View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by White Rose Research Online

The University Of Sheffield.

This is a repository copy of *Recommending access points to individual mobile users via automatic group learning*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/126298/

Version: Accepted Version

Proceedings Paper:
Partov, B., Leith, D.J. and Checco, A. orcid.org/0000-0002-0981-3409 (2017) Recommending access points to individual mobile users via automatic group learning. In: IEEE International Conference on Communications. 2017 IEEE International Conference on Communications (ICC), 21-25 May 2017, Paris, France. IEEE . ISBN 9781467389990

https://doi.org/10.1109/ICC.2017.7997424

© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works. Reproduced in accordance with the publisher's self-archiving policy.

Reuse
Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Recommending Access Points to Individual Mobile Users via Automatic Group Learning

Bahar Partov [1,2], Douglas J. Leith[1], and Alessandro Checco[3]
[1] Trinity College Dublin, Ireland, [2] SENSEable City Lab, Massachusetts Institute of Technology, USA
[3] University of Sheffield, U.K.

Abstract: We consider user to cell association in a heterogeneous network with a mix of LTE/3G and WiFi cells. Individual user preferences are often neglected when a user to cell association decision is made. In this paper we propose use of a recommender system to inform the mapping of users to cells. We demonstrate the effectiveness of the proposed grouped-based user to cell associations for a set of synthetically generated user/cell ratings.

I. Introduction

In this paper we study the use of collaborative filtering based recommender systems to assist users with wireless access point selection. In metropolitan areas there is widespread availability of both cellular and WiFi services. LTE/3G coverage is ubiquitous in urban areas. Many enterprises, schools, and cities provide WiFi services to individuals. Hotspot directories report large numbers of WiFi access points in urban areas, e.g. jiwire in the US reports 400 to 1000 commercial WiFi networks in each of the top ten U.S. metropolitan areas [1] and the Fon service has aggregated more than 3 million access points in the UK alone [2]. This is in addition to home-based WiFi services. Currently almost all mobile devices possess both cellular and WiFi interfaces, and smartphone users can, and do, switch among different WiFi access points and their cellular connection. Users therefore often have a great deal of choice but, currently, little information on which to base this decision other than the signal level bars displayed to them by their handset. Note that from now on we will use access point to interchangeably mean a WiFi access point or a cellular connection since our recommendation approach is agnostic to the wireless technology used. Unfortunately, signal strength by itself is often not a good indicator of the usefulness of an access point to a particular user: access points might block certain applications [3], may have different terms and conditions regarding user privacy, cost of the service etc and in other ways may have poorer performance than that proclaimed by the signal strength [4]. It is this observation which motivates the provision of recommendations to a user as to which of the access points available in a location are likely to be most satisfactory for that particular user.

The setup we consider is illustrated schematically in Fig 1. We have a set of n users and m access points. After user u makes use of an access point v they rate the utility provided to them – this is a matter of personal preference that reflects the dollar price paid, ease of use, QoS, applications supported and so on. Gathering these ratings together gives an $n \times m$ ratings matrix $\boldsymbol{R}$, where element R_{uv} is the rating by user u of access point v. This matrix is typically sparse, since each user individually may rate only a small number of access points. Our aim is to estimate the missing entries in $\boldsymbol{R}$, i.e. to provide predictions of the ratings a user would make for access points which are currently unrated. Those access points which are predicted to have a high rating are then recommended to a user. In addition to the sparseness of rating matrix $\boldsymbol{R}$, which is common in recommender systems, challenges include the heterogeneity of the user population and the sensitivity of ratings (and so recommendations) to user location. It is these challenges which we seek to address in this paper.

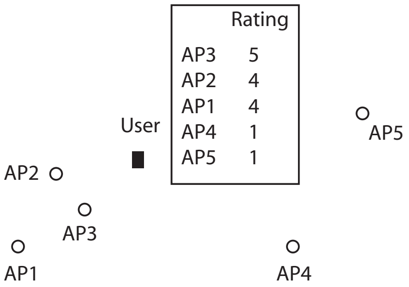


Fig. 1: Illustrating the setup considered. A user is in an area with multiple access points (WiFi access points and LTE/3G cellular connections). In the past the user has rated a subset of these access points, as have other users. These ratings reflect user satisfaction and take account of aspects such as cost, applications blocked/allowed, congestion as well as signal strength. The aim is to use this data to collaboratively recommend to a user the access points which they are most likely to prefer.

Our main contributions are as follows. We propose a novel matrix factorisation based recommender approach for use with wireless access point data. This approach uses a mixture model to capture sensitivity of access point ratings to user location and clustering of users into a number of groups to capture user heterogeneity while still allowing accurate predictions when the ratings data is sparse. We evaluate the performance of this approach for a range of network conditions, including a realistic model of a downtown university campus.

II. Related Work

The task of recommending cell selection based on learning of user preferences has received very little attention in the literature to date. Of course there exists an extensive body of

Work supported by SFI grant 11/PI/1177 and 13/RC/2077.

research that highlight the benefits of offloading user traffic to the WiFi or unlicensed LTE networks [5] - [6]. However, the task of network selection is almnost always viewed as a network-wide utility optimisation problem whereby the network provider seeks to maximise data rates and/or minimise delays. This work rarely takes much account of the requirements and preferences of the individual users. Today different WiFi APs and cellular providers offer a variety of services and a user can manually switch between different APs in order to identify its preferred AP. Users are typically only informed of the available access points and their associated signal quality. A notable exception is [7] in which a more sophisticated approach called WiFi-Reports is proposed. This is a collaborative service that provides WiFi clients with historical information about AP performance and application support i.e. beyond just signal strength.

III. Preliminaries

Matrix factorization approaches are a popular and successful way of making recommendations. The ratings matrix $\boldsymbol{R} \in \mathbb{R}^{n\times m}$ is modelled as the product $\boldsymbol{U}^T\boldsymbol{V}$ of matrix $\boldsymbol{U} \in \mathbb{R}^{d\times n}$ and matrix $\boldsymbol{V} \in \mathbb{R}^{d\times m}$. The idea is that the entries in column V_v of matrix $\boldsymbol{V}$ capture the characteristics of resource v as a point in a latent d-dimensional feature space and the entries of column U_u the weights that user u attaches to these, so that the rating of resource u by user u is the inner product $U_u^T V_v$. Importantly, d is much smaller than either n or m. It is this which allows predictions to be made even when the matrix $\boldsymbol{R}$ is sparse.

To estimate $\boldsymbol{U}$ and $\boldsymbol{V}$ a common approach is to adopt the following statistical model. The rating supplied by user u for resource v is a Gaussian random variable $X_{R_{uv}} \in \mathbb{R}$ with mean $\boldsymbol{U}_u^T\boldsymbol{V}_v$ and variance σ^2. That is,

$$Prob(X_{R_{uv}} = R_{uv}|\tilde{\boldsymbol{U}},\boldsymbol{V}) \sim e^{-\phi_{uv}(R_{uv})/\sigma^2}$$

where $\phi_{uv}(R_{uv}) := (R_{uv} - U_u^T V_v)^2$. Let $\mathcal{O} \subset \{1,\cdots,n\} \times \{1,\cdots,m\}$ denote the set of user-resource rating pairs that are contributed by the users and $\mathcal{Z}_{\mathcal{O}} = \{X_{R_{uv}}, (u,v) \in \mathcal{O}\}$, $\mathcal{R}_{\mathcal{O}} = \{R_{uv}, (u,v) \in \mathcal{O}\}$. The conditional distribution over these observed ratings is,

$$Prob(\mathcal{Z}_{\mathcal{O}} = \mathcal{R}_{\mathcal{O}}|\boldsymbol{U},\boldsymbol{V}) \quad \sim \prod_{(u,v)\in\mathcal{O}} e^{-\phi_{uv}(R_{uv})/\sigma^2}$$

Assuming Gaussian priors for $\boldsymbol{U}$ and $\boldsymbol{V}$ with zero mean and variance σ_U^2 and σ_V^2, respectively, the log-posterior is then

$$-\frac{1}{\sigma^2}\sum_{(u,v)\in\mathcal{O}} \phi_{uv}(R_{uv}) - \frac{1}{\sigma_U^2} tr\boldsymbol{U}^T\boldsymbol{U} - \frac{1}{\sigma_V^2} tr\boldsymbol{V}^T\boldsymbol{V} + C \qquad (1)$$

where C is a normalising constant. We now estimate $\boldsymbol{U}$ and $\boldsymbol{V}$ as the matrices which maximise this log-posterior. Observe that since d is small this estimation can be carried out successfully even when the set of observations $\mathcal{O}$ is small (i.e. the observed elements of matrix $\boldsymbol{R}$ are sparse).

It is well known that users can often be grouped together by their preferences. For example, users in the same group may use similar mobile applications, have similar price sensitivity and so on. Following [8] this can be captured by further factorising matrix $\boldsymbol{U}$ as $\tilde{\boldsymbol{U}}\boldsymbol{P}$ where $\tilde{U} \in \mathbb{R}^{d\times p}$ and $\boldsymbol{P} \in \mathbb{R}^{p\times n}$. Column $\tilde{U}_g$ captures the preferences of the g'th group of users (referred to as a *nym* in [8]) and P_u the mapping from user u to these groups. In [8] the elements of $\boldsymbol{P}$ are $\{0,1\}$ valued and $\boldsymbol{P}$ is column stochastic (the columns sum to one) so that each user is a member of a single group. By learning $\boldsymbol{P}$ as well as $\tilde{\boldsymbol{U}}$ and $\boldsymbol{V}$ based on the observed data we can carry out automatic clustering of users into groups in parallel with factorising the ratring matrix. With this change the log-posterior is now

$$-\frac{1}{\sigma^2}\sum_{(u,v)\in\mathcal{O}} \tilde{\phi}_{uv}(R_{uv}) - \frac{1}{\sigma_{\tilde{U}}^2} tr\tilde{\boldsymbol{U}}^T\tilde{\boldsymbol{U}} - \frac{1}{\sigma_V^2} tr\boldsymbol{V}^T\boldsymbol{V} + C \qquad (2)$$

with $\tilde{\phi}_{uv}(R_{uv}) := (R_{uv} - (\boldsymbol{P}\tilde{\boldsymbol{U}})_u^T V_v)^2$. To maximise the log posterior (2), we can apply an iterative algorithm which alternates between the following two steps:

1) Using the current estimates for the matrix of average nym-item ratings $\tilde{\boldsymbol{R}}$ and the number $\boldsymbol{\Lambda}(v)$ of users in each nym who rate item v, estimate $\tilde{\boldsymbol{U}}$, $\boldsymbol{V}$.
2) Given the current estimates for $\tilde{\boldsymbol{U}}$, $\boldsymbol{V}$ each user u updates their column $\boldsymbol{P}_u$ in $\boldsymbol{P}$ by solving $\min_{\boldsymbol{P}_u\in\mathcal{I}} \sum_{v\in\mathcal{V}(u)}(R_{uv} - \boldsymbol{P}_u^T\tilde{\boldsymbol{U}}^T\boldsymbol{V}_v)^2$ where $\mathcal{I} = \{\boldsymbol{e}_i, i = 1,p\}$, $\boldsymbol{e}_i$ the vector for which element i equals 1 and all other elements equal to 0.

IV. Recommending Wireless Access Points

Our problem differs in a number of significant ways from the standard matrix factorisation setup outlined above. Perhaps the most important difference is that we expect the rating assigned by a user to an access point to depend on the user's location. Namely, when close to an access point we expect that the rating may be higher than when further away. We do not, therefore, have a single rating by a user for an access point and cannot construct a single rating matrix $\boldsymbol{R}$ of user-item ratings.

A. Location-based Mixture Model

To accommodate this location dependence we propose the following mixture model approach. We begin by dividing the geographical area A of interest into a set of (possibly overlapping) smaller patches $A_i \subset A$, $i = 1,2,\ldots,q$ such that $\cup_{i=1}^q A_i = A$. These patches might, for example, be selected to be centered on regions where the WiFi access points are most dense or based on local geographical knowledge. The

$$\boldsymbol{R}_{uv} = \sum_{i=1}^{q} \frac{d(u,i)}{\sum_{j=1}^{q} d(u,j)} \boldsymbol{R}_{i,uv} \qquad (3)$$

where $\boldsymbol{R}_i \in \mathbb{R}^{n\times m}$ is a rating matrix associated with patch A_i (ratings made when users are located in patch i), and $d(u,i)$ is the distance between the current position of user u and patch i. We will return to the choice of an appropriate distance metric shortly. Decomposing $\boldsymbol{R}_i$ as $\boldsymbol{U}^T\boldsymbol{V}_i$ and assuming Gaussian priors on $\boldsymbol{U}$, $\boldsymbol{V}_i$ and Gaussian noise on the ratings the log-posterior is

$$-\frac{1}{\sigma^2}\sum_{(u,v)\in\mathcal{O}} \psi_{uv}(R_{uv}) - \frac{1}{\sigma_{\tilde{U}}^2}\sum_{i=1}^{q} tr\boldsymbol{U}^T\boldsymbol{U} - \frac{1}{\sigma_V^2}\sum_{i=1}^{q} tr\boldsymbol{V}_i^T\boldsymbol{V}_i + C \qquad (4)$$

with $\psi_{uv}(R_{uv}) := (R_{uv} - \sum_{i=1}^{q} \frac{d(u,i)}{\sum_{j=1}^{q} d(u,j)} \boldsymbol{U}_u^T (\boldsymbol{V}_i)_v)^2$.

To allow automatic clustering of users into groups (which we refer to as automated group learning) we further factorise $\boldsymbol{U}$ as $\tilde{\boldsymbol{U}}\boldsymbol{P}$ where $\tilde{\boldsymbol{U}} \in \mathbb{R}^{d\times p}$ and $\boldsymbol{P} \in \mathbb{R}^{p\times n}$. With this change the log-posterior becomes

$$-\frac{1}{\sigma^2} \sum_{(u,v)\in\mathcal{O}} \tilde{\psi}_{uv}(R_{uv}) - \frac{1}{\sigma_{\tilde{U}}^2} \sum_{i=1}^{q} tr\tilde{\boldsymbol{U}}^T \tilde{\boldsymbol{U}} - \frac{1}{\sigma_V^2} \sum_{i=1}^{q} tr\boldsymbol{V}_i^T \boldsymbol{V}_i + C \tag{5}$$

with

$$\tilde{\psi}_{uv}(R_{uv}) := (R_{uv} - (\tilde{\boldsymbol{U}}\boldsymbol{P})_u^T \sum_{i=1}^{q} \frac{d(u,i)}{\sum_{j=1}^{q} d(u,j)} (\boldsymbol{V}_i)_v)^2 \tag{6}$$

Note that we use the same matrix $\boldsymbol{P}$ mapping from users to groups for all patches A_i since we assume user preferences do not change significantly with location. User preferences may, of course, vary with their role which in turn may vary with location, e.g. when at home and when at the workplace, but we can capture this within our model by treating a change in role as effectively the introduction of a new user.

B. Learning Algorithm

We assume[1] that the set of patches A_i, $i = 1, \ldots, q$ is given and also the distance metric $d(u,i)$. This allows existing matrix factorisation approaches to be applied in a relatively straightforward manner. Namely, to maximise the log posterior (5), we can apply an iterative algorithm which alternates between the following two steps:

1) Using the current estimates for the matrix of group-item ratings $\tilde{\boldsymbol{R}}_i := \tilde{\boldsymbol{U}}^T \boldsymbol{V}_i$ in patch A_i and the number $\boldsymbol{\Lambda}(v)$ of users in each group who rate item v, estimate each of $\tilde{\boldsymbol{U}}$, $\boldsymbol{V}_i$, $i = 1, \ldots, q$ in turn. That is, we first hold $\boldsymbol{V}_i$, $i = 1, \ldots, q$ fixed and estimate $\tilde{\boldsymbol{U}}$ then hold $\tilde{\boldsymbol{U}}$ fixed and estimate $\boldsymbol{V}_i$ in turn for $i = 1, \ldots, q$. Each of these optimisations is convex and, indeed, is just a least squares problem and so its solution is known in closed-form.
2) Given the current estimates for $\tilde{\boldsymbol{U}}$, $\boldsymbol{V}_i$ each user u updates their column $\boldsymbol{P}_u$ in $\boldsymbol{P}$ by solving

$$\min_{\boldsymbol{P}_u \in \mathcal{I}} \sum_{v\in\mathcal{V}(u)} (R_{uv} - \boldsymbol{P}_u^T \tilde{\boldsymbol{U}}^T \sum_{i=1}^{q} \frac{d(u,i)}{\sum_{j=1}^{q} d(u,j)} (\boldsymbol{V}_i)_v)^2 \tag{7}$$

where[2] $\mathcal{I} = \{\boldsymbol{e}_i, i = 1, p\}$, $\boldsymbol{e}_i$ the vector for which element i equals 1 and all other elements equal to 0. This optimisation is non-convex but can be trivially solved by simply calculating the objective for each element in (small) set $\mathcal{I}$ and selecting the lowest valued.

C. Convergence

We omit the proof for sake of brevity but it can be readily verified that each step of the learning algorithm in Section IV-B is a descent update. Hence, the algorithm is guaranteed to converge to a stationary point of the log-posterior. Since the log-posterior is non-convex (even though it is convex in $\boldsymbol{P}$, $\tilde{\boldsymbol{U}}$, $\boldsymbol{V}_i$, $i = 1, \ldots, q$ individually it is not jointly convex in these matrices) we have no guarantee that the stationary point to which the algorithm converges is not a local minimum or even a saddle point. However, by starting from a number of different initial conditions we can gain some confidence in its convergence to a reasonable point and, as we will see in the next section, experimental studies indicate that convergence is typically both fast and to a reasonable minimum.

V. Performance Evaluation

We evaluate the performance of the proposed access point recommender system with automatic group learning using synthetic datasets, i.e. where we know the ground truth.

A. User Rating Model

For evaluation purposes we model the rating $R_{u,a}$ of access point a by user u located in patch A_i as

$$R_{i,ua} = s_{i,ua} - \mathcal{C}_u \tag{8}$$

where $s_{i,ua}$ is the data rate of user u when using access point a from patch A_i and $\mathcal{C}_u$ denotes the cost of user u (e.g. the cost charged for data access).

Having the user rating depend on the achieved rate $s_{i,ua}$ is natural. For our evaluation we assume that users belong to one of two groups, namely have either $\mathcal{C}_u := 0$ or $\mathcal{C}_u := 25$. This allows us to capture, albeit in a crude way, users with differing price sensitivities. Users are assigned uniformly at random to one of these two groups. This simple model neglects the impact of e.g. blocking of certain applications by access points, but can be readily extended to include such effects.

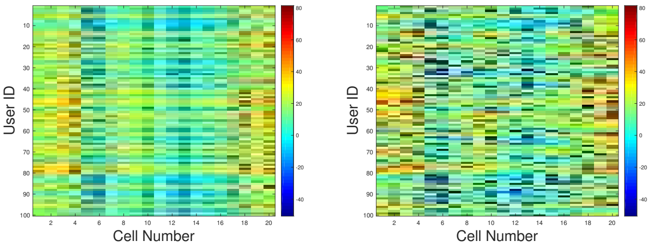
B. Small Geographic Area

We begin by considering a small geographic area corresponding to a single patch A_i within which user ratings are captured by rating matrix $\boldsymbol{R}_i$. We will consider a more accurate path loss model shortly, but initially we let the rate s_{ua} of user u when using access point a be drawn from a Gaussian distribution with mean μ_s and standard deviation σ_s. Fig 2 illustrates the corresponding user ratings (9) for two example realisations. In the left-hand plot the variance σ_s of the rates s_{ua} is relatively small and the presence of two groups of users is evident. In the right-hand plot the variance in the rates is higher with the result that the user ratings are also more variable.

Unless otherwise stated we use the network settings and recommender system parameters summarised in Table I. We measure the performance of the recommender system by

[1] Local geographical information is often available that makes defining the patches A_i relatively straightforward e.g. using knowledge of buildings with many access points where users tend to congregate. Alternatively, clustering approaches such as k-nearest neighbours might be used to induce patches based on measurements but we leave this as future work.

[2] Note that it is straightforward to extend consideration to vectors $\boldsymbol{P}_u$ which are not restricted to be $(0,1)$ valued but this comes at the cost of increased computational complexity and potentially also reduced privacy since when $\boldsymbol{P}_u$ is $(0,1)$ valued step 2 can be efficiently carried out locally within a users mobile handset. Although we do not consider privacy further here due to lack of space, it is increasingly recognised as being an important issue in recommender systems e.g. see [8] and references therein.

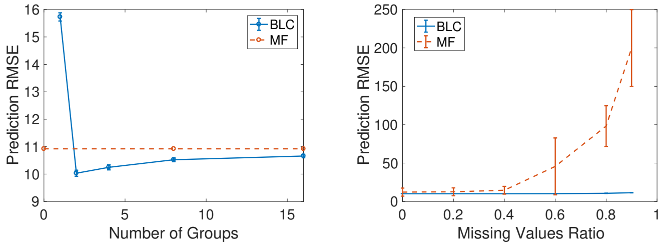


(a) $\sigma_s = 5$ (b) $\sigma_s = 15$

Fig. 2: Illustrating the full user ratings matrix when $\mu_s = 50$ Mbps.

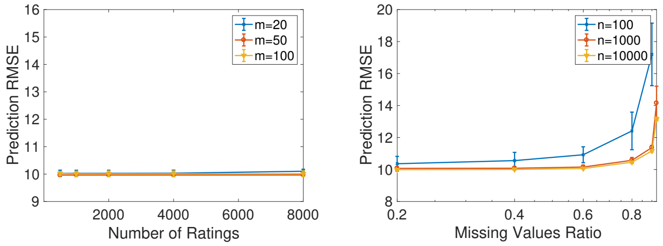
Parameter	Value
Network	
Number of cells (m)	20
Mean users' throughput (μ_s)	50
Standard deviation of users' throughput (σ_s)	10
Learning Algorithm	
Number of groups (p)	2
Dimension of the feature space (d)	2
Number of runs	10

TABLE I: Network and recommender system parameters.



(a) (b)

Fig. 3: RMSE of the recommender system predictions vs. the number of groups p used and the missing values ratio. $m = 20$ access points, $n = 1000$ users. In (b) with automatic group learning $p = 2$ is used. In legend *BLC* denotes automatic group learning and *MF* ordinary matrix factorisation.



(a) (b)

Fig. 4: RMSE of the recommender system predictions with automatic group learning vs. the number of ratings and the impact of the number of users n on the RMSE vs the missing values ratio. Unless otherwise stated $m = 20$, $n = 1000$, $p = 2$.

holding back a subset consisting of 20% of the ratings selected uniformly at radnom without replacement and then calculating the root mean square (RMSE) between the predicted ratings $\sum_{i=1}^{q} \frac{d(u,i)}{\sum_{j=1}^{q} d(u,j)} \boldsymbol{U}^T \boldsymbol{V}_i$ and true user ratings $\boldsymbol{R}$ in this subset. Results shown are the mean and standard deviations over at least 10 random realizations.

1) Prediction error vs. number of groups used and sparsity of ratings: Figure 3(a) illustrates the impact of the number of groups used on the automatic group learning algorithm (i.e. when grouping matrix $\boldsymbol{P}$ is estimated and the dimension of $\boldsymbol{P}$ is varied). It can the seen from Figure 3(a) that as the number of groups used in increased from 1 to 2 there is a sharp drop in the RMSE, as might be expected. As the number of groups used is increased further, the RMSE increases slightly since the number of groups is now larger than the true number of user groups. Also shown for comparison is the RMSE obtained when using ordinary matrix factorisation ($\boldsymbol{P}$ is held equal to the identity matrix $\boldsymbol{I}$, corresponding to each user being assigned to their own group). It can be seem that this RMSE is slightly higher than that with automatic group learning and that as the number of groups in increased the RMSE with automatic group learning rises towards the value for ordinary matrix factorisation.

While Fig 2 shows the full rating matrix in practice each user typically only rates a relatively small number of access points and so the observed ratings matrix is sparse (has many missing entries). We evaluate the impact of the degree of sparsity on recommender performance by removing random subsets of the ratings in each row. The ratio of the removed ratings to the size of the ratings matrix is referred to as the *missing values ratio*. When this is zero then the full rating matrix is observed, when it is close to one then only very few user ratings are observed. In online recommender systems a missing values ratio of 90% or greater is common.

Figure 3(b) plots the measured RMSe vs the missing values ratio for both automated group learning and ordinary matrix factorisation. It can be seen that the RMSE rises sharply for ordinary matrix factorisation as the missing values ratio increases, but increases much more slowly when automatic group learning is used. Indeed when group learning is not used and the missing values ratio is 0.8 the error in the predictions is comparable with the ratings themselves i.e the predictions are largely useless. What is happening here is that learning the group structure allows users in the same group to leverage each others ratings when making predictions, and so greatly increase accuracy especially when the ratings are sparse.

2) Prediction error vs. number of users and access points: Fig 4(a) shows the impact on the prediction RMSE as the number of user ratings available for training the recommender system is varied and as the number of access points m is varied. Note that the dimension of the ratings matrix R changes as m is varied but by holding the number of user ratings available constant we can still directly compare RMSEs as m varies. It can be seen from Fig 4(a) that the RMSE is insensitive to both the number of ratings (so long as this number is not too small, as we know from Figure 3(b); the lowest number of ratings used is 500 in Fig 4(a)) and the number of access points.

Fig 4(b) plot the prediction RMSE vs the missing values ratio and the number of users n.

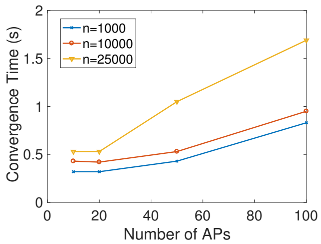


Fig. 5: Convergence time of learning algorithm vs. number of access points and number of users. Missing values ratio is zero.

The RMSE of the recommender system predictions is shown in Fig 4 as the number of users n and the number of access points m is varied. It can be seen that as the missing values ratio is increased the prediction error tends to increase, as might be expected. It can also be seen that as the number of users increases the prediction error tends to decrease, although this effect is relatively small and also diminishes as n increases.

3) Convergence Time: Fig 5 shows the convergence time of the learning algorithm as the size of the network (number of access points, number of users) is varied. Results are shown for commodity hardware (a standard MacBook laptop equipped with an Intel Core i7 2.5GHz CPU having 6 MB L3 cache and an NVIDIA GeForce GT750M 2GB GPU). The group learning matrix factorisation approach lends itself readily to parallelisation and used of the GPU. As a result it can be seen from Fig 5 that it runs fairly quickly even for reasonably large numbers of ratings e.g. with 100 APs and 25000 users there are 2.5M ratings and the computations take about 1.5s in total. It can also be seen from Fig 5 that the the convergence time increases roughly linearly with the number of access points, but sublinearly in the number of users. The latter is to be expected since $\tilde{U}_i$ and $\boldsymbol{V}_i$ scale with the number of groups rather than the number of users.

C. Larger Area

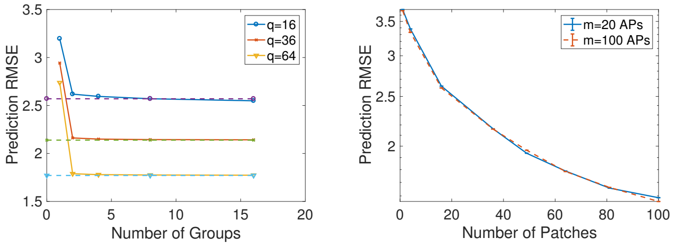
We now extend consideration to a larger area of $300m^2$, where the user rating of an access point is now strongly dependent on their location. The access point locations are selected in turn by drawing a position uniformly at random within the area considered, this position is rejected if it is within 10m of another AP and another position is draw, otherwise the position is retained and the location of the next AP is considered. Users are located uniformly at random within the area of interest. We use the path loss model from 3GPP standard [9]. The rates $s_{u,a}$ are then calculated using the standard Shannon formula for an AWGN channel. The network parameters used are summarised in Table II.

We define patches A_i, $i = 1, \ldots, q$ by dividing the area into a grid of q squares. Each user is mobile and can potentially rate every access point from every patch, although in practice we only observe a subset of these ratings. The rating of access point a by user u located in patch A_i is calculated as

$$R_{i,ua} = \min\{40Mbps, \max\{1Mbp, s_{i,ua}\}\} - \mathcal{C}_u \quad (9)$$

Parameter	**Value**
Network	
Number of cells (m)	20
Carrier Frequency	2 GHz
Channel Bandwidth	20 MHz
Path Loss model	micro urban [9]
AP transmit power	100 mW
UE noise power	-80 dBm
Learning Algorithm	
Number of groups (p)	2
Dimension of the feature space (d)	2
Number of runs	10

TABLE II: Network and learning algorithm parameters.



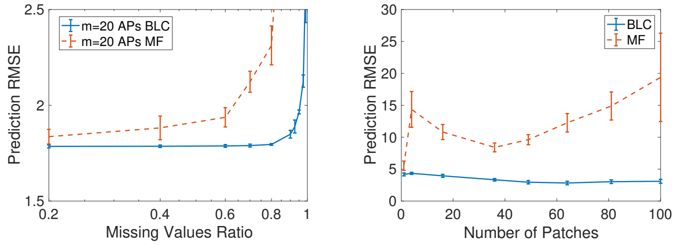
(a) (b)

Fig. 6: RMSE of the prediction error of group learning with $n = 1000$ users vs. number of p of groups used and number q of patches used. The sashed lines in plot (a) indicate results with ordinary matrix factorisation. Unless otherwise stated $m = 20$, $p = 2$, $q = 64$. Full ratings matrix used.

That is, we cap the rate at 40Mbps and ensure that the minimum is at least 1Mbps. In the recommender system we select distance metric $d(u, i) = 1$ when user u is in the patch A_i and $d(u, i) = 0$ otherwise.

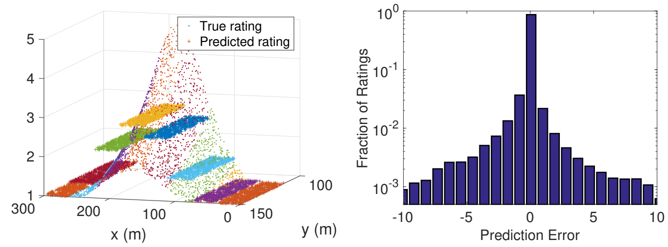
We begin with a sanity check using the full matrix of ratings for both training and testing (i.e. without a 20% hold-back being used for testing). Fig 6(a) plots the measured RMSE of the prediction error vs the number of groups used by the automatic group learning approach. It can be seen that as the number of groups increases from 1 to 2 there is a sharp drop in the RMSE, as expected. Further, that the RMSE is similar with both automatic group learning and ordinary matrix factorisation. This confirms that both approaches are able to model the full measured ratings matrix (i.e. with no missing observations) fairly well and with similar levels of accuracy. Fig 6(b) plots the RMSE with automatic group learning as the number of patches is increased. Here the full ratings matrix is again observed, with each user rating every AP from every patch so that for 100 patches, 20 APs and 1000 users there are $100 \times 20 \times 1000 = 2M$ ratings. It can be seen that the RMSE falls as the number of patches q is increased, as might be expected since more fine grained information becomes available as q increases. Results are shown for both 20 and 100 APs, and are much the same for both.

We now proceed to consider the impact of missing observations and to evaluate the predictive power of the recommender system for data not used for training i.e. its generalisation



(a) $q = 64$ patches (b) $10,000$ ratings

Fig. 7: RMSE of the prediction error vs sparsity of measured ratings available. Data shown both with group learning (indicated as BLC) and ordinary matrix factorisation (indicated as MF). Unless otherwise stated, $n = 1000$ users, $m = 20$ APs, $q = 64$ patches, $p = 2$ groups.



(a) (b)

Fig. 8: Comparison of predicted and actual ratings for a representative AP and users located in a single patch (location indicated on the x and y axes). For $n = 1000$ users, $m = 20$ APs, $q = 64$ patches, $p = 2$ groups and with group learning.

performance. A subset of 20% of the full rating matrix is selected unformly at random without replacement to be held back and used for testing of prediction accuracy. Of the remaining 80% of ratings a subset are selected unformly at random without replacement according to a specified missing values ratio (a missing values ratio of 0.5 corresponding to drawing 50% of these ratings).

Fig 7(a) plots the prediction RMSE as the missing values ratio is varied. Results are shown both with automated group learning and with ordinary matrix factorisation. It can be seen that, similarly to Figure 3(b), when automated group learning is used the RMSE increases much more slowly as the number of missing values increases.

Fig 7(b) shows the prediction RMSE as the number of ratings is held constant and the number of patches is varied. Note that the setup differs from Fig 6(b) not only in that a 20% hold-back is used for testing but also in that the number of ratings is held constant at 10,000 whereas in Fig 6(b) the number of ratings varies with the number of patches used (the full ratings matrix is of size $q \times m \times n$). Hence, in Fig 7(b) the missing values ratio increases as the number of patches increases. This is closer to the situation in reality, and leads to a trade-off whereby as the patch size is made smaller (i.e. the number of patches is increased) the diversity of ratings for an access point due to location variations can be expected to decrease, tending to improve prediction accuracy, but at the same time the number of observed user ratings in each patch will also tend to decrease, tending to degrade prediction accuracy. As a result, we observe in Fig 7(b) that the RMSE exhibits a minimum. For 10,000 ratings when ordinary matrix factorisation is used the minimum is when the number of patches is around $q = 36$ and when automated group learning is used the minimum is at around $q = 64$ patches. The increase in the optimum with automated group learning is due to its better performance when ratings are sparse, see Fig 7(a).

We can gain some more insight into the prediction performance from Figure 8. Figure 8(a) plots the true and predicted ratings for one example AP as those users with $\mathcal{C}_u = 0$ move through a sequence of patches running through the middle of the area considered (the physical locations are indicated by the x and y axes of the plot). It can be seen that the true ratings peak around the centre-back of the plot and decrease smoothly as the distance from the AP increases, in line with the path loss model used. In contrast, within each patch the predicted ratings for users sharing the same group are essentially constant, with this constant value roughly equal to the mean rating of users from that group in that patch. Since the recommender system lacks location information more detailed than the patch in which a user is located, this clearly is a sensible strategy and serves to give some confidence that the recommender system is indeed working in a reasonable fashion.

Figure 8(b) plots the distribution of prediction errors over all ratings (not the RMSE but the error for each individual rating). It can be seen that the errors are concentrated around zero and that the probability of exceeding 2 is less than 1%.

VI. Conclusions

In this paper we consider user to cell association in a wireless network with multiple access points. We propose use of a recommender system to inform the mapping of users to cells. We demonstrate the effectiveness of the proposed grouped-based user to cell associations for a set of synthetically generated user/cell ratings.

References

[1] "Jiwire Hotspot Directory.." http://www.jiwire.com.

[2] "FON Wireless.." https://fon.com.

[3] C. Doctorow, "Why hotel wifi sucks," 2005.

[4] "T-Mobile Germany Blocks iPhone Skype over 3G and WiFi." http://jkontherun.com/2009/04/06/t-mobile-germany-blocks-iphone-skype-over-3g-too, 2010.

[5] S. Singh, H. S. Dhillon, and J. G. Andrews, "Offloading in heterogeneous networks: Modeling, analysis, and design insights," *Wireless Communications, IEEE Transactions on*, vol. 12, no. 5, pp. 2484–2497, 2013.

[6] M. Bennis, M. Simsek, A. Czylwik, W. Saad, S. Valentin, and M. Debbah, "When cellular meets wifi in wireless small cell networks," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 44–50, 2013.

[7] J. Pang, B. Greenstein, M. Kaminsky, D. McCoy, and S. Seshan, "Wifi-reports: Improving wireless network selection with collaboration," *IEEE Transactions on Mobile Computing*, vol. 9, no. 12, pp. 1713–1731, 2010.

[8] A. Checco, G. Bianchi, and D. Leith, "Blc: Private matrix factorization recommenders via automatic group learning," *ACM Trans Security and Privacy*, 2017.

[9] *3GPP TR 36.819 V11.1.0 (2011-12) Technical Specification Group Radio Access Network; 3rd Generation Partnership Project; Coordinated multi-point operation for LTE physical layer aspects (Release 11)*, 2011.