diffeRS: a Mobile Recommender Service

Lucia Del Prete
Dept. of Computer Science
University College London
Gower Street, London WC1E 6BT, UK
Email: L.DelPrete@cs.ucl.ac.uk

Licia Capra
Dept. of Computer Science
University College London
Gower Street, London WC1E 6BT, UK
Email: L.Capra@cs.ucl.ac.uk

***Abstract*—Thanks to advances in mobile technology, modern mobile devices have become essential companions, assisting their users in attaining their daily tasks. It will not be long before these devices will become *recommending companions*, advising users about what data (e.g., restaurants) and what services (e.g., podcast channels) they may enjoy in the local area at the present time. Because of the very nature of the items (both data and services) being suggested (i.e., *location* dependent and *mobile* with respect to the consuming user), recommendations cannot be computed on central servers and then pushed to the users. Rather, a novel *decentralised* mobile recommender service will have to be developed and deployed; instead of relying on global knowledge about users' profiles, such service will have to exploit the wisdom of local communities to compute recommendations. Moreover, because of resource limitations of mobile devices, the algorithms it will employ will have to be computationally light. In this paper, we propose *diffeRS*, a totally decentralised mobile recommender service specifically designed for pervasive environments. diffeRS crafts a virtual view of the local community's preferences, by exchanging users' profiles via radio technology (e.g., Bluetooth) during periods of colocation. Profiles are stored locally and recommendations are computed using a lightweight algorithm. As our experimental evaluations demonstrate, diffeRS achieves an accuracy and coverage that are comparable to those of centralized recommender systems in use today. .**

I. Introduction

Modern mobile devices have seen their computing capabilities grow according to Moore's law. Additional functionalities, such as digital cameras, MP3 players, and GPS receivers, have been integrated on such devices, together with a variety of wireless network technologies of increasing bandwidth, enabling the on-the-fly creation of networks of devices in proximity. As a result, these devices have become essential companions, assisting their users in attaining their daily tasks.

A plethora of applications are being developed for these devices that, in a not so far future, could be exposed as services to applications (and services) running on other devices in the surroundings. Services could include, for example, podcast channels, route finders, events listings, etc. Besides digital services, an abundance of data is becoming accessible to mobile users, ranging from information about the surroundings (e.g., local amenities, shops, gigs, events) to information that colocated users generate themselves and are willing to share (e.g., microblogs, pictures, videos, etc.). A challenge thus arises as to how to assist mobile users in finding what services and/or data they might enjoy in this abundance.

Recommender systems [2] have successfully addressed the problem of information overload [19] on the Web: users' profiles (i.e., sets of 'ratings' about some items, such as books) are *centrally* collected on powerful servers; using algorithms such as k-Nearest Neighbour Collaborative Filtering [8], profiles are processed and accurate recommendations computed. While the effort to build, host, run and maintain a traditional centralised recommender system is well justified by the business benefit of appealing to a large crowd, it is unlikely to be so for pervasive scenarios. In fact, pervasive items (both services and data) are, by their own nature, *local*, that is, of interest only to people in the surroundings; as such, rather than being universally accessible, they will only serve relatively small local communities. Apart from being local, pervasive items are also *mobile*, as either offered by devices embedded in buildings that users transit through, or by users to users while on the go via their portable devices. A centralised recommender service would thus have to continuously monitor both users' *and* items' locations, in order to compute meaningful recommendations, something which is practically infeasible. Even if such challenge could be solved in convenient ways, there is another issue worth considering: as we consume pervasive items continuously along the day, a centralised recommender service which monitors users' and items' location would end up knowing a great deal about us (i.e., not just our music or shopping tastes, but also our habits); there would hence be a much greater opportunity for users to have, or to feel, their privacy as being violated.

To overcome these weaknesses, we propose *diffeRS*, a fully distributed mobile recommender service for pervasive computing. Users maintain their own profile (i.e., ratings about previously consumed pervasive items - services and data) locally on their mobile device. Using radio technology (e.g., Bluetooth), profiles are exchanged between devices during periods of colocation, so that each user gradually builds a virtual view of the local community's preferences. When a recommendation has to be computed, diffeRS dynamically assesses whether the user is *mass-like minded* or *individual*: in the former case, no sophisticated recommendation algorithm is required, and the mean rating for any given item is returned; in the latter case, a novel algorithm is used instead . We have conducted an extensive evaluation of diffeRS. Our results demonstrate that diffeRS achieves an accuracy and coverage that are comparable to those of centralized recommender systems in use today.

The reminder of the paper is structured as follow: we begin with an overview of what a recommender system is, and discuss the state of the art in mobile recommender systems research (Section II). We then present *diffeRS* (Section III). The results of an extensive evaluation are discussed (Section IV), before presenting our concluding remarks (Section V).

II. Background

Recommender systems have been designed in response to a well-known problem: information overload. Rather than letting users browse through an infinite collection of information, recommender systems present users with a short list of items that they may enjoy. Collaborative filtering (CF) [16] [11][9][17] and especially user-based CF, has established itself as the industry de-facto standard for generating these recommendations. The mechanics behind it are quite simple in principle: a user's profile is represented as a vector, containing a rating value (in a rating scale, e.g., $[1, 5]$) for each item that the user has consumed/rated in the past from a given catalogue of items; profiles from all users are centrally collected and organised in a user-by-item rating matrix. Whenever a user a is looking for a recommendation, the CF recommender system first identifies those users who are deemed most similar to a (i.e., k-Nearest Neighbours), by computing a similarity metric (e.g., cosine similarity, Pearson correlation) between their respective profiles. A score for each item that a has yet to consume is then computed, so that those items that have received a high rating by similar users are ranked higher, and eventually recommended.

Research in web-based recommender systems has been extremely active in the past decade, trying to deliver ever more accurate and scalable techniques. Some research has been looking to decentralised recommenders in the context of traditional fixed P2P networks [7][21]. However, as highlighted in the previous section, their applicability to the mobile and pervasive setting is limited. Research on recommender systems specifically developed for these new environments is still in its early days. At first, researchers have been looking at ways to improve the usability of traditional recommender systems on mobile devices [13]. Since then, research on mobile recommender systems and their specific challenges has started to gain increasing interest. Whilst traditional recommender systems rely on almost unlimited processing and memory resources to store and process rating matrixes, mobile devices are severely constrained in this regard. Consequently, they cannot persist all the information (about users/items/ratings) that traditional recommenders store in centralized repositories; nor can they afford to process large amounts of data. To circumvent this problem, semi-decentralised approaches have been proposed and shown to be suitable for peer-to-peer mobile settings [4][20][14]. Very little work has been done to target decentralised pervasive environments. In this domain, not only the information upon which recommendations can be computed is much scarcer than that gathered in centralised repositories, but also – unlike decentralised recommenders for traditional settings – resources are limited. The few works that exist in the area of totally decentralised recommender systems for mobile environments have thus focused on ways to compute accurate recommendations while relying of sparse information: in [5], for example, users' similarity is computed based on the amount of time such users spend in the same places at the same time (the more time is spent in the same places, the more similar the users are); such an assumption is dubious and, as the approach has never been evaluated, its accuracy cannot be assessed. In [6], users' profiles are made of keywords expressing interests, rather than ratings on items In this case, the accuracy of the approach strictly depends on the nature of the items being recommended, and cannot be extended to domains where the correlation between interest keywords and tastes is loose. Rather than re-defining users' similarity metrics, in [18] an epidemic protocol to propagate users' profiles and compute (traditional) users' similarity is proposed instead; while we support the choice of using well-established and thoroughly-assessed users' similarity metrics, their evaluation is rather limited: in fact, unrealistic mobility models are being used where an incredibly high number of interactions with randomly chosen users occur (thus facilitating information dissemination), while in real human networks encounters are much fewer and non-random [10].

In the following sections, we present *diffeRS*, a fully decetralised mobile recommender system, specifically designed to run efficiently on mobile devices, deployed in human pervasive networks. As our evaluation will demonstrate, diffeRS is capable of providing an accuracy and coverage comparable to those of centralised recommender systems.

III. diffeRS

diffeRS is a fully decentralised recommender service that runs on users' mobile devices and that targets, as recommendable items, those services and content that are local and mobile with respect to the user itself. diffeRS has been built based upon the following three observations:

1) Rating information for pervasive services and content is meaningful only to people that are (or will be) in the same surroundings. It does make sense then to distribute rating information only to local users, rather then broadcasting it multi-hop across all users.
2) Any rating community is formed by two and fundamentally different groups of users: mass-like minded users and individual users (i.e., users that have rather atypical tastes). As Mark Penn remarked [12], the tastes of mass-like minded users (macro-trends) are easy to notice and do not necessarily require clever engines to highlight and exploit them; on the contrary, patterns within the preferences of individual users (micro-trends) are counter-intuitive, difficult to discover and have the potential to elicit behaviors otherwise extremely difficult to understand and guess. Hence, recommender systems should focus on comprehending and predicting the tastes of atypical users.

3) The more frequently a user conforms to the rating community she belongs to, the more the community represents her, and viceversa. Paradoxically, in traditional recommender systems, popular items and trend followers risk driving predictions for atypical users too. This is because items of universal appeal are those for which more users' feedback exists (i.e., they appear in the profiles of many users, and thus drive the quantification of users' similarity as the only items in overlap).

diffeRS exploits these observations as follow: first, rating profiles are only exchanged between colocated people (i.e., one hop) during encounters (observation 1); a virtual view of the local community is then built on each device running diffeRS. Second, users are dynamically distinguished between mass-like minded and individual, simply by looking at the average deviation of their profile from the preferences expressed by the community as a whole (observations 2 and 3). Finally, for mass-like minded users, predictions are simply computed as the average of the preferences expressed by the rating community they belong to. For atypical users, instead, a user-based CF approach is used; however, recommenders are not searched within the whole rating community but only among those other atypical users who most similarly deviate from the rating community (i.e., those users who are *similarly different*).

This section is further structured as follow: we first introduce the abstract data model used within diffeRS (Section III-A), and then present the details of diffeRS recommending engine (Section III-B).

A. Abstract Data Model

In diffeRS, a user's profile is simply a vector of ratings. During encounters, users' vectors are exchanged, so that each user builds, over time, a local rating matrix P containing the ratings of other users in the surroundings (which we call *virtual community*). Such matrix approximates the global rating matrix used in traditional recommenders. Unlike conventional recommender systems, diffeRS does not exploit the whole local rating matrix P for computing recommendations. This is because mass-like-minded users (or trend followers) U^M are in agreement with the rating community; as such, it should suffice to predict their preferences as the community mean ratings. We store these mean ratings in a vector that we call Community Profile u_C.

By definition, individuals (or atypical users) U^I are in disagreement with the rating community and, consequently, with mass-like minded users. It hence makes very little sense to seek recommenders for individual users among trend followers. When predicting ratings for individual users, we thus exclusively seek the advice from other atypical customers; we do so by exploiting a sub-matrix P^I of the local rating matrix P that contains the preferences expressed by users in U^I only.

As a result, diffeRS conceptually decomposes a rating matrix P in a vector, the Community Profile u_C (used to predict preferences for U^M), and a smaller rating matrix P^I (used to predict preferences for U^I), as shown in Fig. 1.

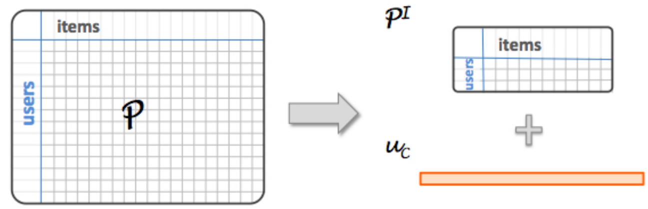

Fig. 1. Rating Matrix Decomposition

B. Recommending Algorithm

Based on the abstract data model presented above, we now describe step-by-step how recommendations are computed. Being u_a the profile of active user a for whom a prediction for item i needs to be computed, traditional user-based CF performs three steps: compute users' similarity to a for all known users (step 1), determine a's neighbours (i.e., recommenders) as those with higher similarity (step 2), and compute a predicted rating for item i based on the preferences expressed by a's neighbours for i (step 3). diffeRS proposes two key changes to this approach: as a pre-step (step 0), it determines whether a is a mass-like minded user or an individual one. It does so by calculating u_a's average deviation d_a from the community profile vector u_C; if the deviation is lower than a constant parameter α, a is deemed to be mass-like minded, so that the predicted rating for item i is simply the community average opinion $u_C[i]$. If the user is deemed individual, the three-step user based CF is performed instead, but with a major difference: rather than using matrix P (i.e., all users and all ratings), it uses the smaller rating matrix P^I. In so doing, it computes the similarity between a and atypical users only, and derive a recommendation based on their opinions alone. The computational complexity of diffeRS is thus considerably reduced, as heavy calculations (i.e., users' similarity) are performed exclusively for 'difficult to predict' people, and using considerably smaller amounts of data. A more detailed description of diffeRS rating prediction algorithm steps follows.

Step 0 - User Classification. Each rating a user expresses conveys some information about the degree to which such user conforms to the rating community she is part of. As the rating community is represented by the items' average ratings (i.e., by the Community Profile u_C), we quantify the overall user *deviation* from the community as the average of the absolute difference between the ratings that such user has expressed and the average feedback for the same items. Being u_a the profile of user a, u_C the community profile, and I_a the set of items for which a has expressed a preference, then:

$$d_a = \frac{1}{|I_a|} \sum_j |u_a[j] - u_C[j]| \tag{1}$$

We use this last to classify people as either mass-like minded or individual users: if the deviation for a given user is lower or equal to a constant parameter $\alpha \in [0, \Delta_r]$, diffeRS classifies the user as mass-like minded, otherwise as individual (with Δ_r being the maximum difference between two rating values).

Step 1 - Computing User Similarities. If a user is deemed individual, the next step is to find suitable recommenders. We do so by looking at how other individual users *similarly differ* from the rating community. This difference is crucial: unlike traditional correlation models (e.g., Cosine and Pearson correlations) that aim to highlight similar rating behaviours within the community as a whole, here we ponder the differences that users manifest against the same rating community instead. In practice, we quantify the similarity $w_{a,k}$ between the active user a and another individual user $k \in U^I$ as follow: first, we compute the difference in their ratings for each item j they have co-rated $|u_k[j] - u_a[j]|$; we then sum these differences (normalised by the maximum differences between two rating values in the rating scale) to infer a measure of users' correlation that boosts 'similarly different' tastes. Finally, we use the Jaccard index $\frac{|I_a \cap I_k|}{|I_a \cup I_k|}$ to weigh the reliability (or confidence) of the computed similarities. Being I_a and I_k the set of items rated by a and k respectively, and Δ_r the maximum rating difference, we compute the similarity between users a and k as:

$$w_{a,k} = \frac{1}{|I_a \cup I_k|} \sum_{j \in I_a \cap I_k} \left(1 - \frac{|u_k[j] - u_a[j]|}{\Delta_r}\right) \qquad (2)$$

Note that the above correlation formula consists of sums rather then products, thus being a computationally lighter formula to deal with.

Step 2 - Selections of Neighbours. In traditional user-based CF systems, neighbours for an active user a are selected by either considering the k nearest neighbours (i.e., k most similar users) or by considering all users whose similarity to a is greater than a given threshold. We take a slightly different approach and consider as neighbours *all individual users* who have rated at least one item in common with the active user a. We do so to avoid further aggravating the data sparsity problem that CF systems suffer from.

Step 3 - Weighted Average. Finally – being $\overline{r_k}$ is the average rating for user k – the predicted rating of the active user a for item i is estimated using a weighted average:

$$r_{a,i} = \overline{r_a} + \frac{\sum_{k \in Neighbourhood} w_{a,k} * (u_k[i] - \overline{r_k})}{\sum_{k \in Neighbourhood} w_{a,k}}. \qquad (3)$$

IV. Evaluation

We have evaluated diffeRS both as a centralised recommender system and as a mobile one. In the former case, we are interested in assessing its accuracy with respect to traditional recommender systems; in the latter case, we are interested in comparing its accuracy and coverage in a realistic pervasive deployment, with respect to a centralised diffeRS deployment. Note that, in the former case, we are not interested in coverage simply because diffeRS can only improve that of traditional recommenders (i.e., by returning items' averages in the case users' neighbours are not found).

In order to mimic a realistic deployment, we have used two datasets: MovieLens Light [15], to model users' ratings, and MIT Reality Mining [1], to model users' encounters. MovieLens is a web-based research recommender system that debuted in late 1997. The data was collected by the GroupLens Research Project at the University of Minnesota during the seven-month period from September 19th, 1997 through April 22nd, 1998, and has been widely used by the recommender systems research community to validate results. We have employed a subset of the whole MovieLens rating data known as MovieLens Light. This is a subset of 943 randomly selected users, providing 100,000 ratings (in a scale 1 to 5) of 1682 movies. To model encounters, we have elected the MIT Reality Mining dataset as our reference mobile scenario as it represents what we consider a typical pervasive computing setting (i.e., a university campus, with some services being available centrally, others accessible from devices embedded in buildings, and others still from peer Bluetooth devices). This dataset records Bluetooth encounters among one hundred users (MIT staff and students) to whom Nokia 6600 were given over a period of nine months.

Experiment 1 - diffeRS as a Recommender System. The first set of experiments we have conducted aimed to assess the accuracy of diffeRS prediction algorithm, irrespective of the mobile setting. To do so, the MovieLens Light rating data has been split in 90% training set and 10% test set; the split has been repeated 100 times at random. For each item in the test set, a prediction has been computed using *diffeRS*, and the Mean Absolute Error (MAE) recorded. This error has then been compared with that obtained when making predictions using Pearson correlations weighted by Jaccard index (in the following, *jpCF*), as this has been shown to provide better results than the traditional Pearson coefficients alone [3]. We have repeated these experiments for different values of k nearest neighbours (for *jpCF*). In all experiments, we have used diffeRS with $\alpha = 1$. In the following we report the results for the cases where we assess diffeRS prediction accuracy across all users (Fig. 2a), just mass-like minded users or individuals separately (Fig. 2b).

As shown, *diffeRS* performs better than *jpCF* for any value of k when considering all users. The gap in accuracy is not trivial: while the MAE for *jpCF* is in the range $[0.85, 0.87]$

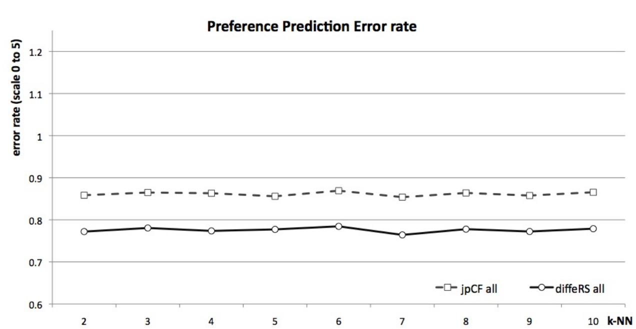

Fig. 2a. Prediction Accuracy for All Users

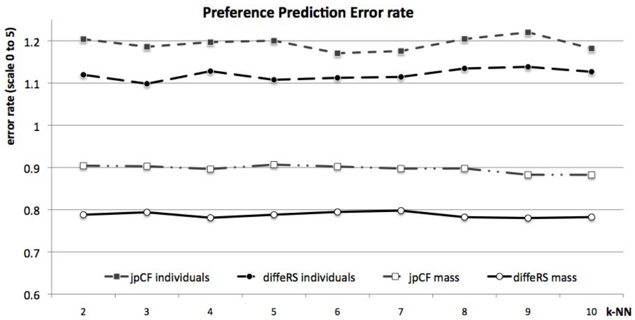

Fig. 2b. Prediction Accuracy for Mass-Like-Minded Users and Individuals

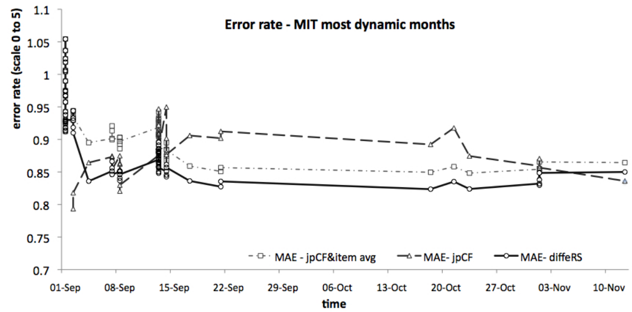

Fig. 3a. Most Dynamic Months - Accuracy

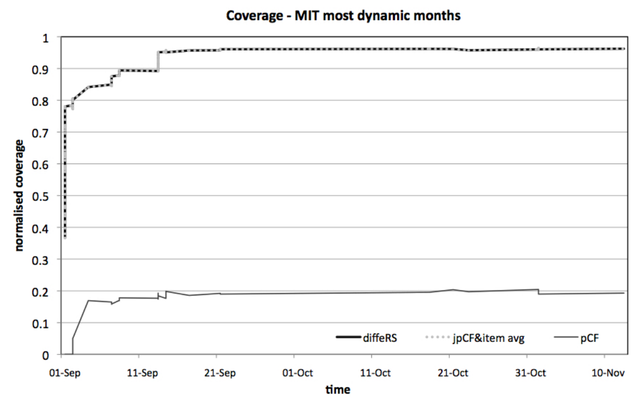

Fig. 3b. Most Dynamic Months - Coverage

for any k, that of *diffeRS* is in $[0.76, 0.79]$. We also observe that *diffeRS* improvement in accuracy is greater for mass-like minded users, confirming our hypothesis that, for this category of users, there is no need to run computationally expensive algorithms, and simple rating averages will do just fine. As expected, atypical users are the most difficult to predict; nonetheless, the MAE reported by *diffeRS* is consistently lower than that reported by *jpCF* even for this category of users. Also to note, the limited dependence on the value of k both for *diffeRS* and *jpCF*; this may be due to the use of Jaccard correlation as a measure of the reliability of similarity coefficients (as well as an additional measure of similarity). In fact the Jaccard correlation allows to achieve more reliable similarity measures and users' neighbourhoods. Hence the benefit of enlarging the size of users' neighbourhoods may be then compensated by the polluting effect of pondering the preferences of users that are less similar to the active one.

Experiment 2 - diffeRS as a Mobile Recommender System. The second set of experiments aimed to assess both accuracy and coverage of diffeRS when deployed in a totally distributed mobile setting. To do so, we have randomly mapped 100 users from the MovieLens Light rating dataset to the 100 users in the MIT Reality Mining dataset, making sure that the resulting ratings-per-user distribution followed the ratings-per-user distribution of the whole MovieLens Light dataset. At the beginning of the simulation, each user only knows its own ratings; upon encounters, users exchange their own rating profile. At regular intervals of time, predictions were computed based on the profiles so far collected. We have then compared accuracy (i.e., MAE) and coverage of three algorithms: *diffeRS*, Pearson correlations weighted by Jaccard index (*jpCF*, as before), and (for fairness) Pearson correlations weighted by Jaccard index while returning item averages if a prediction could be not made (*jpCF&item avg*). For the last two algorithms, we have set the number of neighbours k to be all users encountered thus far, otherwise both accuracy and coverage were severely compromised due to data sparsity. We report results for two distinct periods of time: the three months (Sep-Oct-Nov) where the largest number of encounters were made, and another three months (Feb-Mar-Apr) where users' encounters are representative of the whole set in that respect. Results for accuracy and coverage are reported in Fig. 3a and 3b for the former period, and in Fig. 4a and 4b for the latter. Note that the last datapoint for the former period is November 10th (instead of November 30th), and February 28th for the latter (instead of April 30th); this is because, after these dates, no new encounters occur (i.e., devices meet the same other devices again and again). As no new knowledge is gained (i.e., no new profiles are being exchanged), accuracy and coverage flatten to the last datapoint plotted.

As one would expect, the greater the number of users' profiles exchanged (during the most dynamic months), the better both the coverage (higher) and the accuracy (lower MAE), irrespective of the algorithm used. We note also that *jpCF* achieves a coverage that is extremely limited compared to the other algorithms. In fact, whilst coverage for *diffeRS* and *jpCF&item avg* is about 95% during the most dynamic months and 90% during representative ones (on average during the whole period), it is just 19% and 14% for *jpCF*. This is because it is the only algorithm not exploiting item averages when users' neighbourhoods cannot be computed.

When looking at accuracy, we notice that *diffeRS* constantly provides better estimates than *jpCF&item avg* with a weighted average MAE of 0.841 and 0.868 (for the most active period and for the representative period respectively), thus improving over the 0.863 and 0.905 of *jpCF&item avg*. diffeRS also provides on average better accuracy than *jpCF* (whose MAE is 0.874 and 0.889 for the two periods, under the same experiment setups). Most importantly, the overall

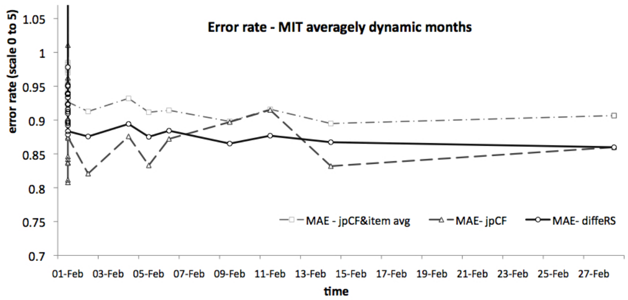

Fig. 4a. Representative Months - Accuracy

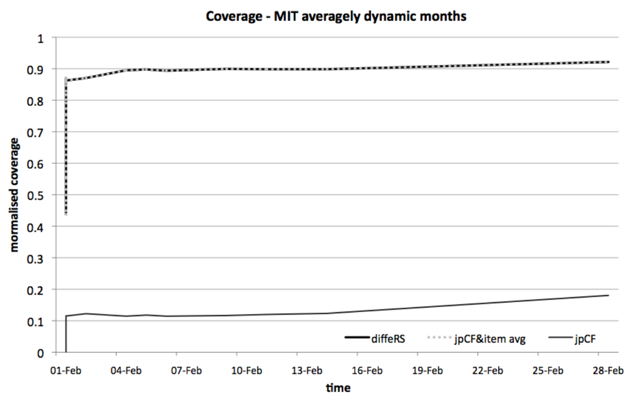

Fig. 4b. Representative Months - Coverage

performance of *diffeRS* in a totally decentralised mobile setting is comparable to that observed using a centralised deployment, thus demonstrating its suitability as a mobile recommender.

V. Discussion and Future Work

In this paper we have presented diffeRS, a fully distributed recommender system specifically designed from the outset to be deployed on mobile devices in pervasive computing scenarios. The main observation underpinning diffeRS is that, for mass-like-minded users, there is no need to run expensive algorithms, and item preferences can be simply (yet accurately) predicted using rating averages. For atypical users instead, diffeRS uses a novel algorithm and companion data structure that, during extensive experimental and analytical evaluation, have proved to provide accurate recommendations at very low computational/memory cost.

We have so far evaluated the accuracy and coverage of diffeRS when creating multiple random overlays between users in the MIT Reality Mining dataset and those in the MovieLens Light users. We plan to conduct further evaluations in two main directions: first, we intend to consider rating datasets of items of pervasive (local) nature (i.e., restaurants in a chosen city), to see if their characteristics (e.g., rating distribution, sparsity) are any different, and in this case, study their effects on accuracy and coverage. Second, the lack of a single dataset containing both users' movement and rating data has forced us to experiment with artificial overlays; we now intend to study the effects of different non-random overlays on our metrics (e.g., users with big rating profiles overlayed onto users with high mobility - as this will favour information dissemination, and viceversa).

References

[1] The Reality Mining Project. http://reality.media.mit.edu/.

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

[3] Laurent Candillier, Frank Meyer, and Françoise Fessant. Designing specific weighted similarity measures to improve collaborative filtering systems. *Advances in Data Mining. Medical Applications, E-Commerce, Marketing, and Theoretical Aspects*, pages 242–255, 2008.

[4] Rickard Coster and Martin Svensson. Incremental collaborative filtering for mobile devices. In *Proc. of the 2005 ACM Symposium on Applied computing*, pages 1102–1106, Santa Fe, New Mexico, USA, 2005.

[5] Alexandre de Spindler, Re De Spindler, Moira C. Norrie, Michael Grossniklaus, and Beat Signer. Spatio-temporal proximity as a basis for collaborative filtering in mobile environments. In *Proc. of the Workshop on Ubiquitous Mobile Information and Collaboration Systems*, pages 912–926, 2006.

[6] Patrick Gratz, Adrian Andronache, and Steffen Rothkugel. Ad hoc collaborative filtering for mobile networks. In *Proc. of the Sensor Networks, Ubiquitous, and Trustworthy Computing Conference*, pages 355–360, Washington, DC, USA, 2008. IEEE Computer Society.

[7] Peng Han, Bo Xie, Fan Yang, and Ruimin Sheng. A scalable p2p recommender system based on distributed collaborative filtering. *Expert systems with applications*, 2004.

[8] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proc. of the 22nd International ACM Conference on Research and Development in Information Retrieval*, pages 230–237, 1999.

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

[10] Seongik Hong, Injong Rhee, Seong J. Kim, Kyunghan Lee, and Song Chong. Routing performance analysis of human-driven delay tolerant networks using the truncated levy walk model. In *Proc. of the 1st ACM Workshop on Mobility Models*, pages 25–32, 2008.

[11] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[12] Penn Mark and Zalesne Kinney. *Microtrends: the small forces behind tomorrow's big changes*. Twelve, 2007.

[13] Bradley N. Miller, Istvan Albert, Shyong K. Lam, Joseph A. Konstan, and John Riedl. Movielens unplugged: Experiences with an occasionally connected recommender system. In *Proc. of the Intelligent User Interfaces Conference*, pages 263–266. ACM Press, 2003.

[14] Bradley N. Miller, Joseph A. Konstan, and John Riedl. Pocketlens: Toward a personal recommender system. *ACM Transactions on Information Systems*, 22(3):437–476, July 2004.

[15] GroupLens Research. MovieLens. www.movielens.org.

[16] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proc. of Computer Supported Cooperative Work Conference*, pages 175–186. ACM Press, 1994.

[17] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John T. Riedl. Application of dimensionality reduction in recommender system – a case study. In *Proc. of ACM Web Mining for E-Commerce Workshop*, 2000.

[18] Rossano Schifanella, André Panisson, Cristina Gena, and Giancarlo Ruffo. Mobhinter: epidemic collaborative filtering and self-organization in mobile ad-hoc networks. In *Proc. of the ACM Conference on Recommender systems*, New York, NY, USA, 2008.

[19] Alvin Toffler. *Future Shock*. Bantam, 1984.

[20] Amund Tveit. Peer-to-peer based recommendations for mobile commerce. In *Proc. of the 1st International Workshop on Mobile Commerce*, pages 26–29, New York, NY, USA, 2001. ACM Press.

[21] Jun Wang, Johan Pouwelse, Reginald L. Lagendijk, and Marcel J. T. Reinders. Distributed collaborative filtering for peer-to-peer file sharing systems, 2006.