A Model for Serendipitous Music Retrieval

Article · May 2012

DOI: 10.1145/2162102.2162105

3 authors, including:

Markus Schedl
Johannes Kepler University Linz

David Hauger
Johannes Kepler University Linz

Proceedings of the

2nd Workshop on Context-awareness in Retrieval and Recommendation

in Conjunction with IUI 2012
February 14, 2012
Lisbon, Portugal

Chairs

Ernesto W. De Luca - University of Applied Sciences Potsdam

Matthias Böhmer - DFKI GmbH

Alan Said - Technische Universität Berlin

Ed H. Chi - Google Inc.

www.carr-workshop.org

ICPS

Published by ACM

Chairs:
Ernesto W. De Luca
deluca@fh-potsdam.de
Matthias Böhmer
matthias.boehmer@dfki.de
Alan Said
alan.said@dai-lab.de
Ed H. Chi
chi@acm.org

Program Committee:
Omar Alonso
Hideki Asoh
Linas Baltrunas
Toine Bogers
Robin Burke
Li Chen
Karen Church
Marco Degemmis
Ido Guy
Qi He
Tim Hussein
Brijnesh-Johannes Jain
Dietmar Jannach
Alexandros Karatzoglou
Carsten Kessler
Alfred Kobsa
Antonio Krüger
Michael Kruppa
Martha Larson
Ulf Leser
Pasquale Lops
Petteri Nurmi
Till Plumbaum
Francesco Ricci
Markus Schedl
Armando Stellato
Domonkos Tikk

The Association for Computing Machinery
2 Penn Plaza, Suite 701
New York New York 10121-0701

ACM ISBN: 978-1-4503-1192-2/11/02

The 2nd Workshop on Context-awareness in Retrieval and Recommendation (CaRR 2012)

CaRR 2012

The aim of the second CaRR workshop was to invite the community to a discussion on new, creative ways to handle context-awareness. Furthermore, the workshop aimed at improving the exchange of ideas between different communities involved in research concerning human-computer interaction, machine learning, information retrieval and recommendation.

The workshop was intended for researchers working on multidisciplinary tasks who wanted to discuss problems and synergies. The focus of the workshop was on creative and collaborative approaches for context-aware retrieval and recommendation.

The participants were encouraged to address the following questions:

- What is context?
- Is context-awareness in retrieval and recommendation necessary?
- Which benefits come from context-aware retrieval and recommendation systems?
- How do user interfaces handle context?
- In what ways can context improve HCI?
- How can we combine general- and user-centric context-aware technologies?
- How should context affect the way information is presented?
- Which new means for collecting user feedback does UbiComp provide?
- What new types of items (beyond books, news and movies) are worth recommending by means of context-aware systems (e.g. places, friends, apps)?

The workshop brought together a group of researchers working on different aspects of context-awareness during the first day of the 2012 ACM Intelligent User Interfaces conference, February 14th 2012. During the workshop, six presentations were given, as well as a starting keynote by Anthony Jameson on the topic of "Roles of Context in Information Retrieval and Recommendation: A Choice and Decision Making Perspective". Authors discussed various topics related to information retrieval systems like personalizing the Web, marketing websites, matrix factorization, recommendation of points of interest, recommendation of music, and tagging-based user models.

The organizers would like to thank all the authors for contributing to CaRR 2012 and all the members of the program committee for ascertaining the scientific quality of the workshop. Additional information about the workshop is provided at the workshop website http://www.carr-workshop.org

Ernesto W. De Luca
Matthias Böhmer
Alan Said
Ed H. Chi

TABLE OF CONTENTS

Introduction

Papers

Appendix

Roles of Context in Information Retrieval and Recommendation: A Choice and Decision Making Perspective

Anthony Jameson
DFKI, German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
jameson@dfki.de

KEYNOTE ABSTRACT

Systems for information retrieval and recommendation can be seen as tools that help people make good choices and decisions: about which documents to read, which products to buy, which people to contact, …. Taking this perspective, we can exploit insights from psychological research on how people make choices and decisions – in particular, on the role played by psychologically relevant contextual factors such as current goals, mood, time pressure, and distractions. In this talk, after looking at a compact overview of the diverse psychological processes involved in choosing – ranging from choices made quickly and intuitively to deliberate decisions – we will consider some questions about the roles of context in these processes: How can context influence a person's current information need or the value that they attach to a particular item? How can context influence how such needs and evaluations are expressed or reflected in behavior (even if it doesn't influence these things themselves)? And in the light of the answers to these questions: To what extent do people actually have needs and evaluations which exist independently of the contexts in which they are reflected in behavior? Implications for the design and study of context-aware systems for retrieval and recommendation will be discussed with audience participation.

CaRR 2012, February 14, 2012, Lisbon, Portugal.

Enhancing Matrix Factorization Through Initialization for Implicit Feedback Databases

Balázs Hidasi
Gravity R&D Ltd.
balazs.hidasi@gravityrd.com
Budapest University of Technology and Economics
hidasi@tmit.bme.hu

Domonkos Tikk
Gravity R&D Ltd.
domonkos.tikk@gravityrd.com

ABSTRACT

The implicit feedback based recommendation problem—when only the user history is available but there are no ratings—is a much harder task than the explicit feedback based recommendation problem, due to the inherent uncertainty of the interpretation of such user feedbacks. Still, this practically important recommendation task received less attention and therefore there are only a few common implicit feedback based algorithms and benchmark datasets. This paper focuses on a common matrix factorization method for the implicit problem and investigates if recommendation performance can be improved by appropriate initialization of the feature vectors before training. We present a general initialization framework that preserves the similarity between entities (users/items) when creating the initial feature vectors, where similarity is defined using e.g. context or metadata information. We demonstrate how the proposed initialization framework can be coupled with MF algorithms. The efficiency of the initialization is evaluated using various context and metadata based similarity concepts on two implicit variants of the MovieLens 10M dataset and one real life implicit database. It is shown that performance gain can attain 10% improvement in recall@50 and in AUC@50.

Author Keywords

Recommender systems, Implicit feedback, Initialization, Similarity, Context information

ACM Classification Keywords

I.2.6 [Artificial Intelligence]: Learning - Parameter Learning

General Terms

Algorithms, Experimentation

CaRR 2012, February 14, 2012, Lisbon, Portugal.

INTRODUCTION

Recommender systems identify specific content that matches users' personal interests within huge content collections. The relevance of an item (the unit of content) with respect to a user is predicted by recommender algorithms; items with the highest prediction scores are displayed to the user.

A typical classification [5] divides recommender algorithms into two main approaches: the content based filtering (CBF) and the collaborative filtering (CF). Content based filtering algorithms use user metadata (e.g. demographic data) and item metadata (e.g. author, genre, etc.) and try to predict the preference of the user based on these attributes. In contrast, collaborative filtering methods do not use metadata, but only data of user–item interactions. Depending on the nature of the interactions, algorithms can also be classified into explicit and implicit feedback based methods. In the former case, users provide explicit information on their item preferences, typically in form of user ratings. In the latter case, however, users express their item preferences only implicitly, as they regularly use an online system; most typical implicit feedback types are viewing and purchasing. Obviously, implicit feedback data is less reliable as we will detail later. CF algorithms proved to be more accurate than CBF methods, if sufficient preference data is available; for a quantification of sufficiency, see e.g. [11]. If this does not hold, the so-called cold-start problem occurs.

In the last few years, latent factor based CF methods gained enhanced popularity, because they were found to be much more accurate in the Netflix Prize, a community contest launched in late 2006 that provided for a long term the largest explicit benchmark dataset (100M ratings) [2]. Latent factor methods build generalized models that intend to capture user preference. These algorithms represent each user and item as a feature vector and the rating of user u for item i is predicted as the scalar product of these vectors. Different matrix factorization (MF) methods are often used to compute these vectors, which approximate the partially known rating matrix using alternating least squares (ALS) [1], gradient descent method [19], coordinate descent method [12], conjugate gradient method [21], singular value decomposition [8],

or a probabilistic framework [16].

CF methods are able to provide accurate recommendations if enough feedback is available. In a few application areas, such as movie rental, travel applications, video streaming, users have motivation to provide ratings to obtain better service. In general, however, users of online e-commerce shops or services do not tend to provide ratings on items even if such an option is available, because (1) when purchasing they have no information on their satisfaction rate (2) they are not motivated to return later to the system to do so. In such cases, user preferences can only be inferred by interpreting user actions (also called *events*). For instance, a recommender system may consider the navigation to a particular product page as an implicit sign of preference for the item shown on that page [15]. The user history specific to items are thus considered as implicit feedback on user taste. Note that the interpretation of implicit feedback data may not necessarily reflect user satisfaction which makes the implicit feedback based preference modeling a much harder task. For instance, a purchased item could be disappointing for the user, so it might not mean a positive feedback. We can neither interpret missing navigational or purchase information as negative feedback, that is such, information is not available.

Despite its practical importance, this harder but more realistic task have been studied less. The implicit alternating least squares (iALS) method [6] is considered the seminal work in this area, which also cast the problem to a latent factor model and keeps its computational efficiency given implicit user feedback using "the implicit trick" (see Section Related work).

In this paper we examine the importance of the initialization of this iALS algorithm. We show that if the usual random or zero initialization is replaced by a similarity based version, the model performance improves significantly. We propose a matrix factorization based initialization method which integrates additional, possibly external, information sources—we performed experiments with context and metadata—to calculate the initial weights in the model. The proposed initialization methodology can be combined with arbitrary implicit feedback matrix factorization method (see e.g. [12], [21]).

The main contributions of this papers are: (1) along a simple idea we propose a general concept of initializing matrix factorization methods; (2) we propose a novel method (SimFactor) that enables to improve the quality of the initial vectors; (3) we run experiments with a large variety of initialization settings using different types of additional information sources on MovieLens 10M and on a real life implicit feedback grocery datasets.

The rest of the paper is organized as follows. *Related work* describes the iALS algorithm and presents currently used initialization approaches. The concept of our initialization methods is described in *Method*. Here we also describes the SimFactor algorithm that can approximate the similarities between entities efficiently using feature vectors. In *Results* we present the results of our experiments with different initialization methods. Finally *Conclusion* sums up this work.

RELATED WORK

We first present the iALS algorithm [6] as our experiments revolve around this matrix factorization algorithm.

We will use the following notation in this work: N is number of users, M is number of items, K denotes the number of features, R is rating matrix, P and Q are user and item feature matrices.

The implicit task is solved in iALS by a weighted matrix factorization. Instead of the R matrix, an $R^{(p)}$ (preference) matrix is constructed in a way that the (u, i) element of the matrix is 1 only if user u has at least one event on item i, otherwise 0. It is important to note that this $R^{(p)}$ matrix is dense unlike the R matrix of the explicit problem (but $R^{(p)}$ contains a lot of zero elements). A W weight matrix is also created: if the (u, i) element of $R^{(p)}$ is 0 then the (u, i) element of W is 1, otherwise it is greater than 1. The specific value can be computed based on the number and type of events between user u and item i. The weights can be computed in several ways. This decomposition of the R matrix can be interpreted as that the presence of an event (e.g. *buy*) provide more reliable information on the user preference than the absence of an event. In other words, we can be more confident in our assumption (*buy* = like) in case of positive implicit feedbacks. We model this by assigning (much) greater weight to positive implicit feedback than to negative one.

Since the $R^{(p)}$ matrix is dense, any algorithm that scales with the number of ratings can not solve this problem efficiently because the number of "implicit ratings" is $N \times M$. Given that the density of the rating matrix is usually below 1 %, the naive implementation would require several orders of magnitude more computation time compared to the explicit case, which scales linearly with the number of ratings.

In [6], an "implicit trick" for ALS is proposed to brake down the computational time. ALS approximates the matrix R as the product of two lower rank matrices, $R \approx PQ$, and performs a series of weighted linear regressions. First, matrices P and Q are initialized with random values. Then we fix matrix Q and compute each column of matrix P using weighted linear regression (minimizing $(R^{(p)}_{u,\bullet} - (P_{\bullet,u})^T Q) W^{(u)} (R^{(p)}_{u,\bullet} - (P_{\bullet,u})^T Q)^T$, where $W^{(u)}$ is a $M \times M$ diagonal matrix and $W^{(u)}_{i,i} = W_{u,i}$). Then, matrix P is fixed and the columns of Q are computed analogously.

The bottleneck in computing a column of P comes from the computation of the $QW^{(u)}Q^T$ that is naively done in $O(K^2M)$. However, $QW^{(u)}Q^T$ can be rewritten as $QQ^T + Q(W^{(u)} - I)Q^T$ (I is the identity matrix), from

which QQ^T can be precalculated. Because $(W^{(u)} - I)$ has only a few non-zero elements, the cost of computing $Q(W^{(u)} - I)Q^T$ is only $O(K^2 n_u)$ where n_u is the number of non-zero element in the uth row of $R^{(p)}$. Hence, the total cost (all N column) of the computation of P is proportional with the number of positive implicit feedback instead of number of all entries in the rating matrix.

The importance of proper initialization was recognized for some matrix factorization algorithms like the Non-negative Matrix Factorization (NMF). It was shown in [17] that a good initialization can improve the speed and accuracy of the algorithms, as it can produce faster convergence to an improved local minimum. The rich literature of NMF initialization includes centroid methods [10], spherical k-means clustering methods [23, 24] that provides low rank representation [4], SVD [3] and sum of randomly selected feature vectors [10]. It is common in all of these methods that they use the same data for initialization and for training the NMF.

In collaborative filtering algorithms, feature weights are typically initialized with small random weights [13, 20]. Certain works report on some parameterized randomization, drawing the random numbers from a normal distribution [22], or defining adjustable lower and upper bounds separately for the item and user weights [20]. To the best of our knowledge, more sophisticated initialization approaches, using external data sources have not been proposed so far.

METHOD

Most of the MF methods are iterative algorithms that are started from a random point: the item and user feature matrices are initialized randomly. After some iterations these methods converge to a local optimum that depends on the starting point. Our hypothesis is that appropriate initialization of feature vectors yields that MF methods will produce more accurate feature vectors and therefore give more accurate predictions.

When investigating the feature vectors of accurate MF models, one can observe similar items (e.g. items belonging to the same product category, or episodes of a movie series) have similar item feature vectors. This suggest that similarity-based initialization of feature vectors may result in more appropriate models. The calculation of the initial item and user feature vectors should be obviously aligned with the learning algorithm applied. To do this, first we have to define the similarity between entities (items, users), which depends on the similarity function and on the available item, user and transactional data. In this paper we use cosine similarity as similarity function for simplicity, but this can be substituted by any other similarity metric. As for the available data, we can make use of any of the following data in our experiments:

- *Item metadata vectors:* let us consider an indexed set of metadata tags, which contains all the possible tags that occur in item metadata (can be textual or categorical). The item metadata vector contains a non-zero value in the ith position if the ith tag occurs in item's metadata. One can apply various weighting schemes (e.g.: tfidf) to determine the elements of the vectors.
- *User/Item event vectors:* a user event vector of M length indicates with a non-zero values for which item the user has at least one event (analog for items).
- *User/Item context state vectors:* let us define the set of context states (C) as the possible combination of values of context variable. Here we consider only categorical context variables with finite range. For instance if we take *seasonality* as context, and a season is a week and time bands are days, then we have 7 context states. When more than one context variable is used then the context states are the Descartes-product of individual context values. I.e. if additionally we store in another context variable if the purchase was made online or offline, then we have 14 context states. Then the ith element in the user context state vector is non-zero if the user has at least one event in the ith context state (analog for items).
- *User/Item context-event vectors:* the user context-event vectors have length $C \cdot M$; each coordinate represents whether user has events on the given item in the given context state (analog for items).

Remark that most of these vectors are typically very sparse, except context state vectors with few context variables. Note that in each of the above cases, one has several choices in creating the item/user description vectors from the raw data: vectors may be binary, may contain event counters, furthermore one may apply normalization or a weighting scheme.

We assemble a matrix, D, from the appropriate input vectors (row-wise), which is referred to as the description of the items (D_I) or users (D_U). For this we select an arbitrary but single data source from the above options; e.g., we use the item context state data vectors to form D. In order to make use of the description as initial weights in a matrix factorization method, one should compress them to be compliant with the feature size of the MF model. This can be performed by any dimension reduction techniques like PCA [7], matrix factorization, auto-associative MLP [9], etc. These methods minimize the information loss at the compression and simultaneously perform noise reduction.

In this paper we use two methods for compression. The first is a simple matrix factorization, the weighted ALS, that minimizes the weighted squared error of the predictions by fixing one of the feature matrices and computing the rows of the other by using weighted linear regression. When factorizing item description, we only keep the item feature matrix after the factorization process (analog for user description), which is then readily used as initial feature vectors in the iALS algorithm.

Our starting hypothesis was that description vectors characterize well the similarities between entities. Therefore the relation of similarities (e.g. ratios, order, etc.) between original description vectors should be carried over to the compressed description vectors. Next we introduce the SimFactor compression method that is able to preserve the relations between the original similarities in a much larger extent than ALS.

SIMFACTOR ALGORITHM

As noted above, standard dimension reduction techniques may distort the system of similarities between the entities. One could design a method that keeps this property by starting from the similarity matrix of the users/items. The problem with such an approach is that it requires the precomputation of the entire similarity matrix, which is computationally very inefficient. Further, this solution does not scale well, because the matrix has to be stored in memory the sake of efficient computation. According to our test, even when sparse data structures are used for storing similarities, the calculation of the similarity matrix takes a considerable amount of time, when N or M is large.

SimFactor is a simple algorithm that compresses the description of the items while preserves the relations between the original similarities as much as possible. This method only works for similarity metrics that can be computed via the scalar product of two (transformed) vectors. The most commonly used metrics in recommendation systems like cosine similarity, adjusted cosine similarity or Pearson correlation [18] can be computed in this way. As for cosine similarity, one needs to ℓ_2-normalize the input vectors then their scalar product will be the same as the cosine similarity between the original vectors. The pseudocode for SimFactor is described in Algorithm 1 (see also Figure 1).

Algorithm 1 SimFactor

Input: D matrix that contains the item or the user description

Output: F matrix that contains the feature vectors of the items or users

procedure SimFactor(D)

1: $D' \leftarrow$ Transform(D)
2: $< X, Y > \leftarrow$ FactorizeMatrix(D')
3: $Z \leftarrow Y^T Y$
4: $< U, \Lambda > \leftarrow$ EigenDecomposition(Z)
5: $F \leftarrow$ matrix of $N_{entities} \times N_{entities}$ size
6: **for** $i = 1, \ldots, N_{entities}$ **do**
7: $\quad F_i = \sqrt{\Lambda_{i,i}} X_i U$
8: **end for**
9: **return** F

end procedure

SimFactor starts with the appropriate transformation of the description matrix (line 1; e.g. ℓ_2-normalization when using cosine similarity). Next in line 2, a matrix factorization is applied on the description, but in contrast to the method described above, both low rank matrices are kept. For the matrix factorization, arbitrary MF method can be used. Here, we applied weighted ALS.

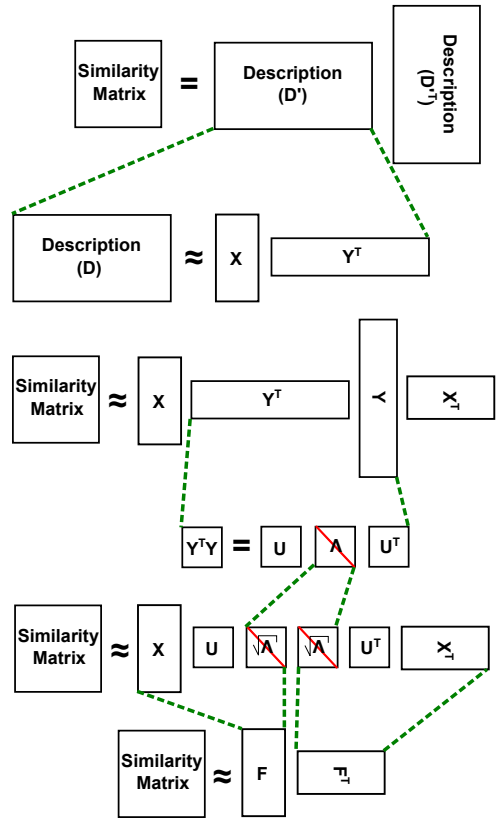


Figure 1. Concept of the matrix transformations in SimFactor

The steps performed between lines 3 and 8 are also depicted on Figure 1. The matrix of similarities (S) is the product of the transformed description matrix and its transpose ($S = D'D'^T$), while the factor matrices (output of the MF method in line 2) approximate the transformed description matrix ($D' \approx XY^T$). Therefore the similarity matrix can be approximated by $S \approx XY^TYX^T$. Y^TY is a $K \times K$ symmetric (non-singular) matrix, thus its eigen-decomposition always exists in the following form: $Y^TY = U\Lambda U^T$. U and Λ are $K \times K$ matrices, the earlier contains the eigenvectors the latter is singular and has the eigenvalues in its diagonal. Λ can be written as the product of two identical matrices denoted with $\sqrt{\Lambda}$. $\sqrt{\Lambda}$ is also diagonal and contains the square roots of the eigenvalues. At this point our approximation of the similarity matrix looks like this:

$S \approx XU\sqrt{\Lambda}\sqrt{\Lambda}U^T X^T$. Introducing the $N_{entities} \times K$ matrix $F = XU\sqrt{\Lambda}$, this can be rewritten $S \approx FF^T$.

In F, every row is a feature vector for an entity and the scalar product of the ith and j^{th} rows approximates the similarity between the corresponding entities. This way SimFactor produces low-rank feature vectors that try to preserve the original similarity values. We can use these feature vectors as the initial features in the iALS algorithm. The complexity of SimFactor—in addition to the initial transformation and matrix factorization—is $O(K^2N + K^3 + K^2M)$, where the subsequent terms correspond to the calculation Y^TY, finding the eigendecomposition and calculating $F = XU\sqrt{\Lambda}$, respectively. We found that, in practice, this cost is negligible compared to the cost of the initial matrix factorization in line 2.

RESULTS

We used two datasets for experimentation. The first dataset is the MovieLens 10M [14] that was transformed into an implicit feedback dataset. We used two different transformations: (1) keeping only the 5 star ratings and (2) keeping ratings with values 4 and above as positive feedbacks. We used the last 20 days for testing (from 08/12/2008) and the rest for training. As the available metadata is not sufficient we did not use the metadata for initialization with the MovieLens database. The second dataset contains purchase events of an online grocery store. The number of events is slightly above 6.24 million targeted on 17,000 items (of them 14,000 has at least one event). We used all but the last month's data for training and the last month for testing.

We used various data sources when creating the description matrix (see details in *Method*). For context information, we chose seasonality because the time stamp is available in almost every dataset. On seasonality we mean that we define periodicity and divide it into smaller time intervals called *time bands*. For example, hours of a day can be time bands of a day, or the weekdays and the weekend can be time bands of a week. Note that the length of the time bands does not have to be identical. The context of an event is the time band in which it happened. We used different periods and time bands and kept only the best results.

Our first experiment compared weighted ALS and SimFactor to characterize their similarity preserving capability. We draw randomly 2 times 100 000 entity pairs, calculated the original similarity values and measured the RMSE (root mean square error) of the similarity value prediction as well as the improvement in the relation of the pairs. Next, we characterized difference between similarity preserving using *ratio improvement* calculated as:

$$\frac{\sum_{i_1,i_2=1}^{100\,000} \left(s_{i_1}/s_{i_2} - s''_{i_1}/s''_{i_2}\right)^2}{\sum_{i_1,i_2=1}^{100\,000} \left(s_{i_1}/s_{i_2} - s'_{i_1}/s'_{i_2}\right)^2} - 1$$

where prime denotes the prediction of SimFactor and double prime denotes the prediction of ALS.

The results in Table 1 show that SimFactor was more accurate in every experiment. The improvement varies between 10–50%. In addition to better accuracy, SimFactor also preserves the original relations of the similarities better than the weighted ALS. The performance metrics depends greatly on the description matrix.

We used recall@50 as primary evaluation metric for the main experiments, which is the fraction of the proportion of relevant items among the top50 (ranked) recommendations for the user and the user's events in the test set. Items considered relevant to a user if the user has at least one event on that item in the test set. Recall@50 is an important measure in practical applications as the user usually sees maximum the top few items (50 items could be seen on multiple pages during a visit). We also present the area under the precision-recall curve (from $1 \ldots 50$) as a secondary metric.

The experiments started by optimizing the hyperparameters of an iALS algorithm. We used low-factor models as they can be used in practice as well. Then we run multiple experiments with different random initializations and chose the best result as the baseline. We used weighted ALS and SimFactor (that also uses a weighted ALS as its first step) to create the initial feature. Note that since iALS is an alternating method that discards the results of previous computations when calculating the feature vectors. we can not initialize both item and user features at once as one of them will be discarded in the first step. We ran multiple experiments for each input data type for the initialization and kept only the best one per input data type.

Table 2 sums the results of our experiments. The results on the grocery database are the most relevant as that database is originally implicit feedback based. The result clearly show the superiority of the SimFactor method over standard factorization as the top performing initializations used SimFactor. We observed the same when experimenting on MovieLens (5star), but on the MovieLens (4star+) SimFactor is less dominant. This can be argued by the additional noise being present in the MovieLens (4star+) datasets (note that "implicitization" was performed in a non-personal way and user may use different ratings scales).

We want to point out that the top performing methods on every dataset use contextual information for initialization. Both context state and context-event information is used amongst them, but on the grocery and MovieLens (5star) data, the context state based methods are the dominant. This suggest that context information, like seasonality, can efficiently discriminate between entities, and this can be exploited in weight initialization: Users have routines and people with similar routines are similar and might have similar taste. Similarly, different item types are typically consumed in different time

Table 1. Accuracy of the similarity prediction

Input data	Method	RMSE	RMSE Improvement	Ratio Improvement
MovieLens 10M (5 star ratings)				
Item context state	ALS	0.2055	45.79%	70.75%
	SimFactor	0.1114		
User context state	ALS	0.3345	11.66%	79.60%
	SimFactor	0.2955		
Item context-event	ALS	0.0568	22.68%	2.46%
	SimFactor	0.0439		
User context-event	ALS	0.1631	33.13%	64.43%
	SimFactor	0.1091		
Item event data	ALS	0.0593	23.98%	87.47%
	SimFactor	0.0451		
User event data	ALS	0.1285	32.03%	86.46%
	SimFactor	0.0874		
MovieLens 10M (4 star ratings and above)				
Item context state	ALS	0.2219	31.55%	74.58%
	SimFactor	0.1519		
User context state	ALS	0.2469	12.40%	61.29%
	SimFactor	0.2163		
Item context-event	ALS	0.0317	35.28%	31.82%
	SimFactor	0.0205		
User context-event	ALS	0.0835	34.64%	83.19%
	SimFactor	0.0546		
Item event data	ALS	0.0658	2.84%	73.53%
	SimFactor	0.0639		
User event data	ALS	0.1803	14.51%	37.39%
	SimFactor	0.1542		
grocery dataset				
Item context state	ALS	0.3254	52.36%	81.74%
	SimFactor	0.1550		
User context state	ALS	0.1178	10.81%	18.69%
	SimFactor	0.1050		
Item context-event	ALS	0.0314	16.86%	71.80%
	SimFactor	0.0261		
User context-event	ALS	0.1518	26.22%	62.55%
	SimFactor	0.1120		
Item event data	ALS	0.0492	13.39%	9.61%
	SimFactor	0.0427		
User event data	ALS	0.1930	48.70%	64.08%
	SimFactor	0.0990		
Item metadata	ALS	0.2709	12.38%	38.39%
	SimFactor	0.2374		

bands; for example adult programs mostly viewed late night. The distribution of the events for an entity in the time bands seems to be an efficient descriptor.

The largest improvement for the grocery dataset is 5.71% in the recall and the fifth best method improves it by 4.04%. Considering that the cost of the initialization is small and that this improvement can be translated into increased profit, we believe that the initialization of the algorithm is beneficial.

CONCLUSION

In this paper we proposed a general framework for initializing MF algorithms. Our hypothesis was that initializing item and user models with weights that reflect the similarity between entities will improve algorithm performance when compared to starting from a random state. This method also allows us to easily incorporate additional information like context or metadata information into the MF framework.

Our proposed SimFactor algorithm can efficiently implement the general idea. We found that SimFactor preserves the original similarities and their relations better than other MF methods which makes this method more suitable for our goals. The additional complexity of SimFactor is its negligible additional cost when compared to an arbitrary MF method.

Using different data sources to compute similarity can greatly affect the performance gain observed with initialization. We found that the greatest improvement can be achieved by using the context information (we experimented with seasonality). Context separates the entities more appropriately than any other information

Table 2. Results of the top10 performing initialization methods on both datasets

Data type	Method	Recall@50 to baseline	Improvement	AUC@50 to baseline	Improvement
MovieLens 10M (5 star ratings)					
Item context state	SimFactor	0.1413	10.00%	$1.4512 * 10^{-3}$	9.85%
User context state	SimFactor	0.1403	9.17%	$1.4000 * 10^{-3}$	5.98%
Item context-event	SimFactor	0.1403	9.17%	$1.3526 * 10^{-3}$	2.39%
Item context-event	MF	0.1403	9.17%	$1.3377 * 10^{-3}$	1.26%
Item context state	MF	0.1403	9.17%	$1.4596 * 10^{-3}$	10.49%
Item event data	SimFactor	0.1392	8.33%	$1.3754 * 10^{-3}$	4.11%
User context-event	MF	0.1392	8.33%	$1.3754 * 10^{-3}$	4.11%
User context state	MF	0.1392	8.33%	$1.4006 * 10^{-3}$	6.02%
User context state	SimFactor	0.1382	7.50%	$1.3899 * 10^{-3}$	5.21%
Item event data	MF	0.1381	7.50%	$1.3725 * 10^{-3}$	3.89%
Random initialization (baseline)		0.1285	N/A	$1.3211 * 10^{-3}$	N/A
MovieLens 10M (4 star ratings and above)					
User context-event	SimFactor	0.08636	5.67%	$1.1612 * 10^{-3}$	5.24%
Item context-event	MF	0.08574	4.91%	$1.1537 * 10^{-3}$	4.56%
User context-event	MF	0.08559	4.73%	$1.1672 * 10^{-3}$	5.79%
Item context state	MF	0.08528	4.35%	$1.1671 * 10^{-3}$	5.78%
Item context-event	SimFactor	0.08512	4.16%	$1.1486 * 10^{-3}$	4.09%
User context state	MF	0.08497	3.97%	$1.1390 * 10^{-3}$	3.23%
User context state	SimFactor	0.08466	3.59%	$1.1345 * 10^{-3}$	2.82%
User event data	MF	0.08451	3.40%	$1.1670 * 10^{-3}$	5.77%
Item event data	SimFactor	0.08435	3.21%	$1.1265 * 10^{-3}$	2.10%
User event data	SimFactor	0.08435	3.21%	$1.1294 * 10^{-3}$	2.35%
Random initialization (baseline)		0.08172	N/A	$1.1034 * 10^{-3}$	N/A
grocery dataset					
User context state	SimFactor	0.1508	5.71%	$1.0692 * 10^{-2}$	10.19%
User context state	MF	0.1496	4.88%	$1.0675 * 10^{-2}$	10.01%
User context-event	SimFactor	0.1488	4.30%	$1.0367 * 10^{-2}$	6.83%
User event data	SimFactor	0.1485	4.12%	$1.0626 * 10^{-2}$	9.50%
User context-event	MF	0.1484	4.04%	$1.0408 * 10^{-2}$	7.25%
Item event data	SimFactor	0.1474	3.29%	$1.0169 * 10^{-2}$	4.79%
Item metadata	MF	0.1472	3.15%	$1.0319 * 10^{-2}$	6.34%
Item context-event	MF	0.1469	2.97%	$1.0280 * 10^{-2}$	5.94%
Item context state	MF	0.1465	2.67%	$1.0222 * 10^{-2}$	5.35%
Item context state	SimFactor	0.1464	2.64%	$1.0199 * 10^{-2}$	5.10%
Random initialization (baseline)		0.1427	N/A	$0.9809 * 10^{-2}$	N/A

we examined and therefore it can not be neglected. An additional 4–6% improvement compared to the random initialization could be achieved through appropriate initialization on a real life dataset using SimFactor and only 3–3.5% when using MF. The similarity preserving property of the SimFactor can be a disadvantage when the description matrix is too noisy, that is the description matrix does not capture item or user similarities.

Future work includes experimentation with other similarity metrics (e.g. Pearson correlation instead of cosine similarity) and the combination of various context information into a single similarity measure.

The utility, that is, whether 4–6% improvement worths the time of the initialization depends on many factors. The main cost factor is the MF step in SimFactor whose complexity greatly depends on the description data and the parameters of the factorization. The number of context states is much lower than the number of items/users and therefore the description matrix is relatively small, thus the initial MF step is relatively fast. Furthermore, the context state based initialization turned out to be the most efficient one, which suggest that context based initialization has positive utility in real-world recommender applications.

REFERENCES

1. Bell, R. M., and Koren, Y. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Proc of. ICDM-07, 7th IEEE Int. Conf. on Data Mining* (Omaha, Nebraska, USA, 2007), 43–52.

2. Bennett, J., and Lanning, S. The Netflix Prize. In *Proc. of KDD Cup Workshop at SIGKDD-07, 13th ACM Int. Conf. on Knowledge Discovery and Data Mining* (San Jose, California, USA, 2007), 3–6.

3. Boutsidis, C., and Gallopoulos, E. Svd based initialization: A head start for nonnegative matrix factorization, 2007.

4. Dhillon, I. S., and Modha, D. S. Concept decompositions for large sparse text data using clustering. In *Machine Learning* (2000), 143–175.

5. Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems 22*, 1 (2004), 5–53.

6. Hu, Y., Koren, Y., and Volinsky, C. Collaborative filtering for implicit feedback datasets. In *Proc. of ICDM-08, 8th IEEE Int. Conf. on Data Mining* (Pisa, Italy, December 15–19, 2008), 263–272.

7. Jolliffe, I. *Principal Component Analysis*. Springer Verlag, 1986.

8. Koren, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. of the 14th ACM Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD'08)* (Las Vegas, Nevada, USA, 2008), 426–434.

9. Kramer, M. A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal 37*, 2 (1991), 233–243.

10. Langville, A. N., Meyer, C. D., and Albright, R. Initializations for the nonnegative matrix factorization, 2006.

11. Pilászy, I., and Tikk, D. Recommending new movies: Even a few ratings are more valuable than metadata. In *Proc. of the 3rd ACM Conf. on Recommender Systems (Recsys'09)*, ACM (New York, NY, USA, 2009), 93–100.

12. Pilászy, I., Zibriczky, D., and Tikk, D. Fast ALS-based matrix factorization for explicit and implicit feedback datasets. In *Proc. of the 4th ACM Conf. on Recommender Systems (RecSys'10)*, ACM (Barcelona, Spain, 2010), 71–78.

13. Rendle, S., and Schmidt-Thieme, L. Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *Proc. of RecSys-08: ACM Conf. on Recommender Systems*, ACM (New York, NY, USA, 2008), 251–258.

14. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. Grouplens: an open architecture for collaborative filtering of netnews. In *Proc. of CSCW-94, 4th ACM Conf. on Computer Supported Cooperative Work* (Chapel Hill, North Carolina, USA, 1994), 175–186.

15. Ricci, F., Rokach, L., and Shapira, B. Introduction to recommender systems handbook. In *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., Artificial Intelligence. Springer US, 2011, 1–35.

16. Salakhutdinov, R., and Mnih, A. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. MIT Press, Cambridge, Massachusetts, USA, 2008.

17. Smilde, A., Bro, R., and Geladi, P. *Multi-way Analysis*. Wiley, West Sussex, England, 2004.

18. Snedecor, G. W., and Cochran, W. G. *Statistical Methods*, 7th ed. Iowa State University Press, 1980.

19. Takács, G., Pilászy, I., Németh, B., and Tikk, D. Major components of the gravity recommendation system. *SIGKDD Explor. Newsl. 9* (December 2007), 80–83.

20. Takács, G., Pilászy, I., Németh, B., and Tikk, D. Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research 10* (2009), 623–656.

21. Takács, G., Pilászy, I., and Tikk, D. Applications of the conjugate gradient method for implicit feedback collaborative filtering. In *RecSys'11: Proc. of the 4th ACM Conf. on Recommender Systems* (Chicago, IL, USA, October 23–27, 2011), 297–300.

22. Thai-Nghe, N., Drumond, L., Horváth, T., Krohn-Grimberghe, A., Nanopoulos, A., and Schmidt-Thieme, L. Factorization techniques for predicting student performance. In *Educational Recommender Systems and Technologies: Practices and Challenges (ERSAT 2011)*. IGI Global, 2012, 1–25.

23. Wild, S., Wild, W. S., Curry, J., Dougherty, A., and Betterton, M. Seeding non-negative matrix factorization with the spherical k-means clustering. Tech. rep., 2003.

24. Wild, S. M., Curry, J. H., and Dougherty, A. Improving non-negative matrix factorizations through structured initialization. *Pattern Recognition 37*, 11 (November 2004), 2217–2232.

A Model for Serendipitous Music Retrieval

Markus Schedl, David Hauger
Johannes Kepler University
Department of Computational Perception
{markus.schedl,david.hauger}@jku.at

Dominik Schnitzer
Austrian Research Institute for
Artificial Intellgence
dominik.schnitzer@ofai.at

ABSTRACT

Music retrieval systems that take into account the user's taste and information or entertainment need when building the results set to a query are of vital interest for academia, industry, and the passionate music listener. Unfortunately, preliminary attempts to incorporate such aspects have been rather sparse so far. Focusing on the problem of music recommendation, we therefore present a new model that combines several factors we deem to be important for personalizing retrieval results: similarity, diversity, popularity, hotness, recentness, novelty, and serendipity. We further propose different ways to measure the corresponding aspects and, where available, point to literature for a more detailed elaboration of the corresponding measures. In addition, we propose the use of social media mining techniques to address the problem of estimating popularity and hotness in a geo-aware manner.

MOTIVATION AND BACKGROUND

Most music retrieval approaches focus on the concept of *musical similarity* to create the results set for a query, which may be an excerpt of a music piece or the name of an artist of a song. This musical similarity may be computed on some kind of acoustic features extracted from the audio signal via signal processing techniques (*content-based*); alternatively, it might be derived from listening co-occurrences among users (*context-based* or *collaborative filtering*). Accordingly, the performance of such a classical retrieval system is judged the better the more similar the returned pieces are to a given seed. Also the most important evaluation forum for music retrieval methods, the annual `Music Information Retrieval Evaluation eXchange` (MIREX) [4] focuses strongly on similarity as relevance criterion. Although this is a very intuitive manner of assessment, it does not take into account that the information need of the user might not be centered around the concept of similarity alone. Indeed, for many popular music retrieval tasks, such as automated playlist generation [6, 10] and music recommendation [9, 7], the listener does not necessarily seek for a list of closest matches in terms of similarity. In a user study we performed to assess the quality of an automated, content-based playlist generation approach [15], we were in fact often told that our playlists were too perfect or homogeneous, thus boring.

CaRR 2012, February 14, 2012, Lisbon, Portugal.

We therefore believe that a new generation of music retrieval systems should not only rely on similarity scores derived from the audio signal or from contextual data, but should instead take the following factors into account:

Similarity:
Similarity should be taken into account in various dimensions. One set of dimensions might be based on music properties such as rhythm, harmony, or timbre, inferred from the audio signal [8, 19]. Another might model resemblance according to other data sources, such as collaborative tags, playlist co-occurrences, or even images of album covers or promotional photographs [12, 11]. A third set of dimensions might be learned from a user's listening preferences, for example, by relating certain properties of the user context to particular categories of music [13]. To give an example, similarity could be defined as pieces that are usually listened together while a user is jogging or while being together with friends.

Diversity:
Although the results of a music retrieval request should contain music items similar to the query, they should also show a certain degree of diversity. For example, there is the well-known "album effect" [20], i.e., due to same recording settings, tracks on the same album usually show a higher level of audio similarity than other tracks (even by the same artist). To alleviate this issue, some retrieval systems filter results from the same album or even by the same artist as the seed. Others offer a parameter N to avoid repetitions of the same artist within N consecutive songs, for example, `YouJuke` [5].

Familiarity/Popularity vs. Hotness/Trendiness:
Familiarity or popularity describes how well-known an artist is, whereas hotness or trendiness relates to the amount of buzz or attention an artist is currently getting [1]. Popularity has a more positive connotation than the neutral expression of familiarity. However, we will use the terms interchangeably in the remainder of the paper, likewise the terms hotness and trendiness. According to the temporal dimension, popularity can be seen as a longer lasting property, whereas hotness usually relates to recent appreciation of typically shorter duration, although hot artists might also be very familiar to many people. To give an example, "The Beatles" are certainly popular, whereas "Lady Gaga" currently tends to rank higher on the hotness dimension.

Recentness:
This aspect distinguishes recently released songs from pieces that are older and therefore have a longer (playing) history. In contrast to the aspect of hotness, recentness does not require an artist to be recently popular, just a temporal closeness to the present.

Novelty:
If a music recommender keeps on suggesting tracks/artists known by the user, he or she will not be satisfied, even if the recommended items are perfectly suited otherwise. Hence, presenting recommendations novel for the user is a vital requirement for a recommender system.

Serendipity:
Serendipity means that a user is surprised in a positive way since he discovered an item he did not expect or was not aware of. Being able to make serendipitous recommendations is hence a well desired property for recommender systems [9]. In the context of music retrieval, we believe that the listener's music preference and taste as well as aspects of artist and song popularity have to be taken into account when we aim at providing serendipitous results. For instance, a fan of medieval folk metal might be rather disappointed and bored if the system recommends the band "Saltatio Mortis", which is well known for this music style. In contrast, for a user occasionally enjoying "Metallica" and "Bob Dylan", the former mentioned band may be a serendipitous recommendation.

A SERENDIPITOUS MUSIC RETRIEVAL MODEL

We regard similarity and diversity as orthogonal aspects. A music retrieval system should hence take into account the user's preference to retrieve music items that are similar according to a particular set of aspects (e.g., rhythm and timbre), albeit also ensuring a degree of diversity by including items that are dissimilar according to another set of aspects (e.g., artist name, song lyrics, or tags).
The proposed model for serendipitous music retrieval given a seed/query i, which can either be an artist a or a track t (by artist a)[1], is described by the following retrieval function:

$$\begin{aligned} r(i) &= \min_j \left(\sum_{s \in S} w_s \cdot \delta_s(i,j) - \sum_{d \in D} w_d \cdot (1 - \delta_d(i,j)) \right) \cdot \\ &\quad (w_p \cdot p(j, reg(l)) + w_h \cdot h(j, reg(l)) + w_r \cdot r(j)) \end{aligned} \tag{1}$$

$\delta_s(i,j)$ represents the sth $[0,1]$-normalized dissimilarity function out of the set S of aspects to measure similarity; D accordingly represents diversity aspects. The similarity and diversity aspects can be measured on the track or artist level or take both into account. w_s and w_d are weights that allow to control the importance of each similarity/diversity aspect in S and D, respectively. These weights can be either defined manually by the user, learned online via relevance feedback, or inferred from the user's past listening behavior. Also a mixture of these three strategies seems reasonable. The same holds for the weights w_f, w_h, w_n, and w_r, which control the influence of the factors familiarity, hotness, novelty, and recentness, respectively.

$f(j, reg(l))$ is a measure of familiarity of artist or track j ($[a,t]$), given a particular region $reg(l)$ of the world. This region may be defined by a location l, for instance, given by longitude and latitude values. It can alternatively be given as a political or cultural region. For example, popularity in a particular region can be estimated by analyzing shared folders in Peer-to-Peer networks of users in that region or from page count estimates of search engines [18]. The hotness of an artist or song $h(j, reg(l))$ in a region $reg(l)$ can be inferred from traditional music charts such as the "Billboard Hot 100" [3]. Unfortunately, such charts suffer from two major shortcomings. First, from a global point of view, there are many countries in which such charts are not available. Second, the computation of the rankings in different countries vary in terms of distribution channels included, for instance, online digital music distribution, classical record sales, or airplays. Alternatively, hotness can be inferred from `last.fm` playcounts. Also query strings from Peer-to-Peer networks were shown to relate to recent artist popularity [14].
Here we propose to estimate familiarity and hotness with respect to a given location from music-related microblogging activity. We derived typical listening patterns for 790 major cities of the world by applying natural language processing techniques to geo-localized `Twitter` streams. Using a data set of track and artist names, we then infer time-dependent (hotness vs. familiarity) and location-dependent ($reg(l)$) popularity estimates on the artist or track level. Artist-to-genre assignments, for example gathered from `allmusic.com`, enable the prediction of cultural listening patterns. To this end, the listening data is normalized, aggregated at the level of cities or countries C, and represented via a genre distribution vector for each c ($\in C$), denoted as g_c. Linking a collection of 48,800 artists with the set of `allmusic`'s 18 major genres[2], we can further compute the most "mainstreamy" and the most independent countries in terms of their music listening behavior. To this end, we compute the deviation of g_c from a mean global genre distribution vector. The result is visualized in Figure 1, where the y-axis illustrates the excess or shortfall of each genre in the countries depicted, respective to the global music taste. Hence, a value of 1.5 for a (genre,country)-pair signifies that the genre is listened to 150% more frequently in that country than the global consumption of this genre suggests. Likewise, a value of -1.0 (a shortfall of 100%) refers to the fact that a particular genre is never listened to in the country under consideration.

The novelty $n(j)$ is particularly important for serendipitous recommendation, as already explained above. A straightforward definition may use a binary attribute describing whether or not the user has item j in her collection. This is of course only an approximation for whether a user knows an item, as it neglects various channels of music consumption (e.g., listening to analog disk records or tapes, music streaming (although in this case the streaming provider might offer an API), listening to music at a party or to somebody else's music collection). Nevertheless, this definition may serve as a good proxy for the actual novelty of the music piece under consideration.

[1]The seed items i and the potential results j might hence be seen as tuples $[a,t]$.

[2]The used genres are avantgarde (av), blues (bl), celtic (ce), classical (cl), country (co), easylistening (ea), electronica (el), folk (fo), gospel (go), jazz (ja), latin (la), newage (ne), rap (ra), reggae (re), rnb (rn), rock (ro), vocal (vo), and world (wo).

The recentness measure $r(j)$ might be expressed by an exponentially decaying function $e^{-(now-rd(j))}$, where $rd(j)$ is the release date/year of track j. The release date can be gathered from the ID3 tag of an audio file or from music databases such as `MusicBrainz` [2].

CONCLUSIONS AND FUTURE WORK

In this paper, we first outlined requirements for building a serendipitous music retrieval system. More precisely, we elaborated on the factors similarity, diversity, familiarity, hotness, novelty, and recentness. We then presented a retrieval model that takes into account all these factors, and we indicated how measures for the individual factors can be defined.

We are currently developing a prototypical music retrieval system that employs the proposed model. Having ready algorithms for computing the different factors, we aim at applying them on a large scale, using a real-world music repository. To this end, we are experimenting with a set of 2.3 million tracks, for which we compute content-based similarity scores with our top-performing[3] signal-based similarity algorithm [19]. As an alternative to content-based similarity, we construct a text-based similarity measure from `last.fm` tags, web pages about music artists [17], and microblogs [16]. Each of these measures can be used as a similarity or diversity function. Familiarity and hotness are derived using social media mining techniques. Recentness is computed using editorial metadata provided by the record labels. Novelty will be assessed in our foreseen retrieval system using a hybrid approach that takes into account several indicators to predict whether a piece is known to a user or not. Such indicators include listening information from music streaming services, such as `last.fm` or `Spotify`, existence of the track in the user's digital music collection, and mentions of the track or artist in the user's microblog posts.

Particularly challenging will be the evaluation of the system as it requires assessing the user's satisfaction in various dimensions and according to different usage scenarios. We might elaborate evaluation experiments based upon some ideas from [21]. Even though validation of the proposed system will be challenging, we are confident that this work will help bring the vision of serendipitous music retrieval a bit closer to reality.

ACKNOWLEDGMENTS

This research is supported by the Austrian Science Funds (FWF): P22856-N23.

REFERENCES

1. http://musicmachinery.com/2009/05/25/artist-similarity-familiarity-and-hotness (access: Sep 2011).
2. http://musicbrainz.org (access: Oct 2011).
3. http://www.billboard.com/charts/hot-100 (access: Oct 2011).
4. http://www.music-ir.org/mirex (access: Sep 2011).
5. http://www.youjuke.org (access: Oct 2011).
6. Aucouturier, J.-J., and Pachet, F. Scaling Up Music Playlist Generation. In *Proc. IEEE ICME* (Aug 2002).
7. Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Lüke, K.-H., and Schwaiger, R. InCarMusic: Context-Aware Music Recommendations in a Car. In *Proc. EC-Web* (Aug–Sep 2011).
8. Casey, M. A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., and Slaney, M. Content-Based Music Information Retrieval: Current Directions and Future Challenges. *Proc. IEEE 96* (Apr 2008).
9. Celma, O. *Music Recommendation and Discovery in the Long Tail*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2008.
10. Knees, P., Pohle, T., Schedl, M., and Widmer, G. Combining Audio-based Similarity with Web-based Data to Accelerate Automatic Music Playlist Generation. In *Proc. ACM SIGMM MIR* (Oct 2006).
11. Lıbeks, J., and Turnbull, D. Exploring "Artist Image" using Content-based Analysis of Promotional Photos. In *Proc. ICMC* (Jun 2010).
12. M. Schedl et al. Towards Automatic Retrieval of Album Covers. In *Proc. ECIR* (Apr 2006).
13. McFee, B., and Lanckriet, G. Heterogeneous Embedding for Subjective Artist Similarity. In *Proc. ISMIR* (Oct 2009).
14. N. Koenigstein et al. Spotting Out Emerging Artists Using Geo-Aware Analysis of P2P Query Strings. In *Proc. ACM SIGKDD* (Aug 2008).
15. Pohle, T., Knees, P., Schedl, M., Pampalk, E., and Widmer, G. "Reinventing the Wheel": A Novel Approach to Music Player Interfaces. *IEEE Transactions on Multimedia 9* (2007).
16. Schedl, M., Knees, P., and Böck, S. Investigating the Similarity Space of Music Artists on the Micro-Blogosphere. In *Proc. ISMIR* (Oct 2011).
17. Schedl, M., Pohle, T., Knees, P., and Widmer, G. Exploring the Music Similarity Space on the Web. *ACM Transactions on Information Systems 29*, 3 (Jul 2011).
18. Schedl, M., Pohle, T., Koenigstein, N., and Knees, P. What's Hot? Estimating Country-Specific Artist Popularity. In *Proc. ISMIR* (Aug 2010).
19. Seyerlehner, K., Schedl, M., Pohle, T., and Knees, P. Using Block-Level Features for Genre Classification, Tag Classification and Music Similarity Estimation. In *Extended Abstract to MIREX/ISMIR* (Aug 2010).
20. Whitman, B., Flake, G., and Lawrence, S. Artist Detection in Music with Minnowmatch. In *Proc. IEEE NNSP* (Sep 2001).
21. Yuan Cao Zhang, Diarmuid O Seaghdha, Daniele Quercia, Tamas Jambor. Auralist: Introducing Serendipity into Music Recommendation. In *Proc. WSDM* (Feb 2012).

[3] Our approach ranked first in the MIREX 2010 and 2011 task of *Audio Music Similarity and Retrieval*.

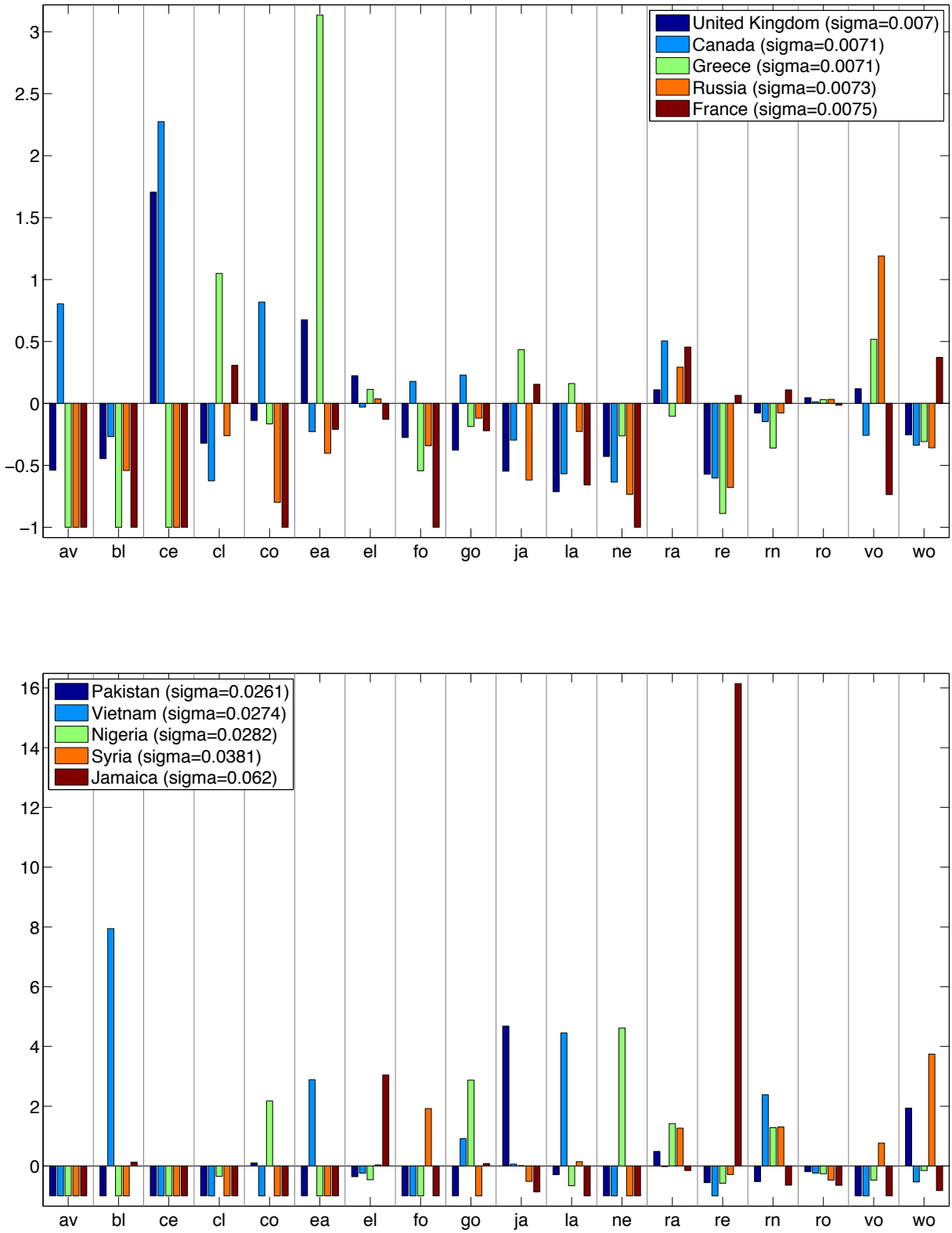


Figure 1. Relative deviations from mean global genre distribution for countries with most (top) and least (bottom) representative listening behavior.

Overview and Analysis of Personal and Social Tagging Context to construct User Models

Karin Schoefegger
Knowledge Management Institute, Graz
University of Technology
Inffeldgasse 21a, 8010 Graz, Austria
k.schoefegger@gmail.com

Michael Granitzer
Knowledge Management Institute, Graz
University of Technology
Know-Center
Inffeldgasse 21a, 8010 Graz, Austria
mgrani@know-center.at

ABSTRACT
The quality and user acceptance of personalized services such as personalized information retrieval and navigation or content recommendation depends depends besides the personalization mechanism on the quality, validity and accuracy of the employed user model. In literature a variety of user model construction methods based on tagging activity in social tagging systems (STS) are presented, relying on different user contexts, e.g., the personal or social context. But up to now there is neither a concise overview of existing construction methods available nor a deeper analysis and discussion of the differences between these models. Such an analysis would for example ease evaluation but also enable system designers to choose the most appropriate one. Our work tackles this problem by providing a short overview of state-of-the art user model construction methods which employ social tags. This is followed by a statistical comparison of four different user model construction methods for STS based on tag-frequency. This analysis unveils that depending on the method chosen (based user's personal tagging behavior as well as community-based social strategies), the user model consists of different tags and tag frequency rankings, thus services employing different models will lead to different results.

Author Keywords
User Modeling, Social Tagging, Recommender Systems, Personalized Information Retrieval

ACM Classification Keywords
H.1.2 User/Machine Systems: Human Factors; H.1.2 Information Systems: Models and Principles—*Human information processing*

General Terms
Algorithms, Human Factors.

CaRR 2012, February 14, 2012, Lisbon, Portugal.

INTRODUCTION
In the last years the web has witnessed an explosion of information created and shared, by individuals and through social interaction, such as for example in social tagging systems or social streams. This amount of information has generated a huge need for more effective access to the information, especially since each user has different expectations, goals, knowledge, information needs and desires to be satisfied. One way to ease the access is by personalization. To enable a system to adapt to e.g. the user's interests, the system usually builds a user model, an accurate machine-readable representation of the user. In social tagging systems, which enable a user to annotate a resource with a freely chosen keyword (tag) for later reuse and sharing, the user's interests are commonly modeled by using the tags available in the system.

In literature, a variety of user model construction methods have been presented but to the best of our knowledge, there is yet no concise overview of common techniques available. E.g. several methods build the user model solely based on the personal tagging context (only the tags the person used) or the social context (the tags all users used for a specific user's resources). Thus, in the first part of this work we present a literature overview and based on that, we present four typical user model construction methods and variations of it which can be used to enhance personalization in a social tagging system. Each of these models can be used for example to personalize resource ranking for retrieval with the FolkRank algorithm [13]. FolkRank provides the possibility to adapt the preference vector for a random surfer component to express user preferences by giving a higher weight to components that represents the user's preferences. It is expected that different user models (based on their context) lead to differently ranked results. This raises the question whether typical UM techniques lead to similar user model representations or whether they exhibit a different user model depending on what technique is chosen; which in turn is expected to lead to different search result rankings. Thus, in a second part, we present and discuss a statistical analysis of the comparison of the four chosen user model construction methods. We thereby focus on general statistics such as the 'tag richness' of a user model, the similarity between different possible models for one user as well as the correlation between the similarity of the possible user models for one user and the resource sharedness of this user.

Contribution of this work is the following:

- We provide a short overview of user model construction methods in Social Tagging Systems.
- We present a set of four possible and popular user modeling strategies to personalize services in STS. Two models are based solely on the user's tagging behavior, two represent the user model based on the tagging behavior of the community (userbase of the system).
- We show that the more tags are assumed to be part of a tag-based user model, the more specific a user model becomes and the different the four possible user models become. For personalized information retrieval a rich, specific user model is preferred, thus it plays an important role which user model is chosen to be fed into the personalization mechanism.
- We furthermore show that the more resources a user shares with the community, the more the personal and community-based profiles differ. For these users, it needs to be further investigated which user model is the most useful one, depending on the use case (e.g. personalized IR, expertise search, navigation).

SOCIAL TAGGING

Tagging has gained a major success in the web 2.0 as it plays an important role of helping user manage their resources. Users are encouraged to add tags to describe a website, a publication, a music track, a picture, etc. and to share these resources tags with other people. These tags indirectly reflect a user's interests, concerned topics, activities in daily life, and many more. Thus social tagging activities can serve as as an interesting source of information to build a user representation for any kind of personalized service.

Definitions

In the following we describe a few definitions of terms that are used throughout this work.

- A *User Library* $ULib(u)$ is the set of all resources a user u has annotated within a social tagging system.
- An *Author Library* $ALib(u)$ is the set of resources who are authored by a specific user u. E.g. an academic publication annotated with 'myown' in Bibsonomy [1] or added to the folder 'My Publications' in Mendeley[2]. Usually, $ALib(u) \subset ULib(u)$ and one resource r can have more than one author.
- A *Folksonomy* $F(R; U; T; TAS)$ is the central data structure of a social tagging system like Bibsonomy (academic publications as well as web resources), which is commonly seen as a lightweight classification structure built from so called tag annotations (TAS) added by different users to their resources. A folksonomy consists thus of a set of users U, a set of tags (i.e. freely chosen keywords) T, and a set of resources R with $ULib(u) \subseteq R \forall u \in U$, together with a ternary relation $TAS \subseteq U \times T \times R$ between them.
- A *Personomy* $P(u) = F(ULib(u); u; T(u); TAS), u \in U$ is a subset of $F(R; U; T; TAS)$, built only from tag annotations of a single user (person).
- The *Tag Sharedness* $TS(t)$ of a tag $t \in T$ is given by $|\{u \in U | t \in T(u)\}|$, thus is the amount of users who used a specific tag t . If we limit the tags of $F(R; U; T)$ to $F_2(R; U; T_2]$ where $T_2 = \{t \in T | TS(t) \geq 2\}$ is the set of tags with tag sharedness greater than 1, this eliminates on the one hand e.g. typical problems of a tag vocabulary such as misspellings etc. but also on the other hand increase the 'information value' of a tag in a user model.
- The *Resource sharedness* $RS(r)$ of a resource $r \in R$ is given by $|\{u \in U | r \in ULib(u)\}|$, thus is the amount of users who have a specific resource r in their library.

[1] http://www.bibsonomy.org
[2] http://www.mendeley.com

OVERVIEW OF USER MODELING APPROACHES FOR SOCIAL TAGGING SYSTEMS

In social tagging systems, it is generally assumed that annotating a resource is a good indicator for the current interests of a user. E.g. if a large number of a user's tagging activities include the tag 'sports', the user is likely to be interested in sports-related content. Some work also models the user's expertise- or knowledge-based on the user's tagging behavior ([14], [28], [4]).

A categorization of the user model construction is usually somehow problematic as some approaches apply more than one construction technique. For example, in concept-based methods often clustering is applied to identify clusters representing one concept. Also, this is not a complete overview but we rather aim at presenting the most common methods together with some prime examples.

Tag-Frequency-based User Modeling

In most approaches presented in literature, a tag-based user model representing the user's interests or expertise is usually provided in form of a weighted tag vector. In its simplest form, a weight is given by the frequency of the tag in the user's personomy. Tag-frequency-based user models are constructed in different ways, the following two main approaches can be distinguished:

Firstly, the user model is based on *tags extracted from a user's personomy*, thus on the tags the user has directly assigned to annotate resources. The work of [18] or [3] follow the naive approach which simply represents the user in form of a tag (frequency) vector which indicates that user u has used a tag t (a certain number of times) to annotate an item.

The personomy-based user model depends on the fact that the user has to collect a sufficient amount of annotation data such that the system can infer a useful user model. [1] present a more lightweight approach which builds the user model based upon *the tags that other users have added* to the resource the specific user clicks on. Similarly, in the work of [9] or [12] a personalization strategy for IR based on folksonomy data is presented, the user model is enriched with the tags other users

added to the resources of the user's library. In the recommendation research literature, tag-frequency-based user models are for example presented in [11] or [29].

An additional approach for user modeling can be based on *the tags chosen by a user from a list of suggested tags* as for example described in [6]. [15] enriches tagging activities with explicit ratings from users to model their likes and dislikes based on other similar users. More formally, a user is modeled as two tag vectors, on vector of tags denotes those tags a user is interested in another vector denotes the tags irrelevant for this user.

Graph-based User Modeling

The assumption behind a graph-based approach presented in [18] is that, if two tags co-occur in a user's tag annotation, there is some kind of relationship and the more often two tags are used together by a user, the stronger is this relationship. The top-k edges of the personomy graph representation with the highest weights (number of co-occurrences) and their incident nodes (tags) are chosen to represent the user model. A variant of this algorithm includes time-based information: every time the user adds another tag annotation, the current edge weights are decreased by a small percentage of their value. This technique is commonly known as evaporation from ant algorithms [10]. Another graph-based approach is for example described in [7] where a user model is constructed to enhance resource retrieval. Information from tagging as well as rating activities are used to personalize search results.

Clustering-based User Modeling

[27] constructs sets of tags which represent different interests of a user, they apply a community-detection algorithm on the tag-document network of a user's personomy. From the derived document clusters, a set of tags which appear on more than f% of the documents in the cluster are chosen. The final user model is a collection of those sets. One major drawback of this approach is that the user profiles are created solely on the basis of a specific personomy, thus the user models representing the interests of more users are not easily comparable. [25] proposes a method to map individual personomies on the corresponding folksonomy. They also presented how the model can be applied to services such as tag recommendations and tag-based social search. The authors of [16] propose besides topic-based clustering also a time-based clustering.

Concept-based User Modeling

In tag-based user modeling, the user model is often built solely on top of the user's personomy. While this approach meet individual needs and preferences, the differences between the users individual tag vocabularies creates discrepancies. E.g., when mapping for example a user model to an item representations (= a tag frequency vector containing tags all users have added to this item) for search or recommendation services or when identifying similar users based on their user models. This can be avoided if a user model contains concepts representing a set of tags rather than individual tags from this user. In the work of [24], the system gathers tag annotations of a specific user from a range of applications, then the user's tags are mapped onto a multi-domain model (provided by Wikipedia categories and their relationships) to filter tags and create a more structured interest vector as a user model. In [21], a hierarchical tag clustering approach is applied to the overall folksonomy where each cluster represents a concept. The user model for his/her interests is then a set of concepts where the user's interest in a specific cluster is given by the number of times the user annotated a resource with a tag from the cluster divided by the total number of the user's tag annotations. [26] introduce a novel user-centric tag model that enables a mappings between personal tag vocabularies and the underlying common folksonomy. An approach to map multilingual tags in a folksonomy is presented in [19] where two tags are considered as a valid translations if they expose similar global tag co-occurrence patterns.

Classifier-based User Modeling

A different approach for user modeling based on tagging data has been taken up in [8], who built a classifier-based user model for each user to recommend tags for when annotating new resources. Given the set of all tags in the user's personomy, binary classifiers are trained where each classifier corresponds to a specific tag in the user's personomy. Positive examples for training are those resources which have been tagged by the user with the corresponding tag and negative examples those which have not been annotated with that tag. In this sense, the user model construction is reduced to a binary text categorization task where each document has to be classified as interesting or not with respect to the user preferences provided by previous tag assignments.

User Modeling based on enriched tag information

A combination of tag annotations and content ratings are used to build a user model in [20]. The strength of an interests for a specific tag is a result of the ratings for items provided by the user which is tagged with that tag. In addition, their framework infers higher preference for those tags a user has applied, tags for which a user has searched as well as a third implicit tag signal, the quality of a tag which in turn is a measure depending on how often tags have been used for search and annotations etc. A tag categorization scheme such as presented in [5] could also be used to understand the meaning behind tags to map different user models and/or resource models. The authors utilize the multi-domain YAGO ontology ([23], a Semantic Web knowledge base with structured information extracted from WordNet and Wikipedia) to classify tags based on the intent of the tag.

USER MODELING: PERSONAL-, SOCIAL- AND AUTHOR-BASED STRATEGIES

As presented in section , tag-frequency-based vectors in the form of

$$UM(u) =< (t_1, f_1), (t_2, f_2), ... > \quad (1)$$

are the most common user models for STS. We present four user modeling construction methods to realize a user model as a tag vector, which we aim to analyze and discuss. Our main intent is not to study all possible user model construction methods (as presented in section), but to use basic ones

to illustrate that even in this rather simple setting, different construction methods lead to different user models and that this has to be carefully taken into account when to apply which user model construction method to which personalization service. The analysis in section shows insight into the differences.

In the following description of the four construction methods chosen, the notation $MethodName(u)$ refers to the corresponding user model for user u u, created with construction method $MethodName$. The notation $ModelName(u).tags$ refers to the set of tags in the corresponding user model for user u.

- In *US(u)*, the tag frequency is based on the amount of tag assignments of a single user u (the user's personomy).
- In *UF(u)*, the tag frequency is based on the tag assignments complete folksonomy, limited to the resources in $ULib(u)$.
- *AS(u)* is based on the tags chosen by a single user u to annotate his/her own publications, thus $AS(u).tags$ it is a subset of $US(u).tags$.
- *AF(u)* is based on the tagging activities of all users for the authored documents of one specific user u to represent the 'author model' of this user.

Each of the four methods UF, AF, US and AS represent different dimensions of a user model: First they are either based on the tagging vocabulary of a single user (US, AS) or the tag vocabulary of the community (UF, AF). And second, besides utilizing all resources in a user's library (US,UF), we analyze user modeling based on the user's authored documents in his/her library (AS,AF) as in literature authorship is assumed to represent the expertise of a user (see e.g. [22]). In addition to the presented four construction methods (presented below), we compare variants where each user model is required to contain exactly the top-k most frequent tags of the corresponding complete user model.These user models can for example be fed into an STS ranking algorithm such as the FolkRank algorithm [13] which provides the possibility to adapt ranking results based on the preference vector for a random surfer component.

Each of the above presented construction methods UF, US, AF or AS generate user models based on a different amount of information as for example the author's publications $ALib(u)$ is usually a subset of the user's library $ULib(u)$. Table 1 presents as an example four possible user models based for one selected user. It can be easily seen, that for this specific user the different user models not only contain some different tags, but also that the frequency-based rankings of the same tags are different in the four models.

Examples of UF,US,AF and AS for one specific user	
(UF)	(US)
toread (121)	studienarbeit (36)
bibsonomy (124)	www2009 (37)
web (141)	ontology (41)
semantic (167)	kassel (44)
web2.0 (172)	folksonomy (48)
ontology (188)	tool (50)
social (218)	online (52)
tagging (542)	toread (60)
folksonomy (564)	diploma_tesis (71)
(AF)	(AS)
2008 (17)	evidence_networks (2)
2009 (20)	www2009 (2)
social (20)	ibm-kde-tagging (3)
similarity (22)	studienarbeit (3)
bibsonomy (23)	2008 (4)
itegpub (23)	2009 (4)
semantic (24)	www2010 (4)
folksonomy (25)	tagorapub (6)
tagging (50)	itegpub (17)

Table 1. Example of different user model representations for one specific user based on 4 different construction methods. In this table the top-k (k=9) most frequent tags with their frequencies in the UM are shown. Details about the dataset are presented in section .

STATISTICAL ANALYSIS OF DERIVED USER MODELS

Dataset

In this work we will concentrate our analysis on Bibsonomy [2], an social bookmark and publication sharing system. The data set (available after signing a license agreement [1]). Some necessary preprocessing has been done before using the data. For example, to identify the semantic relationship between identical tags, we ignored capital letters and used stemming to such that 'Book', 'book', or 'Books' refers to the same tag 'book'. The dataset after preprocessing consists of 2.434.387 tag assignments by 6.463 users of the tagging system with 192.445 distinct tags for 551.540 resources. Out of the overall amount of users, 98 users used the tag 'myown' to tag their own publications. These tag annotations were used to identify a user's authored publications. The dataset shows the following statistics for tag sharedness (the number of users per of a tag) and resource sharedness (the number of users of a resource):

- Tag Sharedness (TS): On average each tag is used by 2.622 users (variance 90.84; Median is 1.0) and there are 60079 tags with $TS(t) > 1$. The highest ranked tags are: web (709), software (599), social (554), internet (484), search (467), blog (463), design (444), web2.0 (430), semantic (425) and research (405).
- Resource Sharedness (RS): On average there are 1.153 users per resource (variance 0.61; Median is 1.0) with a maximum sharedness of 84 users for a specific resource.

A variant of using the complete folksonomy F for analysis is to limit it to tags with a certain tag sharedness threshold. This eliminates tags that are used very rarely or might be the result of misspelling etc. For our dataset F_2, we set the tag sharedness threshold to value 2.

[1] http://www.kde.cs.uni-kassel.de/bibsonomy/dumps

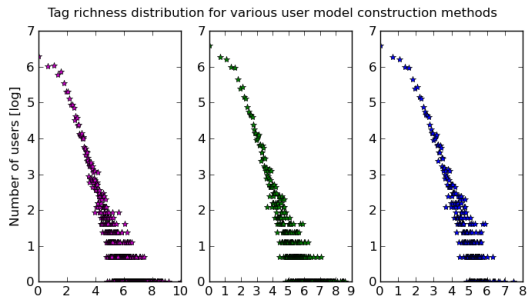


Figure 1. Tag Richness (= number of distinct tags in a personomy; on the x-axis) distribution for different user models UF (plot on the left), US for F (plot in the middle) and US for F2 (plot on the right). It can be seen that only a small number of users have a very rich user model (which is independent of construction method), although UF provides a richer user model than US.

Tag Richness of different User Models				
User Model Construction Method	Min	Max	Mean	Median
$UF(UF_2)$	1(1)	25879 (21742)	140.9 (104.0)	12.0 (10.0)
$US(US_2)$	1(1)	21381 (18829)	78.1 (57.4)	8.0 (7.0)
$AF(AF_2)$	1(1)	637 (510)	65.4 (51.1)	28.0 (23.0)
$AS(AS_2)$	1(1)	294 (265)	29.6 (24.0)	17.0 (14.5)

Table 2. Analysis of Tag Richness (= number of distinct tags) for different user models. Measures are provided for the TR vector containing the TR of each user for which the corresponding user model. User models based on tags from all users (UF,AF) are richer than those derived from the the user's tags (US,AS).

Amount and Richness of derived User Models

We define the Tag Richness of a user model as the number of distinct tags in a user model. The richer in tags a user model is, either the more specific the user interests are described or the more different interests a user has. Figure 1 shows the tag richness distribution for different user modeling construction methods, namely (UF and US) for folksonomy F as well as US for Folksonomy F_2. Data for author-based models is not visualized as they contain only a small amount of users. Table 2 presents the details about the tag richness for user models constructed by the four methods in more detail.

Impact of k on the amount of possible user models that can be constructed

The tag richness distribution follows a power-law distribution (independent on the user model construction method chosen), thus the number of users with a large amount of tags in their complete user model is rather small compared to the overall number of users. For 98 users, author-based models $AF(u)$ and $AF(u)$ can be constructed; if we set $k = 15$, there are only 71 users left for which $AF(u)$ and 51 for which $AS(u)$ can be constructed. In addition, it seems that users who manage their own publication in the tagging system by using the tag 'myown' show a higher TR, which might be the result of the fact that those users are in general more active taggers.

Impact of tag sharedness on the amount of possible user models that can be constructed

Using dataset F_2 instead of F decreases the number of possible user models that can be constructed as well as the tag richness of the user models. For example, if we apply restriction 2 to construct $UF(u)$, this reduces the number of user models from 6463 users to 6176 users.

$$TR(t) \geq 2 \forall t \in UF(u).tags. \quad (2)$$

Impact of single- vs. community-based user modeling method

User models based on tags from all users of the system (UF) and (AF) are richer in terms of tags than the user models derived from the user's personomy (US) or a subset of the personomy (AS). Ideally, if the view of a user u on the resources in $ULib(u)$ corresponds to the view of the community on these resources, the models $UF(u)$ and $US(u)$ would result in the same set of distinct tags and an equal tag ranking. But neither do all users have the same expertise and motivation for tagging, nor do users usually use the same tags for one and the same concept, thus the user models contain different (number of) tags. If the view of a user differs significantly from the view of the community, then the most appropriate user model for information retrieval or recommendation of new resources would be the a community-based one (either UF or AF), whilst for tag recommendation and navigation in one's own library, the most appropriate user model would be the personal one (either US or AS). If the author-based representation differs significantly from the user-based representation, then the user's expertise is best represented by the author-based one (e.g. for expertise search) and the user's interests might be best modeled by the user-based one (for personalized IR or recommendations).

Based on these observations we analyze in the following the similarity between between various possible models for one user and the correlation between similarity and resource sharedness.

Vocabulary Overlap in derived User Models

For one specific user, different user model construction methods lead to user models for this user which consist of a different amount of tags. E.g., comparing two models for one user, there might be additional tags in $UF(u).tags$ which are not contained in $US(u).tags$. Thus, in a next step we analyze the similarity between the different user model construction methods. As a comparison measure, we use the Vocabulary Overlap coefficient [17], which is for example for a user u and the two user-based models calculated as follows:

$$VOC(UF(u), US(u)) = \frac{|UF(u) \cap US(u)|}{min(|UF(u)|, |US(u)|)}, \quad (3)$$

where VOC takes a value in $[0, 1]$ and $VOC = 1.0$ means that all tags of the smaller set are contained in the larger

set. Due to better readability, we wrote $UF(u)$ instead of $UF(u).tags$. We consider a VOC value in the range of $[0.8, 1]$ as a good indicator for similarity. We do not use more common set overlap metrics such as the Dice Coefficient because this or related metrics are sensitive to the relative size of the two sets of tag vocabularies; in addition, if a user model representation is of a fixed size k as it is the case when limiting a user model to the top-k most frequent tags, then the Dice coefficient is equal to VOC.

Clearly, for the pairs $(AS(u).tags, AF(u).tags)$, $(US(u).tags, UF(u).tags)$ and $(AS(u).tags, UF(u).tags)$ with no limitation to the top-k most frequent tags, VOC equals 1.0 as these are subsets of each other. The mean of the VOC for pairs $(AF(u).tags, US(u).tags)$ for all users u is 0.78 with a variance of 0.05 and a median of 0.84. This means that for half of the authors, at least 84% of the tags in $AF(u).tags$ are contained in $US(u).tags$, thus in general these two construction methods overlap quite well.

Figure 2 shows the VOC distribution for the folksonomies F as well as F_2 for various values of k, where k limits the user model to the top-k most frequent tags (and it also means that the initial user model has to contain at least k tags and thus, the less users exist for which this model can be constructed). The distributions do not change significantly for the two different folksonomies F and F_2. First, it can be seen, that the lower k is chosen, the more users exist for which the tag sets in e.g. $(UF(u).tags, US(u).tags)$ or in $(AF(u).tags, AS(u).tags)$ overlap; the higher k is chosen, the more 'specialized' the user models becomes (as more more users exist for which the possible user models for this specific user contain different tags). Second, in general it is clear that the more similar $UF(u)$ and $US(u)$ are, the more similar is the view of one person on his/her library to the view of the community on this set of resources. Based on the figure we can conclude that for 40-50% of users these two views show a good overlap (depending on the value chosen for k). It can be seen that the amount of users with a higher overlap between (UF) and (US) decreases the higher k is (and thus the more tags are in the user model). This means that for higher ranked tags the vocabulary overlap is larger than for lower ranked tags. In general, the tag vocabulary overlap is for a large number of users below 0.8.

Figure 3 shows the comparison of tag overlap between various combinations of user model construction methods for $k = 15$. As we compare with author models and the number of authors in our dataset is limited, the number of user models to compare is rather small. Nonetheless, it is obvious that the VOC distribution (with a coefficient between 0.4 an 1.0) seems rather uniform.

Tag Overlap between user-based and community-based User Models and Resource Sharedness

In the previous section, we have shown that depending on the k-value chosen for the profile construction, for about half of the users the personal tagging behavior matches the community tagging behavior quite well. For the rest, the community-based user models do not overlap very well with the personal

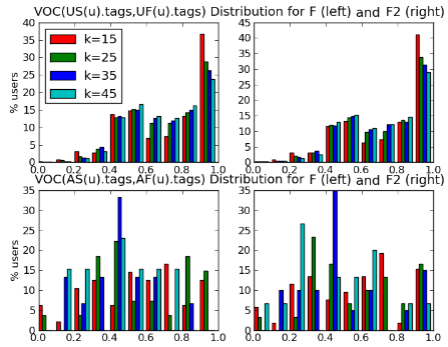


Figure 2. Tag vocabulary overlap distribution for (UF,US) in upper row and for (AF, AS) in lower row various values of k; Left column shows data for folksonomy F, right column for F_2 The x-axis shows the VOC value, the y-axis the % of users with this VOC. The higher k, the more less frequent tags are part of the user model.

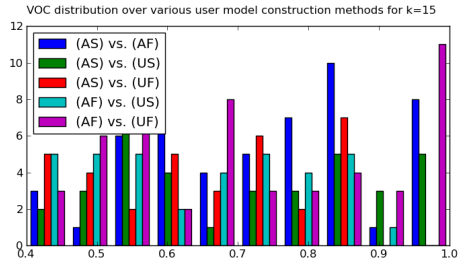


Figure 3. Tag vocabulary overlap comparison between various user models construction methods. The x-axis shows the VOC, the y-axis the amount of users. Note that the number of author models is limited, thus the amount of user models to compare is also rather small.

models. At this stage, the question arises whether there is a correlation between VOC and the overall resource sharedness $ORS(u)$ of a user. As the overall resource sharedness of a user u we define the average of the resource sharedness of the resources in a user's library; more precisely it is given by the following equation:

$$ORS(u) = \frac{\sum_{r \in ULib(u)} RS(r)}{|ULib(u)|} \tag{4}$$

where $RS(r)$ denotes the resource sharedness of resource r (see definitions in section). The correlation between $ORS(u)$ and $VOC(US(u), UF(u))$ provides an indication whether the divergence in community and personal profiles for some users emerge from the sharedness of resources. For folksonomy F the Spearman correlation has a value of -0.85 (p value 0.0) at $k = 25$ and -0.86 (p value 0.0) at $k = 45$. For F_2 the Spearman coefficient has a value of -0.82 (p value 0.0) at $k = 25$ and -0.81 (p value 0.0) at $k = 45$. Thus, there is a high negative correlation between the overall resource sharedness and the VOC of a user, the higher $ORS(u)$ the lower is the $VOC(US(u), UF(u))$ in our dataset. This means that the more resources a user shares with others, the more the community and personal profiles differ.

DISCUSSION AND FUTURE PLANS

In this work we presented a concise overview of possible user modeling methods for users, of which tag-frequency-based methods are the most common. We statistically analyzed two different versions, a community-based and single-user-based (personal) tag-frequency model and variations of those and showed that the more tags we require the user model to contain, the less user models can be constructed solely on tag annotations. For personalized IR, a rich and specific user model is preferred, thus for users with a low number of tags additional information is needed to construct a valid model. Author-based user modeling provides an alternative to library-based methods as it better reflects the expertise, whilst the user library itself might contain resources the user has tagged for different purposes; not many users provide this information in a STS. We showed that different tag-frequency-based user models for a user show a high tag overlap for about half or less of the users, but the % of users with a high VOC between community and personal models decreases the more tags a user model is required to contain. We also showed that the more resources a user shares with others, the more the community and personal profiles differ.

The derived empirical results are limited to a small-scale study, in which we wanted to find an indication if different popular user modeling construction methods lead to different user profiles (which in turn influences personalized services). These results, while not very surprising, are a first step towards a more focused work on comparing more sophisticated user models. The construction methods for comparison will includes not only tag-frequency based methods but also the graph-based, clustering-based or concept-based as presented in section . We will also expand our analysis to a larger datasets to compare author- and user-based profiles in more detail as well as enlarge the user model construction methods to more sophisticated ones: (a) as the user's interests change over time we want to take concept drift into account) and (b) as users tend to tag resources by opinion tags or use different tags to describe the same concept we want to apply preprocessing methods to filter out those tags that do not relate to topical interests and cluster those relating to the same topic.

The most important issue for future work is to test and experiment with the proposed user models in real tasks. We plan to apply the user models to a fixed set of algorithms for personalized search as well as user and item recommendations that uses theses user models. This should provide a clearer picture on the effects of the user models and its features on the selected personalization mechanisms.

Acknowledgments.
This work has been funded by the European Commission as part of the FP7 Marie Curie IAPP project TEAM (grant no. 251514) and by the OEFG (Austrian Science Foundation) under MOEL grant no. 490.

REFERENCES

1. Abel, F., Baldoni, M., Baroglio, C., Henze, N., Krause, D., and Patti, V. Context-based ranking in folksonomies. *Proceedings of the 20th ACM conference on Hypertext and hypermedia - HT '09* (2009), 209.
2. Benz, D., Hotho, A., Jäschke, R., Krause, B., Mitzlaff, F., Schmitz, C., and Stumme, G. The social bookmark and publication management system bibsonomy. *The VLDB Journal 19*, 6 (2010), 849–875.
3. Bogers, T., and van Den Bosch, A. Collaborative and Content-based Filtering for Item Recommendation on Social Bookmarking Websites. In *ACM RecSys 09 Workshop on Recommender Systems and the Social Web* (New York, USA, 2009).
4. Budura, A., Bourges-Waldegg, D., and Riordan, J. Deriving Expertise Profiles from Tags. In *CSE '09: Proceedings of the 2009 International Conference on Computational Science and Engineering*, IEEE Computer Society (Washington, DC, USA, 2009), 34–41.
5. Cantador, I., Konstas, I., and Jose, J. M. Categorising social tags to improve folksonomy-based recommendations. *Web Semantics Science Services and Agents on the World Wide Web 9*, 1 (2011), 1–15.
6. Carmagnola, F., Cena, F., Cortassa, O., Gena, C., and Torre, I. Towards a Tag-Based User Model: How Can User Model Benefit from Tags? In *User Modeling 2007*, C. Conati, K. McCoy, and G. Paliouras, Eds., vol. 4511 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2007, 445–449.
7. Clements, M., Vries, A. P. D., and Reinders, M. J. T. Optimizing single term queries using a personalized Markov random walk over the social graph. *Processing* (2008), 1–7.
8. de Gemmis, M., Lops, P., Semeraro, G., and Basile, P. Integrating tags in a semantic content-based recommender. In *Proceedings of the 2008 ACM conference on Recommender systems - RecSys '08*, ACM Press (New York, New York, USA, 2008), 163.
9. De Meo, P., Quattrone, G., and Ursino, D. A query expansion and user profile enrichment approach to improve the performance of recommender systems operating on a folksonomy. *User Modeling and User-Adapted Interaction 20*, 1 (Jan. 2010), 41–86.
10. Dorigo, M., and Di Caro, G. The Ant Colony Optimization Meta-Heuristic. In *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. McGraw-Hill, 1999, ch. 2, 11–32.
11. Firan, C. S., Nejdl, W., and Paiu, R. The Benefit of Using Tag-Based Profiles. *2007 Latin American Web Conference (LA-WEB 2007)*, Section 6 (Oct. 2007), 32–41.
12. Gemmell, J., Schimoler, T., Ramezani, M., Christiansen, L., and Mobasher, B. Improving folkrank with item-based collaborative filtering. *Recommender Systems & the Social Web* (2009).

13. Hotho, A., J, R., Schmitz, C., and Stumme, G. FolkRank : A Ranking Algorithm for Folksonomies. In *The Semantic Web: Research and Applications*, Y. Sure and J. Domingo, Eds., Lecture Notes in Computer Science, Springer (2006), 411–426.

14. John, A., and Seligmann, D. Collaborative Tagging and Expertise in the Enterprise. In *Proceedings of Collaborative Web Tagging Workshop held in Conjunction with WWW 2006* (2006).

15. Kim, H.-N., Alkhaldi, A., El Saddik, A., and Jo, G.-S. Collaborative user modeling with user-generated tags for social recommender systems. *Expert Systems with Applications 38*, 7 (July 2011), 8488–8496.

16. Koutsonikola, V., Vakali, A., Giannakidou, E., and Kompatsiaris, I. Clustering of social tagging system users: A topic and time based approach. *Web Information Systems Engineering-WISE 2009* (2009), 75–86.

17. Manning, C. D., and Schütze, H. *Foundations of Statistical Natural Language Processing*, vol. 26. MIT Press, 1999.

18. Michlmayr, E., and Cayzer, S. Learning user profiles from tagging data and leveraging them for personal(ized) information access. In *Proceedings of the Workshop on Tagging and Metadata for Social Information Organization, 16th International World Wide Web Conference* (2007).

19. Noh, T.-G., Park, S.-B., Yoon, H.-G., Lee, S.-J., and Park, S.-Y. An automatic translation of tags for multimedia contents using folksonomy networks. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval SIGIR 09* (2009), 492.

20. Sen, S., Vig, J., and Riedl, J. Tagommenders: connecting users to items through tags. In *Proceedings of the 18th international conference on World wide web*, ACM (2009), 671–680.

21. Shepitsen, A., Gemmell, J., Mobasher, B., and Burke, R. Personalized recommendation in social tagging systems using hierarchical clustering. *Proceedings of the 2008 ACM conference on Recommender systems - RecSys '08* (2008), 259.

22. Song, X., Tseng, B. L., Lin, C.-y., and Sun, M.-t. ExpertiseNet : Relational and Evolutionary Expert Modeling. In *User Modeling 2005* (2005), 99–108.

23. Suchanek, F., Kasneci, G., and Weikum, G. YAGO: A Large Ontology from Wikipedia and WordNet. *Web Semantics Science Services and Agents on the World Wide Web 6*, 3 (2008), 203–217.

24. Szomszor, M., Alani, H., Cantador, I., and Shadbolt, N. Semantic modelling of user interests based on cross-folksonomy analysis. *The Semantic Web-ISWC 2008* (2008), 632–648.

25. Szomszor, M., Cantador, I., and Alani, H. Correlating User Profiles from Multiple Folksonomies. In *Proceedings of the 19th ACM Conference on Hypertext and Hypermedia (Hypertext 2008)*, ACM (New York, NY, USA, 2008), 33–42.

26. Wetzker, R., Zimmermann, C., Bauckhage, C., and Albayrak, S. I Tag, You Tag: Translating Tags for Advanced User Models. In *WSDM '10: Proceedings of the Third ACM International Conference on Web Search and Data Mining*, ACM (New York, NY, USA, 2010), 71–80.

27. Yeung, A., Man, C., Gibbins, N., and Shadbolt, N. A Study of User Profile Generation from Folksonomies. *Proceedings of the Workshop on Social Web and Knowledge Management at WWW2008 356* (2008).

28. Yeung, C.-m. A., Noll, M. G., Gibbins, N., Meinel, C., and Shadbolt, N. On Measuring Expertise in Collaborative Tagging Systems. *Analysis*.

29. Zhao, S., Du, N., Nauerz, A., Zhang, X., Yuan, Q., and Fu, R. Improved recommendation based on collaborative tagging behaviors. *Proceedings of the 13th international conference on Intelligent user interfaces - IUI '08* (2008), 413.

PersonalWeb: An Extensible Framework to Recommend Web and Personal Information

João Guerreiro, Juliana Gomes, Daniel Gonçalves
UTL/INESC-ID
Rua Alves Redol, 9, 1000-029 Lisboa, Portugal
joao.p.guerreiro@ist.utl.pt, juliana.m.l.gomes@gmail.com, daniel.goncalves@inesc-id.pt

ABSTRACT
The large amount of information spread out among applications raised several challenges when trying to retrieve it, as information useful in the past is potentially useful in the future. Most recommender systems resort to contextual information to provide single-source document suggestions, disregarding the manifold Personal Information (PI) sources. We believe that PI can provide a richer background of the user's interests, providing personally-relevant and user-centered suggestions. In this paper, we describe an extensible framework that makes use of both contextual and personal information to provide recommendations that are relevant to the user and the task at hand, instead of only the latter. Herein we also present a preliminary evaluation of our framework still mostly based on contextual information. This pilot study presented satisfying results disambiguating contexts, suggesting web and personal documents and dealing with context changes, paving the way to its enrichment with more PI sources.

Author Keywords
Personal Information, Context-Awareness, Recommender Systems, Personally Relevant Results.

ACM Classification Keywords
H3.m. Information search and retrieval: Miscellaneous.

General Terms
Design, Experimentation, Human Factors.

INTRODUCTION
The increasing capacity of personal devices' storage along with the advent of the internet and its underlying services encourage users to maintain a growing amount of information. Our Personal Information (PI) is now scattered among several applications and *places*, such as the file system, online repositories, e-mail platforms, browser or social networks. In addition, users are reluctant in classifying their data as it is hard to predict its future value. We regularly misjudge the difficulty of re-finding it in the future and its value is often understood only then [10], neglecting that information useful in the past is potentially useful in the future [6]. Meanwhile, we have never had so many information at our disposal and available to support our tasks. However, searching on the web or personal space, people get distracted from their tasks from the moment they start searching, as unexpected search results may interrupt the users rather than help them completing their tasks [7].

There are many systems recommending task-related information. Several need user intervention [2, 8], which can be a distraction, and most recommend single-source documents (mainly web pages or personal documents). Context-awareness is vital to provide these suggestions, as it allows understanding the task at hand by resorting, for instance, to the web/personal documents opened or words written. However, considering it alone neglects the user's interests and previous interactions, providing only task-centered suggestions; though not user-centered, as their interests and needs may not be the same. Extreme cases are ambiguous words, such as *python* (snake vs programming language), but it goes beyond these cases as the suggestions relevance vary among users. For example, when talking with a specific friend, *John* always discusses *java* code-related issues. By being aware of previous interactions, it would suggest *java* related items instead of other programming languages. Resorting only to contextual information, it would depend on *java* being mentioned in this conversation to correctly direct the suggestions.

We believe that the enormous amount of PI scattered among applications can provide a richer background of the user's interests, providing user-centered results shaped to her/his needs. We also argue that PI sources should be part of a diverse set of suggestions, as users are not aware of the useful information they have at their disposal.

We built a standalone framework that considers the current context (what the user is doing, eg: words written, selected, documents/web pages opened) and the user's PI to provide user and task-centered recommendations, instead of only the latter. We performed a pilot study using our framework, yet with limited PI sources, but with the ambition to add more, making use of the framework extensibility. Results suggested good recommendations, supporting the use of PI in recommender systems, which motivated us to enrich our approach integrating more PI in the future.

CaRR 2012, February 14, 2012, Lisbon, Portugal.

RELATED WORK

There is an effort improving information retrieval by personalizing search results on demand [2, 8], however performing searches deviates the user from the current task as they need explicit input to present relevant information.

There are some systems that try to help the user by recommending documents proactively. Most of them resort to the current context to suggest web or personal documents related to the task. For example, *Watson* [1] resorts to the current document to suggest web documents that may be useful. *Tell Me More* [5] considers web and personal documents to provide additional information filtered by topics (quotes, actors, figures,..). While most approaches are limited in the information types they present, *TaskTracer* [3] identifies the current activity and present files, email messages or contacts that may be related to the task. Also, the information sources they consider to gather the context are limited as most of them only resort to the current documents. Even those resorting to other options focus only in the browser history [9].

We argue that resorting to the big amount of PI existent in our devices and web would present more diverse and user-oriented suggestions. It provides the opportunity to present more types of information, adding e-mail messages, instant messaging, contacts, posts, *tweets* to the previously used web and personal documents.

DESIGN AND RATIONALE

We built an extensible framework aiming at providing proactive user and task-related suggestions. We found crucial to provide flexibility at both ends: contextual sources to determine users' current activity; and PI sources used to direct the suggestions to users' interests and needs.

Gathering information from heterogeneous sources raises the inherent difficulty of managing it. We found a single representation to manage the information as a coherent whole instead of separate chunks, so the same information from different sources can be classified as such. With that it is possible to reinforce the information extracted from more than one source. With the most relevant *keywords* for the current context, we resort to Open Directory Project (ODP) to help disambiguating the context before sending it to the PI *Plugins*. These will be the ones providing personally relevant and user-centered suggestions by themselves and filtering the public ones, such as search engine results.

While context helps understanding what the user is doing at the moment, feeding this information to PI sources allows to naturally directing the results to what is relevant to that specific user (which may not be relevant to other users). In fact, we believe this integration between contextual and PI is the right way to provide task and user-centered results.

Framework Design

Our Framework is based in five main components: *Contextual Plugins*; *Plugin Manager*; *Data Coordinator*; *Retrieval Manager* and *Recommendation Plugins*. The *Contextual Plugins* are responsible for monitoring user activity and extracting the context from each active application and send it to the *Plugin Manager*. Its adapters convert the heterogeneous data into a common representation. The *Data Coordinator* weights the content to determine the most important attributes to store in a context database and removes the irrelevant ones. When needing suggestions, the *Retrieval Manager* requests the context (duration can be specified) where it basis the requests sent to the *Recommendation Plugins*, responsible to extract the context-related information.

Contextual Plugins

Most *Plugins* are application extensions able to extract the current context, which is the content and user's activity from a set of opened documents. This information includes the documents' entire text and some with special attributes (e.g. title, bold, selected or copied). At the moment, we have developed *Plugins* for *Mozilla Firefox*, *Mozilla Thunderbird*, *Office Excel*, *PowerPoint* and *Word*. We added two transversal *Plugins* to deal with additional information that the others cannot handle. One aims at counting the foreground windows duration and the other uses the *ContextLib* [4], a standalone library used to capture users' actions and that allowed us to be aware of the last written words, the contents in the clipboard and the items in the explorer (independent of the application).

Plugin Manager

This module contains an adapter to each *Plugin* which purpose is to convert the data to a single representation. Each *document* (independent of the *Plugin)* is converted to a representation containing the *id, date, type, duration (*opened*)* and *description* (the document text) and all the events that occurred on that document are associated to it using the *id*. Each event contains the *date*, the *documentId*, the *eventType (*e.g. selection, write) and the *text*. The information about the type and events allows us to weight differently the words. Likewise, we keep the records of the people related to each document (mainly the e-mail participants, document authors or comments).

Data Coordinator

This module removes irrelevant words (*stopwords)* and stores the information in the database. Then it assigns a weight to the terms collected based on term frequency (*tf*). Our corpus the set of contextual documents, so *tf* alone provides the most relevant terms for that corpus. If we considered the entire personal space, a measure like *tfidf* would probably be a better choice since it considers the term weight for each document regarding the entire corpus.

Retrieval Manager

Retrieval Manager requests the context of a specific period of time (e.g. last 2 minutes). We considered this configurable request so it was possible to deal with task

detection in the future or other similar feature that may benefit from context period specification. It extracts the concepts related to each one of the top ranked contextual keywords (from ODP). The most frequent concepts are the ones related to the actual context, which helps to narrow the context avoiding ambiguous results. Then these concepts and keywords are sent to PI *Recommendation Plugins*. The results from these *Plugins*, are then used, together with the queries, to help filtering information from public sources.

Recommendation Plugins

The *Recommendation Plugins* used resorted to *Bing*, for web pages and *Windows Search* for personal documents. These *Plugins* resort to the queries sent by the *Retrieval Manager* to extract the most relevant documents for that context (orderly). Adding *Plugins* is straightforward as resorting to the provided queries it is only necessary to add the necessary code to extract the relevant information to find context-related information.

Prototype

As a proof of concept prototype we developed a very simple standalone interface, which only concern was to present the recommended items (not aesthetics). The interface (Figure 1) consists of 2 panes to present personal and web documents. Some documents are presented with a thumbnail to ease the process of document recognition. Each document contains an image and a title, both clickable to open the document.

PILOT STUDY

We carried out a pilot study aimed at ascertaining if our framework could provide good recommendations resorting to personal and web documents, before introducing suggestions from other sources. At this phase, our main focuses were: disambiguation resorting to contextual information and ODP; the relevance of the documents suggested, to determine the quality of the queries sent to the *Recommendation Plugins*; and dealing with context changes without resorting to session boundary recognition.

Methodology

Firstly, we handed a questionnaire to the users that allowed us to create a profile (age, gender, educational attainment, profession) and perceive the usage they give to computer tools. Then the users had to perform three tasks, where they had to create a context and analyze the suggestions.

Task 1. An ambiguous specific subject (*python*, the snake) was previously defined and we provided a set of documents for the users to interact with. Then some web pages were suggested to the users by *PersonalWeb*. Users mentioned those that were related to the subject they interacted with.

Task 2. Participants had to choose a subject that they were used to work at and open a few documents related to that context. We asked them to mention documents that they were expecting to be suggested. Then we analyzed, with them, if the documents they mentioned were suggested and if there were other relevant ones besides those.

Task 3. Participants had to choose two different subjects. They navigated in those subjects sequentially and afterwards analyzed the suggestions to check if they were related with the second task only.

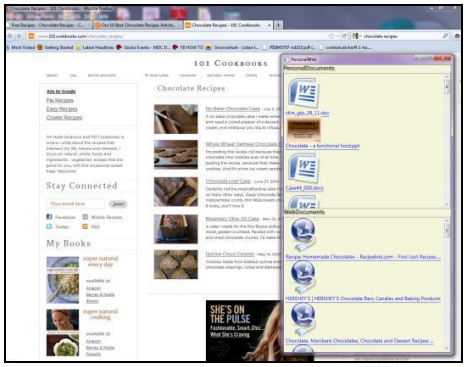


Figure 1. *Personal Web* recommending user and task related documents

Participants

We recruited 10 volunteers, mainly students from our university with ages between 18 and 25 (6); graduated between 26 and 40 (3); and one between 41 and 71.

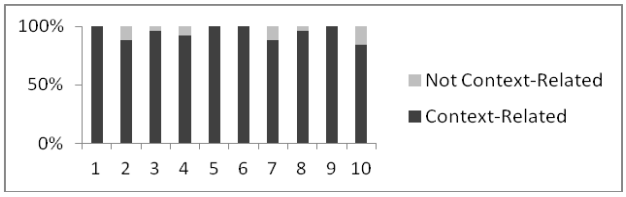


Figure 2. Percentage of Context-Related and Not Context-Related documents (total of 26) suggested, per user, in Task 1.

Results

We focused on a different aspect for each task; we wanted to assure that our framework was ready before adding more PI sources, instead of performing a full evaluation of our system. Our main goal with the first task was to acknowledge the success to disambiguate contexts resorting only to contextual information and ODP. We provided the documents used to create the context, so we focused the results only on web suggestions. In a total of 26 web pages suggested to each user, most of the suggestions (median=25) were found to be context-related (Figure 2).

In the second task, the context was from the users' workspace and they mentioned a set of documents they expected to be suggested. In this case, we focused on the personal documents' results. In half cases, all documents that the user was expecting were suggested (Figure 3) and in other two, more than a half (2 in 3 and 3 in 4).

It also presented relevant information that was not expected by the users (they tagged it as relevant in a post-test analysis). It reinforces the idea that we lose the notion of

the information we have on our personal computers. Figure 4 shows the total number of relevant and irrelevant documents suggested and overall the results were positive.

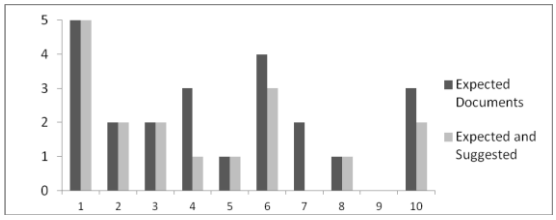


Figure 3. The number of *Expected Documents* and those that were *Expected and Suggested*, in task 2, for each user.

In task 3 our main goal was to verify if documents from the first context appeared on suggestions when working at the second one. That did not happen for 7 users but others had 1, 3 and 6 suggestions from the first context, mostly due to proximity among contexts (e.g. soccer and video games that shared some keywords, such as "*play"* and "*fun*").

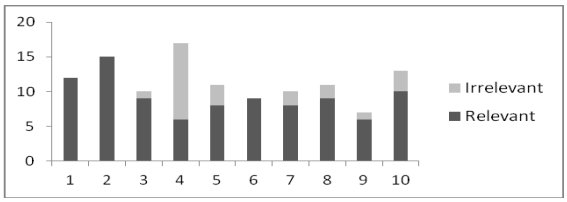


Figure 4. Total number of Relevant and Irrelevant documents in Task 2 suggestions, per user.

Discussion

These three tasks were a first impulse to our framework as it resulted in satisfying results. Task 1 showed its ability to deal with ambiguous results; Task 2 presented good personal document recommendations, either expected by the user or not; and Task 3 showed good results in dealing with context changes even without any concerns to session boundary recognition, as we believed that the context would be enough to make that distinction.

Most of the suggestions provided were considered relevant, which suggests that the queries sent to the *Recommendation Plugins* were successful. Adding more PI sources will provide more diverse and personally relevant results as web and personal documents are just a part of the information that may help the user to complete her/his task.

CONCLUSION

Information overload and fragmentation raised the big challenge of dealing with this great amount of data scattered among applications. At the same time, it also raised the opportunity to make use of all this information that was useful in the past and will probably be useful in the future. Most systems that try to recommend potentially useful information focus on contextual information and single-source recommendations, lacking the diversity and user-oriented suggestions. We built an extensible framework that is prepared to make use of the PI spread among application to provide user-centered instead of just contextual and task-related results. As a first step we evaluated a prototype with the basic functionalities, resorting to contextual information and providing recommendations of web and personal documents. Results suggested that the joint use contextual information with PI can help the user by recommending items both in line with the task at hand and with the user's background. These results motivated us to add a manifold set of PI sources (social networks, mobile data, mail messages, among others), which we believe will boost the framework results in a further and more exhaustive evaluation. As we were not concerned with the aesthetics in the first prototype, we will also focus our efforts on understanding how to make the suggestions without unnecessarily disturbing the user.

ACKNOWLEDGMENTS

João Guerreiro was supported by the Portuguese Foundation for Science and Technology, grant SFRH/BD/66550/2009.

REFERENCES

1. Budzik, J. and Hammond, K.J. User interactions with everyday applications as context for just-in-time information access. *In Proc of IUI, ACM* (2000).
2. Daoud, M. et al. A session based personalized search using an ontological user profile. *In Proc. of Symposium on Applied Computing, ACM Press* (2009).
3. Dragunov, A.N. et al. TaskTracer: a Desktop Environment to support multi-tasking knowledge workers. *In Proc. of IUI, ACM* (2005).
4. Barata, G. Blaze: Automating User Interaction in Graphical User Interfaces. *Master Thesis at Instituto Superior Técnico* (2009).
5. Iacobelli, F. et al. Tell Me More, not just "More of the Same". *In Proc. of IUI, ACM Press* (2010).
6. Jones, W. Finders, keepers? The present and future perfect in support of personal information management. *First Monday, 9* (2004).
7. Marshall, C.C. and Jones, W. Keeping encountered information, pages 66-67. ACM (2006).
8. Sieg, A., Mobasher, B., Lytinen, S. and Burke, R. Using concept hierarchies to enhance user queries in web-based information retrieval. *In Proc. of ICAIA* (2004).
9. Teevan, S. T. Dumais, and E. Horvitz. Personalizing Search via Automated Analysis of Interests and Activities. In Proc. of Research and Development in Information Retrieval, ACM Press, (2005).
10. Teevan, J. et al. How people find personal information. Personal Information Management, chapter 2, *University of Washington Press* (2007).

Monitoring of Real Interaction in Marketing Websites: Australia and Portugal's Perspective

Tomayess Issa
Curtin University
Address: GPO Box U1987, Perth WA, 6845
e-mail address: Tomayess.Issa@cbs.curtin.edu.au

Pedro Isaias
Universidade Aberta (Portuguese Open University)
Address: Palacio Ceia Rua da Escola Politécnica, nº 141-147
1269-001 Lisbon – Portugal
e-mail address: pisaias@univ-ab.pt

ABSTRACT
This paper discusses the importance of monitoring real interaction in marketing websites. Real interaction becomes indispensable in marketing websites, as designers can track users' behaviour toward the website to achieve both users and designers specific objectives. Real interaction has been either not fully considered or totally ignored in methodologies for information systems, marketing, and website development, as it needs training and specialists to interpret the statistical outcomes. This empirical evidence was collected using quantitative and qualitative approaches, which were employed in both Australia and Portugal. Survey data were gathered from 103 Australian and Portuguese respondents. The study released interesting results, which indicated that real interaction is very important in the website development process, in attracting more users to the websites and increasing the client's profit. It is important that clients should respond rapidly to users' comments. Finally, before real interaction is adopted in the website development process, privacy and security should be considered.

Author Keywords
Real Interaction, Marketing Methodology, Australia, Portugal

ACM Classification Keywords
H.5.3 [Group and Organization Interfaces]: Web-Based Interaction

General Terms
Design, Human Factors

INTRODUCTION
Effective human-computer interaction is a critical factor in website design. This aspect is very important in e-commerce websites to prevent any frustration to the consumers. "To make the business booming and fruitful, the vendor needs to answer and meet user requirements regarding services, products and prices; if end-user demand and requirements are dispersed, frustration will occur" [1; p.89]. It is important to understand 'real interaction' (what actually occurs) and how to improve a website. Tracking is considered an important aspect in web design as it is the means by which the designer can study real user behavior: "Tracking tells marketers where visitors cluster, more than anything else, these behavioral patterns demonstrate what attracts and engages visitors"[2; p.62]. Forrester Research found that "98% of site owners use traffic, such as hits and unique visitors, to gauge performances. While such indicators are useful, it[is] impossible to draw accurate conclusions about site performance from this data"[3]. Hence, a more detailed approach is needed. Detailed information about user behavior is very useful to enhance a website. According to Robinson et al. [2; p.62], "Using this information, marketers can build more intelligent and interactive environments that continue to attract more visitors, keep them engaged longer, and create the opportunity to build(ing) lasting relationships." Recent studies [4-9] have indicated that using real interaction in the web development process, especially for marketing websites, can help the designers to identify their users, and discover who they are, where they are coming from, and how they use their website. In addition, statistical information on real interaction can assist designers, users and organizations to make suitable decisions toward their websites, on major changes or "tweaking and minor changes reflective of shifts in customer usage or in your own current programs and services" [4; p.5]. This paper will assess the importance of real interaction in Australia and Portugal, and will answer the research question: "Can real interaction implementation in the website development process improve website usability?"

BACKGROUND
Real interaction is the actual way that real users interact with the site. Real interaction can be tracked to trace the performance of website visitors and how often they return to the website, either at the prototype stage or after initial implementation. According to Ramey [10], real interaction can be tracked by using the server log file (the record of activity on a site [10, p.398]) data to enhance the structure of the website. Analysis of the following types of data will help the designer to learn and understand how real interaction can be captured for use in the design process: the patterns of the dates and times of transactions; the IP

CaRR 2012, February 14, 2012, Lisbon, Portugal.

addresses, translated into domain names or countries of origin; the number of hits and the number of page views; the referring pages from which visitors come to a site; the amount of time spent on each page; the search terms used to hit the website pages; the search terms used to search within the website; the most frequent paths through the site; the most and least frequently visited pages. This information is very useful to the designers in helping them to enhance the structure of the website to attract more users to visit it. This set of analysis guidelines "focuses on marketing or technical issues rather than rhetorical issues like audience analysis" [10; p.397]. 'Real interaction' is an essential concept in website design, since, via the outcomes of real interaction, the designer has the ability to enhance the structure of the website to attract more users to visit it, not only the previous visitors but new visitors also. Therefore, this study will discuss the importance of the real interaction aspect and address the real interaction role in the New Participative Methodology for Marketing Websites (NPMMW)[11]. Real interaction is an important process; hence, it is considered one of the four key principles for the first author's PhD research. Bort [12; p.66] stated that real interaction is essential since one can answer questions like: "Where do people go most often? Which search engines do they most often come from? How long does it take them until they complete their transaction? And most importantly, how do you minimize the number of clicks to get them to the point where you want them to go?" This process will help the designers to track the client/customer users' (external) activities on the website and to "understand what people like and do not like" (Wagner 1997, p. 63). Usually, the real interaction software (i.e. Media Temple, Flash, FarCry, Smart Tools, GFI web Monitor, SpectorSoft and Google Analytics) will generate a log file, which will help the designers to understand the users' behaviour on the website. By the same token, recent studies [4, 13-17] confirmed that using the real interaction aspect in the web development process will provide tips and hints to designers and users on how to implement the changes or modifications to the website contents, design and navigation. Real Interaction is considered as one of the steps within the maintenance stage of the New Participative Methodology for Marketing Websites (NPMMW). The maintenance stage (SA6) consists of ongoing maintenance to the website, including update changes and the correction of errors in the website. Within this stage, there are two steps: Maintenance – Real Interaction and Feedback Tools (SE6.1): during the maintenance stage, real interaction needs to be tracked by using the server log file. This information is very useful to the designers in improving and enhancing the structure and the functionality of the website to encourage more users to visit it. In addition, feedback tools should be available on the website to allow the users to be able to contact the website owner for information or personal communication and to provide feedback about the website. For example, forms, a survey, a discussion forum, a contact form, a telephone number, and a prize should be available in the website to encourage the users to provide feedback. The researcher recommended that, in order to prevent spam, the organization's e-mail address should not be made available on the website. Maintenance – Project Review (SE6.2): this step should be available to ensure that the website is working within the project goals. This means that, after putting the website online, the designers need to check the website after one week to evaluate if the website structure is working according to the users' needs and requirements. One example of a tool that can be used for the project review is the checklist; i.e., a checklist for the goals and objectives, usability and technical requirements.

RESEARCH QUESTION AND METHODOLOGY

The main aim of the study is to address the following research question: "Can real interaction implementation in the website development process improve website usability?" This was achieved by reviewing the literature to identify the reasons why many users are frustrated and confused when working with websites. In addition, an online survey was carried out in Australia and Portugal to examine and assess the importance of real interaction in the web development process, especially with marketing websites. For this study, the researchers employed both qualitative and quantitative approaches, which are considered as opposite approaches for collecting and analyzing data. It was remarked that using both approaches would provide rich and valid information for the study [18-21]. Furthermore, using both these approaches will assist the researchers to reduce the gap between the research results and improve the data outcomes, since each approach has its own strengths and weaknesses, and using both approaches will serve to reduce the weaknesses and improve the strengths in the study [19, 21-23]. In summary, several studies [22, 24, 25] indicated that using mixed methods blends "eclectic views of knowledge, traditions of enquiry, methods and results; stays practice-orientated; and uses 'what works', not elitist stances. Classifications do not necessarily clarify 'the mix' though" [23; p.96].

DISCUSSION

Real interaction was introduced to identify the importance of real data on the way users interact with the website and to discover how this data can assist in the website development and maintenance process. Under the New Participative Methodology for Marketing Websites (NPMMW), real interaction is available during the maintenance stage to track users' behaviour and to improve the structure and the functionality of the website to encourage the users to return to the website to do auxiliary business. The most significant among the real interaction tools is commitment, meaning that the clients or the company need to deliver and modify all the necessary changes in the website strategy in order to meet the users' requirements; as Bort [12; p.65] stated, "using commercial web tracking tools implies a certain level of commitment, because you have to be prepared to respond". Based on the first author's PhD research, a new post-doctoral survey was

developed and distributed in Australia and Portugal to examine the importance of real interaction in the web development process. The audience for this survey was Information Systems Professionals from Australia (48 participants) and Portugal (55 participants). The outcomes from the survey were analyzed using SPSS. The researchers compared means and used T-tests (Tables 1 and 2). These statistics assisted the researchers to assess and evaluate the similarities and differences between the IS professionals in Australia and Portugal. The online survey targeted real interaction in responses to the following statements: (1) Monitoring of real interaction is very important in the web development process. (2) The industry should encourage its clients to use monitoring of real interaction in their websites to teach them the benefits behind it. (3) Monitoring of real interaction will increase the client's profit. (4) Monitoring of real interaction will attract more users to the website. (5) By adopting monitoring of real interaction in the website, clients should respond rapidly to the users' comments. In the survey, a five-point Likert scale – Strongly Disagree (SD), Disagree (D), Neutral (N), Agree (A), Strongly Agree (SA) – was used to examine the level of agreement with each statement of IS professionals in Australia and Portugal.

Table 1: Monitoring of Real Interaction Likert Scale

Q#	Country	SD	D	N	A	SA	Responses #
Q1	Australia	2	0	6	27	13	48
	Portugal	1	0	7	37	10	55
Q2	Australia	2	2	7	30	7	48
	Portugal	0	2	9	39	5	55
Q3	Australia	2	4	15	21	6	48
	Portugal	0	1	29	21	4	55
Q4	Australia	3	4	13	24	4	48
	Portugal	0	3	11	35	6	55
Q5	Australia	3	3	12	22	8	48
	Portugal	0	0	17	32	5	54

Table 2: Monitoring Real Interaction– t-Tests

Q#	Country	Responses #	Mean	SD	Mean Comparison t-Test
Q1	Australia	48	4.02	0.887	t:.134, df: 101, p:.894
	Portugal	55	4.00	0.694	
Q2	Australia	48	3.79	0.898	t: -.417, df: 101, p:.677
	Portugal	55	3.85	0.621	
Q3	Australia	48	3.52	0.967	t: 0.73, df: 101, p:.942
	Portugal	55	3.51	0.663	
Q4	Australia	48	3.46	0.988	t: -2.039, df:101, p: .044
	Portugal	55	3.80	0.704	
Q5	Australia	48	3.80	1.047	t:-1.040, df: 100, p:.301
	Portugal	54	3.78	0.604	

The results for question one indicated that real interaction is a very important aspect in the website development process, especially for marketing websites, with high average results (out of a maximum value of 5) for both countries (Australia: M=4.02, SD=.887; Portugal: M=4.00, SD=0.694). There was a non-significant difference between Australian and Portuguese respondents, since both sets of IS professionals agreed that monitoring of real interaction becomes an important aspect in the website development process to attract more users to visit the website and engage with it [7, 14]. As for question two, which asked if the industry should encourage its clients to monitor real interaction in their websites in order to teach them the benefits behind it, the results indicated that both Australia and Portugal shared similar average results (Australia: M=3.79, SD=0.898; Portugal: M=3.85, SD=0.621). There was a non-significant difference between Australian and Portuguese respondents. However, Table 1 confirmed that the majority of IT professionals fully agreed that industry and designers should take the role of encouraging their clients to identify the benefits behind using real interaction in their websites [8, 9]. IT professionals generously provided some comments regarding the monitoring of real interaction: "Better customer service will then be provided"; "Real interaction will be useful for development and progress of the website." The survey examined whether monitoring of real interaction would increase the client's profit (question three); the outcomes from Australia and Portugal have similar averages (Australia: M=3.52, SD=0.967; Portugal: M=3.51, SD=0.663). There was a non-significant difference between both countries, however: from the Table 1 it was noted that more IT professionals in Portugal had neutral responses. Survey respondents commented, "Monitoring real interaction could end both ways (loss/gain) since real interaction helps in process improvement." The survey went on to examine and assess whether monitoring of real interaction would attract more users to the website (question four). IT professionals from both Australia and Portugal agreed that using the real interaction aspect in the website development process would attract more users to the website: based on their feedback, designers will carry out the changes in content, design, and navigation. There was a significant difference between the two countries (Australia: M=3.46, SD=0.988; Portugal: M=3.80, SD=0.704). Since more people in Portugal agree with the phrase whilst in Australia less people agree with it. However, it was noted from Table 1 that the majority of IT professionals from Portugal confirmed that using real interaction in the website development process would attract more users to visit the website. Survey respondents commented: "Yes, tracking user behaviour on the website can be used as feedback to improve the usability of the website." "Do not focus on the profit, focus on increasing usability and profit will come as usability increases?" "Feedback forms are generally used for complaints, but simply monitoring the web stats will allow clients and developers to concentrate on the areas that end users visit and drop (or alter) those that are not visited." Finally, the researchers examined and investigated whether, by adopting the monitoring of real interaction in the website, clients should respond rapidly to the users' comments (question five). The results from Tables 1 and 2 indicated that IT professionals from both countries are fully agreed that clients should check the real interaction and

statistics to improve their website and respond to the clients' observations and comments. There was a non-significant difference between both countries (Australia: M=3.80, SD=1.047; Portugal: M=3.78, SD=0.604). The IS professionals generously shared with us their perspective: "Feedback links are a good idea but it is not the primary means that the team should use to understand their users." "Understand the acceptability level and site utility." "Interaction enables us to identify and correct errors in a timely manner and better adapt the site to the client's needs." "Monitoring real interaction will give your insight on what is hype now and what users are into nowadays and you can adjust the website with those insights." Finally, IS professionals from Australia and Portugal confirmed and corroborated the study research hypothesis that implementing the monitoring of real interaction in the website development process, especially from the marketing perspective, will enhance website usability, and designers and users can work closely to enhance and improve the website contents, and design. As a result of these changes, more users will visit the websites, and later the clients will start to win their users' trust toward the website. The survey results highlighted that the majority of websites of small online businesses are lacking in consistency and coherence, which might endanger an increase in purchases, through the limiting of visits to such websites.

CONCLUSION

This paper discussed the role of monitoring real interaction in the web development process. The survey results confirmed that monitoring real interaction is essential and a crucial concept in the web development process, especially in the maintenance stage, as IS professionals from both Australia and Portugal indicated its importance. Monitoring real interaction, if adopted in the website methodologies, will enhance and improve the quality of the websites in terms of efficiency, effectiveness, functionality, performance, user satisfaction, and other specific goals. Furthermore, user frustration and dissatisfaction will be diminished. As a result of these changes, the client will witness a reduction in user frustration, by making the website more approachable, friendly and interesting, and win the trust of the site visitors by meeting users' requirements. Finally, further research should be carried out to examine the relationship between monitoring real interaction, privacy, and security, as the majority of survey respondents raised this concern.

REFERENCES

1. Issa, T., *Online Shopping and Human Factors (Unpublished)*. 1999, Murdoch University: Perth. p. 99.
2. Robinson, R. and M. Peroff, *Learning from the Internet pioneers*. Medical Marketing and Media, 1999. **34**(2): p. 60.
3. Carroll, M. *Usability testing leads to better ROI*. 2004 [cited 2011 14 April 2011]; Available from: http://www.theusabilitycompany.com/news/media_coverage/pdfs/2003/NewMediaAge_270303.pdf.
4. Marek, K., *Web Analytics Overview*, in *Using Web Analytics in the Library*. 2011, American Library Association p. 5-10.
5. Marek, K., *Reporting and Analysis* in *Using Web Analytics in the Library*. 2011, American Library Association p. 26 - 32.
6. Marek, K., *Getting to Know web Analytics* in *Using Web Analytics in the Library*. 2011, American Library Association p. 11 - 16.
7. Rose, S., N. Hair, and M. Clark, *Online Customer Experience: A Review of the Business-to-Consumer Online Purchase Context* International Journal of Management Reviews 2011. **13**: p. 24 - 39.
8. Stauffer, C. and W. Grimson, *Learning Patterns of Activity Using Real Time Tracking* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000. **22**(8): p. 747 - 757.
9. Wu, G. and X. Hu, *Effects of Perceived Interactivity, Perceived Web Assurance and Disposition to Trust on Initial Online Trust*. Journal of Computer-Mediated Communication, 2010. **1**(26): p. 1-26.
10. Ramey, J., *Guidelines for Web Data Collection: Understanding and Interacting with Your Users*. Technical Communication, 2000. **47**(3): p. 397-410.
11. Issa, T. *Development and Evaluation of a Methodology for Developing Websites - PhD Thesis, Curtin University, Western Australia*. 2008; Available from: http://espace.library.curtin.edu.au:1802/view/action/nmets.do?DOCCHOICE=17908.xml&dvs=1235702350272~864&locale=en_US&search_terms=17908&usePid1=true&usePid2=true.
12. Bort, J., *Web tracking tools chart Internet ROI*, in *Software Magazine*. July 1997. p. 65-67.
13. Atterer, R., M. Wnuk, and A. Schmidt. *Knowing the User's Every Move – User Activity Tracking for Website Usability Evaluation and Implicit Interaction*. in *International World Wide Web Conference Committee (IW3C2)*. 2006. Edinburgh, Scotland
14. Kindberg, T., et al., *People, Places, Things: Web Presence for the RealWorld*. Mobile Networks and Applications, 2002. **7**: p. 365 - 376.
15. O'Brien, H. and E. Toms, *The Development and Evaluation of a Survey to Measure User Engagement*. Journal of American Society for Information Science and Technology 2010. **61**(1): p. 50 - 69.
16. O'Brien, H. and E. Toms. *Is there a Universal Instrument for Measuring Interactive Information Retrieval? The Case of the User Engagement Scale*. in *IIiX2010* 2010 New Brunswick, New Jersey, USA
17. Zheng, B. and J. Rajapakse, *Human Brain Anatomical Connectivity Analysis Using Sequential Sampling and Resampling*, in *Proceedings of the 2nd IAPR international conference on Pattern recognition in bioinformatics*. 2007 Springer-Verlag: Singapore p. 391 - 400.
18. Kaplan, B. and D. Duchon, *Combining Qualitative and Quantitative Methods in Information Systems Research: A Case Study* MIS Quarterly 1988. **12**(4): p. 571-586.
19. Hesse-Biber, S., *Emerging Methodologies and Methods Practices in the Field of Mixed Methods Research*. Qualitative Inquiry, 2010. **16**(6): p. 415 - 418.
20. Lopatovska, I. and I. Arapakis, *Theories, methods and current research on emotions in library and information science, information retrieval and human-computer interaction*. Information Processing & Management, 2010. **47**(4): p. 575 - 592.
21. Harrison, R. and T. Reilly, *Mixed Methods designs in Marketing Research* Qualitative Market Research: An International Journal 2011. **14**(1): p. 7 - 26.
22. Teddlie, C. and A. Tashakkori, *Foundations of Mixed Methods Research - Integrating Quantitative and Qualitative Approaches in the Social and Behavioral Sciences*. 2009, USA: SAGE Publisher
23. Maudsley, G., *Mixing it but not mixed-up: Mixed methods research in medical education (a critical narrative review)* Medical Teacher 2011. **33**: p. 92 - 104.
24. Creswell, J.W., *Research Design Qualitative, Quantitative, and Mixed Methods Approaches*. Second ed. 2003, USA: SAGE Publications.
25. Tashakkori, A. and C. Teddlie, *Mixed Methdology: Combining qualitative and quantitative approaches* 1998, CA: Thousand Oaks, SAGE.

Context Aware Point of Interest Adaptive Recommendation

Paulo Pombinho, Maria Beatriz Carmo, Ana Paula Afonso
Faculdade de Ciências da Universidade de Lisboa
Campo Grande, Lisboa, Portugal
ppombinho@lasige.di.fc.ul.pt, {bc, apa}@di.fc.ul.pt

ABSTRACT
Applications that allow the users to search for nearby points of interest have, recently, become very popular amongst mobile device users. However, the increasing amount of available information and the limitations of current mobile devices can hinder an efficient and helpful user experience. It is fundamental that what is shown to the user is relevant. We propose an adaptive recommendation function that uses location and temporal contexts combined with the historical context of the previous searches to quantify the relevance of the points of interest shown to the user.

Author Keywords
Context Aware, Mobile Devices, Degree of Interest.

ACM Classification Keywords
H.5.2. Information interfaces and presentation: User Interfaces - *Prototyping*.

General Terms
Algorithms, Experimentation.

INTRODUCTION
Nowadays it is very common for new mobile devices to have integrated positioning devices. This has fostered the development of location based services and, for this reason, the real time information about the location of users has become widely used in an extensive range of applications.

Applications, like Google Maps [1] that allow users to search nearby points of interest (PoI) using their mobile devices have become very popular. These applications allow the user to perform numerous tasks like finding relevant locations in the vicinity, such as, a gas station or a restaurant, or to calculate the shortest route to another location. The use of maps also allows the user to compare alternative locations, helping understand where each PoI is located and their geographical relation.

However, despite the usefulness of current systems, the huge amount of information that we have to deal with, creates a limitation for the users to correctly and easily perceive what is shown to them [2]. It is thus essential that we enforce that what is shown on the screen is important information for the user [3]. Furthermore, it is fundamental to include recommendations that provide users with information to guide them in choosing amongst the available information.

Recommender systems have been a popular research topic. However, traditional recommender systems do not take into account richer contextual dimensions, such as the type of location or the time of day, which are easily obtained using current mobile devices [4]. The adaptation to these context dimensions is a key feature to mitigate the limitations in the usability of small screens. According to the definition of Reichenbacher [5], adaptive visualization concerns the adjustment of all components of the visualization process, such as the interface, the information extracted from the data and the data codification, according to a particular context. This adaptive principle is especially important to increase the usability of searching information in mobile devices and to reduce the cognitive load inherent to mobile usage contexts.

In this paper we present an adaptive degree of interest function that uses historical context information, combined with location and temporal contexts, to automatically adapt its values to the previous user choices and searches, according to when and where they were made. This function also takes into account temporal distances.

The next section describes some of the related work. Afterwards, we present our previously proposed degree of interest function and, next, how this function can be improved with the adaptation to the historical, location and temporal contexts, and the addition of temporal distances. We then describe the developed prototype and finally present the conclusions and future work.

RELATED WORK
Although not directly related to mobile devices, Furnas [6] explores the presentation of large structures in windows of reduced size and uses fisheye views to address this issue. To formalize his conception of fisheye views, he introduces the concept of a degree of interest function. This function describes the interest the user has on a certain object. This function is defined as the combination of two components: an *a priori* importance that represents the global interest on the object, and *a posteriori* importance that depends on what the user is focusing on at the moment, expressed by a distance function.

CaRR 2012, February 14, 2012, Lisbon, Portugal.

In [7] the authors created the VisDB system which considers not only the objects that match a query, but also those that only satisfy it partially. To determine the relevance of each object, distance functions are used for each attribute specified by the user. Since different attributes may have different priorities, the users are allowed to set a weight factor for each of them. The resulting distances are then combined.

Based on the previous work, Reichenbacher [8] focuses on mobile device cartography. He considers not only the object and its location but also the time of events related to the object. To calculate the relevance of each object Reichenbacher combines three distances: a topical distance, a spatial distance and a temporal distance.

In a previous work [9], we have proposed a degree of interest function that enables the user to quantify the relevance of each point of interest. This function, described in more detail in [9], is based on Furnas's degree of interest function [6] and the work of Kheim and Kriegel [7], both described earlier, and will be briefly described in the next section.

PREVIOUS DEGREE OF INTEREST FUNCTION

The degree of interest function (DoI) quantifies the interest the user has on certain point of interest, p_j, as the average of the user interest (UI) on the k different attributes a_i, $i=1,2,\ldots,k$, multiplied by a weight w_{cat} for the category of p_j. Both a_i and w_{cat} are specified by the user.

$$DoI(p_j) = \frac{\sum_{i=1}^{k} UI(a_i, p_{ji})}{k} \times w_{cat} \in [0,1]$$

The User Interest function $UI(a_i,p_{ji})$ depends on the distance between the value selected by the user for the attribute a_i and the value p_{ji} of the point of interest p_j in the same attribute. The following distance functions were defined:

- For nominal attributes with l alternative values (e.g. types of restaurant)

$$Dist\left(a_i, p_{ji}\right) = \begin{cases} 0, if\ a_{i1} = p_{ji} \vee a_{i2} = p_{ji} \vee \ldots \vee a_{il} = p_{ji} \\ 1, if\ a_{i1} \neq p_{ji} \wedge a_{i2} \neq p_{ji} \wedge \ldots \wedge a_{il} \neq p_{ji} \end{cases}$$

- For numerical attributes with l alternative values (e.g. number of stars of a hotel)

$$Dist(a_i, p_{ji}) = min\left\{\left|\frac{a_{i1} - p_{ji}}{max_i - min_i}\right|, \left|\frac{a_{i2} - p_{ji}}{max_i - min_i}\right|, \ldots, \left|\frac{a_{il} - p_{ji}}{max_i - min_i}\right|\right\}$$

- For numerical attributes with a range of values (e.g. price)

$$Dist(a_i, p_{ji}) = \begin{cases} 0, if\ a_{i1} \leq p_{ji} \leq a_{i2} \\ \left|\frac{a_{i1} - p_{ji}}{max_i - min_i}\right|, if\ p_{ji} < a_{i1} \\ \left|\frac{a_{i2} - p_{ji}}{max_i - min_i}\right|, if\ p_{ji} > a_{i2} \end{cases}$$

where max_i and min_i correspond, respectively, to the maximum and minimum values known for that attribute.

- For the geographical distance, we use a normalized Euclidean distance

$$Dist(a_i, p_{ji}) = \sqrt{\left(\frac{x_a - x_{p_i}}{max_x - min_x}\right)^2 + \left(\frac{y_a - y_{p_i}}{max_y - min_y}\right)^2}$$

where (x_a, y_a) and (x_{pi}, y_{pi}), correspond, respectively, to the position of interest defined by the user, and the location of the point of interest.

After having calculated the distances, it is possible to determine the value of the UI function:

$$UI(a_i, p_{ji}) = 1 - Dist(a_i, p_{ji}) \times w_i,\ w_i \in [0,1]$$

where w_i is the weight for the attribute a_i, which can be defined by the user to specify the importance he gives to that attribute in the query being made.

Since all the distance functions, as well as the UI function, can only have values between 0 and 1, the result of the DoI function is also between 0 and 1, reflecting the degree of interest the user has on a certain point of interest.

By using the degree of interest function, we were thus able to order the various points of interest and present only the most relevant ones, while providing a numerical recommendation for those displayed.

ADAPTIVE DEGREE OF INTEREST FUNCTION

After evaluating the previously described degree of interest function [10] it was concluded that although considered useful, the use of this type of function can be confusing. One of the reasons for this is the need to specify a large set of attributes for each query made and simultaneously understand and specify the weights for each of them.

To overcome these limitations we propose some enhancements to the previous DoI function. These improvements are detailed in the following sub-sections.

Historical Context

To allow the reduction of the cognitive load when the user is specifying a query, we have added an option to use the historical context to automatically enhance the queries.

For each pair (attribute type, attribute value) we store an internal count of how many times it was queried. Whenever the user specifies a query, the attributes specified and their value are updated in the internal database. This historical log allows a summary of the interest of the user to be assembled over time. For instance, if the user almost always goes to Italian restaurants, it is possible to use this information to automatically specify the "type of restaurant" attribute without further action from the user. The weight used in the automatically defined attributes is calculated as a function of how many times it was chosen versus other queries.

Location and Temporal Aware Historical Context

The type of searches made by the users is, however, not always the same according to the location and temporal contexts. As an example, a user might have a different

interest whether he is searching for a restaurant at noon near his work or looking for one at dinner time near his home.

For this reason, we allow the users to define geographical areas that are relevant for them (for example, a work area, or a home area), by selecting, on the map, two of the area's corners. When the user performs a query, the logs are recorded / updated in the appropriate geographical area / time of day section. Whenever a new query is made, the attributes are automatically adapted according to the user's current location and temporal contexts.

Similarly, we also added the option of automatically storing the number of visits to specific points of interest, so that the application could identify which ones might be more important to the user and those that have never been visited.

Finally, it is important to stress out the need for the user to always have the option of over imposing the options automatically chosen by the application with the ones explicitly chosen by him/her.

Temporal Distances

As important as understanding what points of interest exist in the vicinity of the user is to identify which of these are open by the time the user gets there. As an example, if the user is searching for a service station to fuel his car, it is not useful to display results for stations that might not be open when the user finally arrives there.

For this reason, we have added a new temporal distance function and time attribute to the degree of interest function. This distance can be subdivided into three different distance calculations, depending on the category of the PoI being searched. On these calculations h_{op} and h_{cl} are, respectively, the point of interest opening and closing hours, h_{ar} is the users expected arrival time (either specified by the user or automatically calculated according to the users current location), Δt_{st} is the minimum staying time interval and Δt_{tol} is a tolerance time interval (both can be specified by the user, but have default values predetermined according to the point of interest category).

- The user intends to arrive only during opening hours, never before or after

$$Dist_{temp} = \begin{cases} 0, if\ h_{op} \leq h_{ar} < (h_{cl} - \Delta\ t_{st}) \\ 1, if\ h_{ar} \geq (h_{cl} - \Delta\ t_{st}) \vee h_{ar} < h_{op} \end{cases}$$

- The user intends to arrive before (using a tolerance time interval) or during opening hours, never after

$$Dist_{temp} = \begin{cases} 0, if\ h_{op} \leq h_{ar} < (h_{cl} - \Delta\ t_{st}) \\ \dfrac{h_{op} - h_{ar}}{\Delta\ t_{tol}}, if\ h_{ar} < h_{op} \wedge h_{ar} \geq (h_{op} - \Delta\ t_{tol}) \\ 1, if\ h_{ar} \geq (h_{cl} - \Delta\ t_{st}) \vee h_{ar} < (h_{op} - \Delta\ t_{tol}) \end{cases}$$

- The user intends to arrive before opening hours (using a tolerance time interval), never during or after

$$Dist_{temp} = \begin{cases} \dfrac{h_{op} - h_{ar}}{\Delta\ t_{tol}}, if\ h_{ar} \geq (h_{op} - \Delta\ t_{tol}) \wedge h_{ar} < h_{op} \\ 1, if\ h_{ar} > h_{op} \vee h_{ar} < (h_{op} - \Delta\ t_{tol}) \end{cases}$$

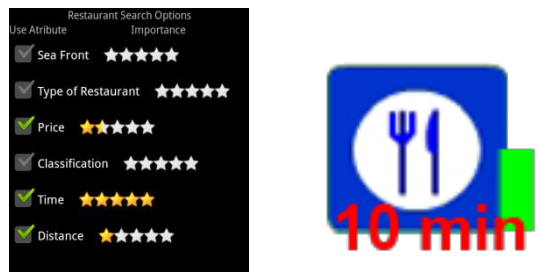
PROTOTYPE

To be able to evaluate the proposed changes to the degree of interest function, we have developed a prototype, for a Samsung Galaxy S mobile device with the Android 2.2 operating system.

The prototype was developed using the Chameleon Adaptive Visualization Framework for mobile devices previously developed [11].

We use a PoI database with data obtained from several collaborative internet sites, aimed at navigation applications, which allowed us to obtain over 8000 PoI, divided in eight categories, with an accurate geographic distribution.

Figure 1 (a) shows an example of the query specification interface. The user is able to select which attributes are explicitly specified and also change the default weight for each of them. As an example, in the figure, the user has selected price and distance with a low weight and the temporal distance with a higher weight.



(a) **(b)**

Figure 1. (a) Query specification interface (b) Point of interest icon.

The results interface displays a map with icons representing each point of interest. These icons are composed of three different areas (Figure 1 (b)): the main area indicates the category of the displayed points of interest; in the right, using the approach proposed in [12], a green bar indicates the relevance value calculated using the DoI function (the higher the bar the more relevant the result); in the lower part of the icon, a number indicates how much time the user would need to get there.

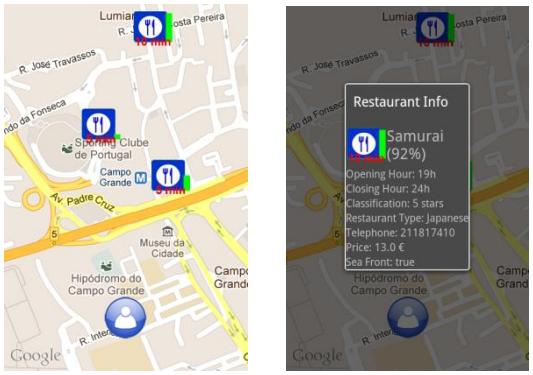
The DoI function can be used in three different recommendation modes: a Standard mode that uses the DoI function without the historical context to recommend the PoI, the Exploratory mode that removes previously visited PoI from the recommendations, and the Adaptive mode that uses the historical context, adapted to the users current location and time, to determine the PoI recommendations.

The Standard mode uses the previous DoI function (enhanced with the temporal distance) without the historical context. In Figure 2 (a) an example is given. In this case the user is searching for a restaurant but has only specified the geographical distance attribute to be used. For this reason, the relevance of each PoI decreases as the geographical distance to the user increases.

The Exploratory mode is intended to be used when the users want to "try something new". It uses the information about previously visited PoI to remove the ones already known to the user. In the example, in Figure 2 (b), two of the restaurant had already been visited previously by the user and were filtered out.

(a) (b)

Figure 2. (a) Standard DoI results (b) Exploration results



(a) (b)

Figure 3. (a) Adaptive DoI results (b) Information about the selected PoI

Finally, the Adaptive mode uses the adaptive historical log to adapt unspecified attributes to the previous queries done by the user, depending on the current temporal and location contexts. In the example, in Figure 3 (a), the user had a previous experience of going mostly to Japanese restaurants for lunch while in his work area. For this reason, the restaurant at the top of the screen (which is Japanese) has a higher relevance. Figure 3 (b) shows contextual on-demand information about the selected restaurant.

CONCLUSIONS AND FUTURE WORK

In this paper we have proposed an adaptive degree of interest function that improves a previously proposed relevance function. This function uses information about historical logged previous queries to be able to adapt new point of interest searches to the user interests.

The next step in our work is to perform an extensive user evaluation of the proposed DoI function and the developed prototype. We also intend to explore the use of additional context dimensions, such as weather conditions or the type of user movement, to further adapt the DoI function. We also intend to explore the use of different adaptive icons that change dynamically according to the contexts and explore the automatic detection and definition of the user's relevant geographical areas.

ACKNOWLEDGMENTS

The work presented here is based on research funded by the FCT - Fundação para a Ciência e Tecnologia (Portuguese Science and Technology Foundation) through the SFRH/BD/46546/2008 scholarship.

REFERENCES

1. Google Maps Mobile. http://www.google.pt/mobile/maps/.
2. Heimonen, T. Information Visualization on Small Display Devices. *Master Thesis. Department of Computer Sciences, University of Tampere,* (2002).
3. Holtzblatt, K. Designing for the Mobile Device: Experiences, Challenges and Methods. In *Communications of the ACM, 48 (7),* (2005), 33-35.
4. Adomavicius, G., Mobasher, B., Ricci, F., and Tuzhilin, A. Context-Aware Recommender Systems. In *AI Magazine, 32 (3),* (2011), 67-80.
5. Reichenbacher, T. Mobile Usage and Adaptive Visualization. In *Encyclopedia of GIS. Part 16. Heidelberg: Springer Verlag,* (2008), 677-682.
6. Furnas, G. W. Generalized Fisheye Views. In *Proc. of the ACM CHI'86.* (1986), 16-23.
7. Kheim, D. A., and Kriegel, H.-P. VisDB: Database Exploration Using Multidimensional Visualization. *IEEE CG&A, 14 (5)* (1994).
8. Reichenbacher, T. Mobile Cartography – Adaptive Visualization of Geographic Information on Mobile Devices. *München. Verlag Dr. Hut.* (2004).
9. Carmo, M. B., Afonso, A. P., Pombinho, P. and Vaz, A. Visualization of Geographic Query Results for Small Screen Devices. *In Proc. of the Visual 2008, LNCS 5188.* (2008), 167-178.
10. Pombinho, P., Carmo, M. B. and Afonso, A. P. Evaluation of Overcluttering Prevention Techniques for Mobile Devices. *In Proc. of the IV'09.* (2009), 127-134.
11. Pombinho, P., Afonso, A. P., and Carmo, M. B. Chameleon – A Context Adaptive Visualization Framework for a Mobile Environment. In *Proc. of the IV'11.* (2011), 151-157.
12. Burigat, S., and Chittaro, L. Interactive Visual Analysis of Geographic Data on Mobile Devices based on Dynamic Queries. In *Journal of Visual Languages and Computing, 19 (1).* (2008), 99-122.

AUTHOR INDEX