, melody, rhythm, mood and tempo changes) compared to rock music. Therefore, classical tracks can be easier for the users to tag with a wider variety of tags which makes the matching procedure more efficient.

We have therefore checked the tag distribution for music items in our database. Figure 4 shows the distribution of tags assigned to music of different genres by the users.

We computed the probability of each tag usage as $\frac{tags}{totalTags}$, where $tags$ is the number of times a tag was assigned to an item and $totalTags$ is the total number of tags assigned to the items in a collection (either classical or rock).

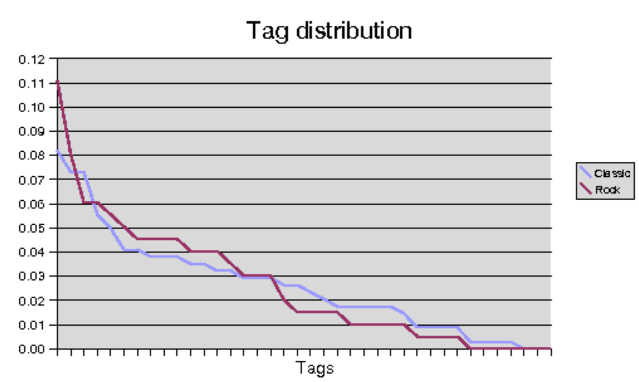

Figure 4: Tags usage probability among the different music genres

The figure 4 shows that the tag distribution of the classical music tracks is more uniform (flat). In other words, the users used more diverse tags while tagging classical music tracks. Therefore computing the tag-based similarity for classical tracks is more effective than for rock tracks.

Interestingly, rock music tracks also have less tags compared to classical music tracks. In fact, during the tagging process rock music received less tags compared to classical music tracks (although similar number of users were tagging both types of music). We have collected on average 16.24 tags for each classical track and 9.9 tags for each rock track in our database. Moreover, during a single tagging session, a user tagging rock music assigned to a track an average of 4.3 tags. Meanwhile, a classical music listener assigned to a track an average of 5.68 tags. Conversely, we have collected on average 11.18 tags for each POI in our database. A user tagging POIs assigned an average of 3.79 tags to a POI.

Another issue that could influence the results of the evaluation is related to the descriptions of the tracks stored in our database and used for music-to-POI similarity computations. While the descriptions of rock music tracks typically contain facts about the recording, the band, process of composing the track etc., classical music tracks are described in much more diverse way, with strong focus on images created by the music, the mood of the piece etc. We believe that this makes mining for music relations with a POI description more effective. This assumption is supported by the fact that classical music tracks typically get higher similarity scores using our text similarity computation methods.

The uneven distribution of tags and the different quality of text descriptions among the different music genres had an impact on music-to-POI similarity computation results. In fact, in the music-to-POI similarity matrix, which contains 2337 elements in total, the first 21 best-matching pairs of POIs and music tracks contain classical music tracks. This result shows that rock music might require different matching procedure.

5 Conclusions and Future Work.

In this work we have studied a new topic in the area of multimedia and cross-domain recommender systems. Our main research hypothesis was that augmenting recommendations of points of interest (POIs) by recommending music tracks together could improve the user satisfaction, and can influence the POI selection process, compared to a simpler POI ranking process that selects the POIs only considering the specific preferences of the user for POIs.

Different ways of combining music tracks and POIs in a joined recommendation have been implemented in order to evaluate the impact that the matching of music to POIs has on the user satisfaction for the recommended itineraries. We hypothesized that: a) user satisfaction can be increased by accompanying each POI with music tracks that are coherent with the user's profile and the POI; and b) simultaneously selecting best-matching pairs of POI and music can provide more satisfying recommendations compared to the approach where first the POIs are selected and then music is matched to each POI.

The results of our system evaluation show that users more often prefer the itineraries generated using music matching to POI techniques. Therefore, we conclude that music matching has positive effect on user satisfaction with the joined music and POI recommendations.

The results show a similar performance of the simultaneous music and POI selection compared to matching music to pre-selected POIs. This could mean that high similarity between a music track and a POI can compensate the lack of POI personalization. However, such results can also be caused by the relatively small set of music tracks, which resulted in similar music tracks recommended in the two approaches.

Furthermore, the evaluation results showed that the performance of the implemented algorithms depend on the music genre preferred by the system users. Our system offered the users two musical genres - classical and rock music. Recommendation sessions where classical music was chosen tend to give better results with the more advanced algorithms that we proposed.

Since we have addressed a new research topic, a lot of questions remain unanswered and many possibilities need to be explored. However, we believe that our results show that this topic is a promising research area, which deserves further work. We have addressed a complex problem of matching heterogeneous objects which can be useful in multimedia data mining and can lead to the development of new viable mobile services.

The future work on this research topic includes fixing the limitations of the current implementation of our approach and performing a more extensive user study with more users and more POIs and music tracks. We observe that user studies in recommender systems are not very popular and most of the validations have been performed off-line; cross validating the accuracy of the recommendation in a given data set [5, 12]. So, our work is still among the few attempts to evaluate the recommendations with real users' feedback rather than trying to reproduce the past user evaluations on recommended items.

With respect to the algorithms used, we note that the computation of the overlapping of the tags assigned to items should use normalized values for tag usage and not the absolute number of tags present in the item profile. The current implementation relies on an ad-hoc solution and can provide imprecise results when performing calculations with bigger amounts of data. Furthermore, extensive testing must be performed with different weights in the formulas in order to tune the algorithm for optimal performance and we shall consider newly proposed similarity functions for tagged objects [10].

Current music-to-POI matching techniques can be improved in several ways. For instance, one may use *TF-IDF* scores for selecting only significant tags in item profiles. We also intend to keep looking for new methods of matching music tracks and POIs as the currently used approaches (tagging and text-based similarity) have limitations related to text descriptions of the tracks and differences in music genres.

The described technique could be used when presenting information on-line (e.g. in tourism web-sites). However, it can also be applied in real-life mobile applications for personalizing the travel experience with ad hoc music. Therefore, transferring the application to mobile platform is necessary for testing the system in real-life use. Our approach can be extended to use stream data for playing music. This could greatly expand the available data and increase the possibilities of our system.

References

[1] Eppan tourism website. `http://www.eppan.com/`.

[2] The official website of south tyrol. `http://www.suedtirol.info/`.

[3] Ritten tourism website. `http://www.ritten.com/`.

[4] Wikipedia. `http://en.wikipedia.org/`.

[5] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[6] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction*, 18(3):245–286, 2008.

[7] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[8] Robin Burke. Hybrid web recommender systems. In *The Adaptive Web*, pages 377–408. Springer Berlin / Heidelberg, 2007.

[9] Rui Cai, Chao Zhang, Chong Wang, Lei Zhang, and Wei-Ying Ma. Musicsense: contextual music recommendation using emotional allocation modeling. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 553–556, New York, NY, USA, 2007. ACM.

[10] Ciro Cattuto, Dominik Benz, Andreas Hotho, and Gerd Stumme. Semantic grounding of tag relatedness in social bookmarking systems. In Amit P. Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy W. Finin, and Krishnaprasad Thirunarayan, editors, *International Semantic Web Conference*, volume 5318 of *Lecture Notes in Computer Science*, pages 615–631. Springer, 2008.

[11] Peter Gutmann. Classical notes. `http://www.classicalnotes.net/`.

[12] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transaction on Information Systems*, 22(1):5–53, January 2004.

[13] Lie Lu, D. Liu, and Hong-Jiang Zhang. Automatic mood detection and tracking of music audio signals. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(1):5–18, Jan. 2006.

[14] Paul Noll and Bernice Noll. List of adjectives in american english. `http://www.paulnoll.com/Books/Clear-English/English-adjectives-1.html`.

[15] Han-Saem Park, Ji-Oh Yoo, and Sung-Bae Cho. S.b.: A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In *FSKD 2006. LNCS (LNAI)*, pages 970–979. Springer, 2006.

[16] Anand Ranganathan and Roy H. Campbell. A middleware for context-aware agents in ubiquitous computing environments. In Markus Endler and Douglas C. Schmidt, editors, *Middleware*, volume 2672 of *Lecture Notes in Computer Science*, pages 143–161. Springer, 2003.

[17] Sasank Reddy and Jeff Mascia. Lifetrak: music in tune with your life. In *HCM '06: Proceedings of the 1st ACM international workshop on Human-centered multimedia*, pages 25–34, New York, NY, USA, 2006. ACM.

[18] Gerard Salton. *Automatic Text Porcessing: The transformation, analysis, and retrieval of information by computer*. Addison-Wesley, 1989.