Personal and Ubiquitous Computing
https://doi.org/10.1007/s00779-020-01382-7

ORIGINAL ARTICLE

Persuasion-based recommender system ensambling matrix factorisation and active learning models

Rubén Sánchez-Corcuera[1] · Diego Casado-Mansilla[1] · Cruz E. Borges[1] · Diego López-de-Ipiña[1]

Received: 31 October 2019 / Accepted: 19 February 2020

Abstract

Recommendation systems are gaining popularity on Internet platforms such as Amazon, Netflix, Spotify or Booking. As more users are joining these online consumer and entertainment sectors, the profile-based data for providing accurate just-in-time recommendations is rising thanks to strategies based on collaborative filtering or content-based metrics. However, these systems merely focus on providing the right item for the users without taking into account what would be the best strategy to suggest the movie, the product or the song (i.e. the strategy to increase the success or impact of the recommendation). Taking this research gap into consideration, this paper proposes a profile-based recommendation system that outputs a set of potential persuasive strategies that can be used with users with similar characteristics. The case study presented provides tailored persuasive strategies to make office-based employees enhance the energy efficiency at work (the dataset used on this research is specific of this sector). Throughout the paper, shreds of evidence are reported assessing the validity of the proposed system. Specifically, two approaches are compared: a profile-based recommendation system (RS) vs. the same RS enriched by adding an ensemble with an active learning model. The results shed light on not only providing effective mechanisms to increase the success of the recommendations but also alleviating the cold start problem when newcomers arrive.

Keywords Recommender systems · Persuasive strategies · User profile · Workplace · Cold start · Preference recommendations

1 Introduction

Recommendation systems (RS) are popularly conceived as tools to infer which is the best item of a set for a particular user and which will be the rating he/she will give to it (i.e. satisfaction evidence). Some of the most well-known RS that we use every day are deployed in the context of movies and series (Netflix) [1], shopping (Amazon) [27] or travels (Booking). The majority of these systems focus on finding the best item for a user to watch, to buy or to book. For that, the existing research literature focuses on finding the best approaches (algorithms) to provide just-in-time recommendations with the right item to the user according to their preferences, profiles or previous historical information using the system (i.e. movies the user has already watched, the items bought or the hotels booked and which were all rated positively by her/him or similar profile-based users). Other research lines in the literature of RS are related to solving the cold start problem (CSP) [5, 7, 23]. It refers to the difficulties a model may have if sufficient information about either users or items it envisages to recommend is not provided. CSP is usually given at the first stages of a recommendation system when there have been none or just a few interactions with it.

This paper addresses both RS's issues in the context of persuasive feedback through ICT as a means to enhance energy awareness of office-based employees of tertiary buildings. Thus, on the one side, a methodology is proposed

✉ Rubén Sánchez-Corcuera
ruben.sanchez@deusto.es

Diego Casado-Mansilla
dcasado@deusto.es

Cruz E. Borges
cruz.borges@deusto.es

Diego López-de-Ipiña
dipina@deusto.es

1 DeustoTech - University of Deusto, Avda. Universidades 24, 48007, Bilbao, Spain

	Item 1	Item 2	Item 3
User A	1.44	1.37	2.26
User B	1.32	1.53	1.85
User C	2.76	3.37	3.73

≈

	F1	F2
User A	0.3	0.6
User B	0.3	0.4
User C	0.7	0.8

X

	Item 1	Item 2	Item 3
F1	1.4	3.1	0.3
F2	2.5	1.5	4.4

Fig. 1 Matrix factorisation decomposition from the rating matrix to user and item matrix

to define which is the minimal information the system should gather from users in order to create a valid profile to propose appropriate persuasion-driven recommendations. On the other side, we propose a cascade-based RS composed of two models. The first model uses matrix factorisation mixed with latent metadata from the users to generate a ranking of the best persuasion strategies for a user. The second model, based on active learning methods, predicts the impact that these strategies would have on the user.

Although throughout the article evidence is reported to assess the validity of the approach minimising CSP and producing the right recommendations, we consider that the core contribution of this paper is, from the best of our knowledge, to propose the first persuasion-based recommender system that does not provide to the users the item they are waiting for but elucidates which would be the best persuasive strategy to use when suggesting a new item for them irrespective of the item nature. Previous works have also emphasised the relevance of having systems that elicit profiles to provide right persuasive cues [12, 13, 20, 22]. However, these papers are more focused on knowing which attributes of the users have an impact on their susceptibility to being persuaded, rather than creating an online system itself.

The rest of the paper is organised as follows: Section 2 introduces some concepts that are important to understand the rest of the paper. Section 3 describes the methodology followed to develop the approach presented. Section 4 presents the dataset and the transformations carried out to adapt it for our experiments. In Section 5, we explain our approach and how we build the different models in it. Section 6 is an explanation of the metrics used to validate our model and the next section a description of the experiments. In Section 8, we analyse the results obtained in the experiments. Finally, in the last section, we conclude our work and propose possible improvements in the future

2 Background

In this section, we will provide an introduction about matrix factorisation, principal component analysis (PCA) and active learning. These concepts are explained to provide the reader with a technical background to better understand some concepts that appear in the manuscript.

2.1 Matrix factorisation

Matrix factorisation [15] is a collaborative filtering algorithm used in RS. This algorithm works by decomposing the user-item interaction into a matrix where the axis are the items and the users in the RS as shown in Fig. 1. Each of the cells of the matrix has the rating that a respectively user has given to a specific item. After that, the idea of MF is to represent the interactions between user and item in a lower

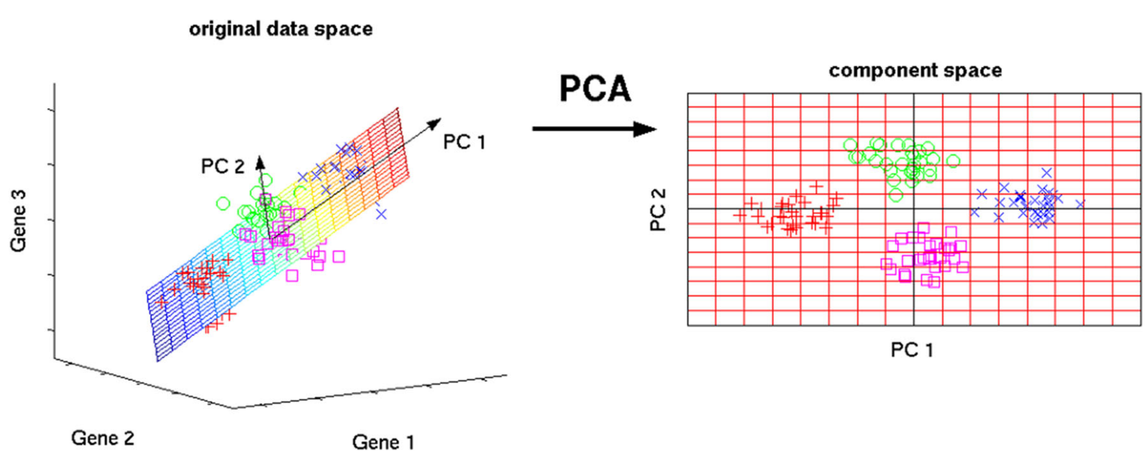

Fig. 2 Principal component analysis conducted to represent points from three-dimensional space to two-dimensional space. Source [19]

dimensional latent space and be able to relate users and items depending on the distance in that dimensional space.

2.2 Principal component analysis

Principal component analysis [29] is a statistical procedure that uses a transformation to convert high-dimensional set of variables into ones with less dimensions while keeping the information. For example, if we have a dataset with n items represented with p variables, applying PCA we can find the minimum number of variables that represent the items, those variables are called principal components. Sometimes, PCA is also used to reduce dimensionality and to be able to represent the clusters in a 2-D or 3-D space for visualisation purposes, for example, as seen in Fig. 2.

2.3 Active learning

Active learning (AL) is a *semi supervised* machine learning algorithm and a subfield of artificial intelligence. Unlike the traditional "passive" learning systems that try to resolve a hypothesis using all the training data available, AL develops and tests new hypotheses as part of iterative and interactive learning process.

AL systems consist of four different elements: (i) the oracle, (ii) a pool of instances, (iii) a query strategy, (iv) and a learner as shown in Fig. 3. The pool of instances refers to the data that needs to be labelled. Usually, in AL systems, this pool is assumed to have a large number of instances with some of them previously labelled. The oracle is the entity that offers the labels for the instances extracted by the query strategy. In online experiments, the oracles are the users when they offer feedback to the system. The learner is the machine learning algorithm used by the system to evaluate the instances and learn their representations.

The query strategies give the difference between active learning and passive learning. While in passive learning, all the instances are sent to the learner sequentially; in active learning, the system, using query strategies, selects the next point to label based on a specific purpose. Query strategies represent the algorithm that will be used to decide which of the instances should be labelled next. There are a significant number of strategies usually organised in different categories based on their purpose, for example, expected error reduction or variance reduction. In essence, using active learning can significantly reduce the amount of labelled data that is needed and hence the experts or the number of interactions with them required to accurately label data.

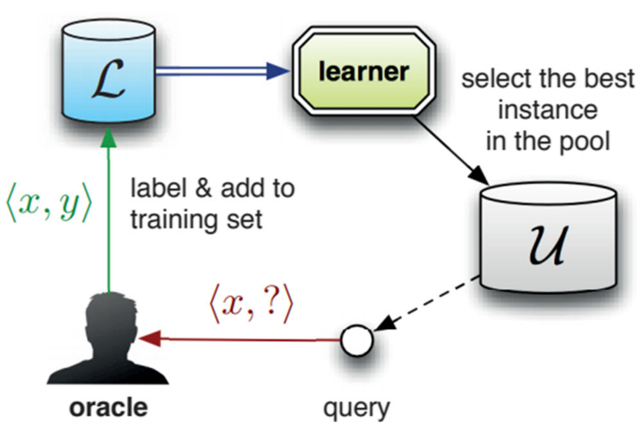

Fig. 3 Pool-based sampling in an active learning system

3 Methodology

For developing the system described in this manuscript, we first gathered the data used for the experimentation. This dataset was generated by and European project that will be explained in Section 4. Then, we preprocess that data in order to select the minimum number of features that represent the user reducing the dimensionality of the feature vector. After this, we analysed the proposed system requirements for our project and selected the architecture and models that fit with it. Our next step was to feed the models with data, record the results and analyse them to validate the whole system.

4 Data pre-processing

The dataset used to pursue the experiments was generated through a questionnaire conducted by the H2020 European project GreenSoul [3]. The aim of the survey was to gather data from the participants of the pilots from the UK, Spain, Greece and Austria so the researchers could have some previous information about the profiles of the participants in relation to energy efficiency in tertiary buildings. The researchers created a dataset by making people with different profiles rate a set of persuasive strategies from 0 to 5. With the data obtained, the scientists conduct several experiments in relation to persuasion (i.e. match which user profiles were more akin to act in favour of the environment by being provided to certain persuasive strategies [3]). Specifically, the whole dataset consists of 303 entries (users) with 37 features per each user (e.g. socio-economic, demographics, cultural or attitudinal attributes) and their ratings for 21 persuasive strategies proposed.

In order to be able to build a RS to provide the most effective persuasive strategies per type of user in relation to saving electricity in shared spaces using the proposed dataset, we set some requirements in relation to the input data.

Table 1 Importance per feature and ranking in extra trees classifier and random forest classifier

Feature name	Extra trees	Random forest
Frequency	0.12 (1)	0.12 (1)
Education	0.11 (2)	0.11 (2)
PST	0.11 (3)	0.11 (3)
Work culture	0.09 (4)	0.09 (4)
Barriers	0.09 (5)	0.08 (6)
Age	0.07 (6)	0.07 (7)
City	0.07 (7)	0.09 (5)
Intentions	0.07 (8)	0.06 (8)
Genre	0.06 (9)	0.06 (9)
Confidence	0.05 (10)	0.05 (12)
Country	0.05 (11)	0.05 (11)
Influenceness	0.05 (12)	0.05 (10)
Susceptibility	0.03 (13)	0.03 (13)
Organisation energy	0.03 (14)	0.03 (14)

We conducted a feature selection using extra trees and random forest as we explain in [25]. As can be seen in Table 1, both of the trees gave different importance ratings to the features in the dataset, but the order of the features was rather similar. Hence, according to [26], we decided to retain the features that provide the 80% of the information regarding the users in the dataset. In our case, starting from the feature which provides more information in relation to the persuasive strategies (Frequency), we choose the following features that in total represent the 80% of the relevance in both of the classifiers, represented in italics in the Table 1. After this transformation, applying the random forest approach, the dataset was ready to be used in our system.

Some features shown in Table 1 are typical socio-economic and geographical information, such as age, genre, education and city. Attitudinal profile (PST) and intentions are two variables more related to norms and beliefs of each user in relation to pro-environmental actions. Barriers to behave more sustainably at work and work culture are two features specific to the workplace. Finally, frequency measures the willingness of the users to receive less or more persuasive cues.

Apart from the attributes collected in the dataset, users were asked to rate 21 persuasive strategies related to enhancing energy efficiency in shared spaces at work. Users had to rate the strategies from 1 to 5 depending on the effect those strategies would have on them in order to reduce the energy they use in their workplaces (they also could provide a 0 rating if the strategy was not applicable to them). The strategies were designed based on 15 different persuasion principles developed by experts in the field [6, 9, 21]. In Table 2, we can see how the 21 persuasive strategies are mapped to these principles of persuasion and an example for each strategy.

Table 2 Mapping of persuasive strategies to unitary principles of persuasion and examples for each persuasive strategy

Persuasive principles	Persuasive strategies	Persuasive strategy example
Authority	v10,v21	Make the people think that the system proposed is an expert on energy efficiency.
Cause and effect	v15	Provide a means to visualise the outcomes if the desired action was achieved.
Conditioning	v6	(Positive reinforcement) Provide incentives for certain actions.
Cooperation and liking	v4	Make the user think that the computer/system is a teammate towards achieving a green goal.
Tailoring and personalisation	v9,v14	Send massages to the user using his/her name or something that is related to him/her.
Physical attractiveness	v5	Create digital or physical interfaces with aesthetics in mind.
Praise	v3	Use praise as a way to provide user feedback information based on his/her behaviours.
Verifiability and real-world feel	v8,v13,v22	Provide information about the measurement equipment.
Reciprocity	v7	Give hints about the efficiency gained by the interactive/smart system.
Reduction	v12	Ease the action by providing steps of completion.
Self-monitoring	v11,v16,v18	Provide tools where the users can see their consumption
Similarity	v20	Find peers that can give advices to target users through social networks or platform.
Social proof	v20, v10	Show the number of followers of the system.
Social recognition	v2,v7	Showing in social networks or in public that someone is the best of the month.
Suggestion	v19	Provide hints/cues just-in time or about-to moments.

5 Proposed approach

To develop our approach, we decided to use the cascade method for recommender systems presented in [2]. Systems build with this method are composed by several RSs that filter the items until they reach the final recommendation. With this methodology, RSs avoid spending resources with items that will not be relevant for the user as they are already filtered by the first models.

As we used the cascade method, we implemented two different models for the RS: the Hybrid item Ranker (HiR) and the Specific impact Predictor (SiP). HiR is based in preference-based filtering methods so its output is a ranking of the items for every user in the system. This model is also in charge of filtering the items, so only the best persuasion strategies for a user reach the second model. The input of the first model is composed of a rating matrix and features from users. As output, it generates a prediction of how the users will rank the items in the dataset. The SiP is an AL model fed with the data from the output of the first one. As output, this model predicts the rating that the users will give to every item. Figure 4 represents the architecture of the RS with the inputs and outputs of both models.

5.1 Hybrid item Ranker

To develop the HiR model, we decided to use the hybrid method for RSs that combines the content-based and collaborative methods. With this method, it is possible to use the data of other users' rates to recommend similar items and data of items to make content analysis so we can mitigate the problems that collaborative method have. Hybrid methods can be implemented with different combinations of techniques that adapt to different problems as explained by Çano and Morisio [2].

To develop this model, we reviewed several techniques that allow us to introduce features from users or items in the calculation for the recommendations. For this purpose, we decided to use a model that includes matrix factorisation (MF) technique [15] for its popularity in RS and their flexibility for modelling real-life situations.

The proposed model is implemented using the LightFM library developed by Kula [16] that allows us to create a hybrid latent representation model. This library has been used in many projects to create recommendation systems before demonstrating its effectiveness. For example, Rubtsov et al. [24] created a music track recommendation system that achieved third place in the RecSys Challenge 2018 using this library.

To describe the model formally as stated by [16], let U be the set of users, I the set of items, F^U the set of user features and F^I the set of item features. Every user interacts with several items positively or negatively. The set of all interaction pairs $(u, i) \in U \times I$ is the union of both positive S^+ and negative interactions S^-.

Both users and items are fully described by their features. Each user u is described by a set of features $f_u \subset F^U$. The same states for each item i described by its features given by $f_i \subset F^I$. Features from users and items are known in advance, for example, the age of a user or the genre of a film.

The parameters added to the model are set in terms of d-dimensional user and item feature embeddings $\mathbf{e}_f^U$ and $\mathbf{e}_f^I$ for each feature f. Every feature is also described by a scalar bias term (b_f^U for user and b_f^I). The latent representation of user u and item i is given by the sum of its feature's latent vectors:

$$\mathbf{q}_u = \sum_{j \in f_u} \mathbf{e}_j^U \quad \text{and} \quad \mathbf{p}_i = \sum_{j \in f_i} \mathbf{e}_j^I$$

The sum of feature biases gives the bias term for user u and item i:

$$b_u = \sum_{j \in f_u} b_j^U \quad \text{and} \quad b_i = \sum_{j \in f_i} b_j^I$$

The predictions that the model generates for user u and item i are given by a function that computes the dot product of user and item representation, adjusted by user and item feature biases:

$$\hat{r}_{ui} = \boldsymbol{f}(\mathbf{q}_u \cdot \mathbf{p}_i + b_u + b_i)$$

The optimisation objective of the model consists of maximising the likelihood of the data conditional on the parameters. The likelihood is given by

$$L\left(\mathbf{e}^U, \mathbf{e}^I, b^U, b^I\right) = \prod_{(u,i) \in S^+} \hat{r}_{ui} \times \prod_{(u,i) \in S^-} (1 - \hat{r}_{ui})$$

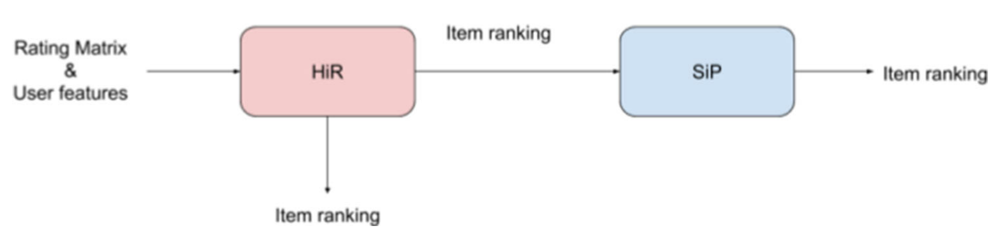

Fig. 4 Architecture of the recommender system divided in its two models

Through this library, we can create a model that enhances the MF method by adding metadata to either user or item vectors. Users and items are represented by a vector that contains the desired metadata; this structure is called latent vector or embedding. A scalar bias also represents each feature of users; the addition of all these biases creates the total bias for a user, the same states for the items. To infer the ratings, a user will give to a specific item we use the bias and the latent vectors for that user and item pair.

Furthermore, the generated embeddings can be represented in an high-dimensional space so the items represented near in the space have similar attributes and the ones far have different attributes. Users can be as well represented so users that are near in the space should have similar features and thus they may have similar preferences for items.

In Fig. 5, we can see a principal component analysis (PCA) conducted to the embeddings generated from the users in our dataset. As we can see in the figure, there are three different clusters. In the right part of the figure, there is a group with very similar features so the points are piled up; however, in the groups at the left, the points are more scattered so the user attributes may be more diverse.

As we have stated before, we want to generate rankings of items (persuasive strategies) for each user in the dataset. To do so, we make the model learn the embeddings generated from both users and items. With this information, the system will be able to encode the preferences that users have over items. When the embeddings are multiplied together, they produce scores for the set of items and a specific user, the highest scores represent the most effective persuasive strategies for a user.

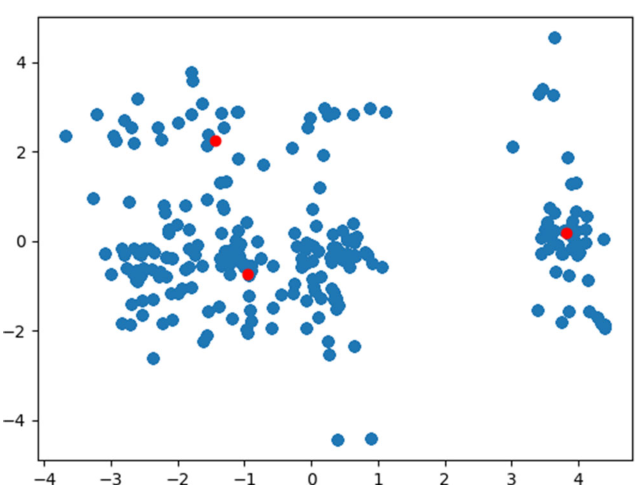

Fig. 5 Representation of PCA conducted on the embeddings generated from user profiles.

5.2 Specific impact Predictor

The objective of the SiP model is predicting the ratings that users will give to items in the dataset. For this task, identified as a regression problem, we created an AL model capable of solving it. The proposed model is implemented using the modAL library, which is designed to create AL models in python and has the query strategies used in this paper already implemented.

As explained in Section 2, AL models have, among other things, a learner and one or several query strategies. For this specific problem, we decided to use a Gaussian process regressor (GPR) as the learner. Gaussian processes have become vital when modelling non-linear pattern in real-world configuration, for example, human preferences [10].

The aim of GPR is to create a function $f(x)$ that an input vector x and the noise ε, that is observed to follow another Gaussian distribution, would have as an output the target value y. The function is composed by the input vector x and a vector of weights w as we can see in this equation:

$$f(x) = x^T w, \qquad y = f(x) + \varepsilon \tag{1}$$

As we explained in Section 2, AL models use query strategies to decide which will be the next data point they will label. In our approach, we used a custom uncertainty query strategy that selects the data points based on which of the predictions of the estimator has the highest deviation. The point chosen will be the one given by the output of this equation:

$$Q(X) = argmax(\hat{r}) \quad \hat{r} = (x_i - \bar{x})^2$$

in which $\hat{r}$ calculates the standard deviation for every predicted point and $Q(X)$ selects the point with the maximum deviation from the predictions done by the estimator as the next query.

6 Evaluation

For the evaluation of the presented model, we used the dataset described above [3]. Due to the architecture of our system, we conducted different evaluations for the HiR model and the SiP model. Reviewing the state of the art, we could not find any work trying to achieve similar objectives, so we proposed a benchmark with three different baselines to evaluate the performance of the HiR model and a benchmark with two passive learning methods and different query strategies for the SiP model.

To determine if the HiR is effective, we have used two different metrics that represent how relevant is the recommendation done to a user and how the ranks created by the model are similar to the ones created by the user. For the first evaluation, we have used a variation of the popular metric F-score, which is a normalisation metric of *precision* and *recall*. Instead of using the default version of the metric, we used the ones proposed by Herlocker et al. [11]. In this version of the metric, there are relevant and irrelevant items for the user instead of the normal true-positive (TP) and false-positive (FP) ones. Relevant items are the items that the user has rated higher than X and irrelevant the items with less rating than X (X may be any number between the possible rankings in the system. In this case, X is 3). We decided to evaluate the precision and recall at 5 in order to get the five best strategies for each user.

As we explained before, the HiR model generates a rank of the strategies for each user in which the most effective strategy is the first of the rank and the last the least effective. We used the normalised distance-based performance measure (NDPM) explained in [31] to evaluate the precision of our approach.

This metric measures the distance between two rankings or orders, usually the one given by the user and the one proposed by the system. The distance is calculated by comparing the position of each item in both of the rankings. This metric follows three rules to determine if the orders are *agree*, *disagree* or are *compatible* regarding two items:

- The orders $\succ_1$ and $\succ_2$ *agree* on items i_1 and i_2 if $i_1 \succ_1 i_2$ and $i_1 \succ_2 i_2$.
- The orders $\succ_1$ and $\succ_2$ *disagree* on items i_1 and i_2 if $i_1 \succ_1 i_2$ and $i_2 \succ_2 i_1$.
- The orders $\succ_1$ and $\succ_2$ are *compatible* on items i_1 and i_2 if $i_1 \succ_1 i_2$ and neither $i_1 \succ_2 i_2$ or $i_2 \succ_2 i_1$. That is, when the items are tied or incomparable because one of them is not in the other order.

Following these rules, if the orders agree, the distance between them will not increase. Instead, if the orders disagree or are compatible, the distance will increase. As this metric calculates distances, the lower the distance, the better will be the prediction.

To measure the effectiveness of the SiP model, we used a predictive accuracy metric. These metrics measure how close the ratings predicted by the model are to the actual user ratings and they are usually used to evaluate non-binary variables (e.g., 1 to 5 stars ratings). Although different papers are positioned towards one or another metric, in this work, we opt to use mean absolute error (MAE) because the penalisation for the errors in the system follows a linear function as explained in these works [4, 28].

The system generates predicted ratings u_{ui} for a test set $\mathcal{T}$ of user-item pairs (u, i) for which true ratings r_{ui} are known.

Table 3 F-score and NDPM for 5 elements ranking evaluation

Baseline/model name	F-score	NDPM
Random	0.32354 ± 0.005	0.49853 ± 0.003
Top 5	0.43919 ± 0.000	0.44433 ± 0.000
HiR without user data	0.34729 ± 0.007	0.11652 ± 0.000
HiR	**0.46083 ± 0.023**	**0.08635 ± 0.02**

Entries in bold represent the lowest (in these tables, the best score) score in each row of the table

MAE measures the error, so the lower the MAE is, the more accurately the recommendation engine predicts user ratings. The MAE between predicted ratings and true ones are given by

$$\mathrm{MAE} = \frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} |\hat{r}_{ui} - r_{ui}|$$

7 Results

As stated before, reviewing the state of the art, we could not find any system working on the same field as ours. For this, we opt to create baselines for both of the models that compose the whole system. For the HiR, we created three baselines to compare. The first one is defined as ‘random’ in which the proposed ranking is randomly selected among the possible twenty-one strategies. The second one is ‘Top 5’, which always proposes the average top 5 strategies for all the users in the dataset. The last benchmark is the same as the proposed model; however, no user data if feed to the model from the user profiles.

To validate the performance of the HiR, we conducted experiments in which the 80% of the dataset represented the training set and the 20% left the test set. To asses that the results achieved by the different baselines in the experiments, we followed the 10-fold cross-validation method. The results of these experiments are shown in Table 3.

For the validation of the SiP model, we have conducted several experiments combining different query strategies with seed sizes. The seed size represents the percentage of the whole dataset that is used as training set for the model. To compare the performance of our model, we compared it with two well-known algorithms for regression problems: Bayesian regressor (BR) and support vector machine (SVM). To assess the results achieved by the different experiments, we used the 10-fold cross-validation method and conduct 10 queries to the instance pool in each of them.

In Tables 4, 5, and 6, the results obtained in the experiments are shown. The first experiment, which results

Table 4 Summary of the MAE and time obtained with all the data

	Active learning						BR		SVM	
	Density		Uncertainty		Random					
	Time	Density	Time	MAE	Time	MAE	Time	MAE	Time	MAE
1%	37.205	0.735	0.059	**0.72**	0.052	0.746	0.002	0.736	0.005	0.746
5%	32.947	0.73	0.048	**0.72**	0.047	0.733	0.002	0.731	0.026	0.748
10%	29.191	0.739	0.045	0.736	0.051	0.74	0.002	**0.733**	0.066	0.745
15%	26.264	0.745	0.048	0.743	0.045	0.746	0.002	**0.737**	0.125	0.752

Entries in bold represent the lowest (in these tables, the best score) score in each row of the table

are represented in Table 4, was conducted using all the dataset as instance pool for the AL model. The second and the third experiments, which results are represented in Tables 5 and 6 respectively, were conducted using the HiR to first filter the best ten and five strategies for all the users.

8 Discussion

Tables from 3 to 6 present the results obtained from the experiments conducted for the validation of both models. In this discussion, we will analyse the performance of both models separately as the output of each of them offers us relevant information to make the recommendations.

Comparing the scores obtained by the HiR model in Table 3, we can see how our model outperforms the benchmarks in both of the metrics that we used for the validation. Addressing the F-score, the HiR model performs better than the benchmarks; however, the Top 5 is only slightly worse than the proposed model. We have analysed this outcome checking the ratings in the dataset realising that there are five strategies in which most people agree that they are useful. In this case, the precision of this baseline will be high due to the large percentage of people rating these strategies with four or five. Furthermore, we also realised that there are people that rated these strategies with low scores. This phenomenon contributes to the fact that the recall of this baseline can rise the F-score making the metric to be closer to the proposed model.

NDPM calculates the distance between two rankings. Hence, the smaller the distance is, the better the model. As we can see in Table 3, our model outperforms all the benchmarks by far except the HiR without user data. In this case, we can appreciate that adding the information about the users allows us to improve the ranking skills of our model and reducing the CSP which is one of the primary outcomes of this work.

Regarding the SiP model, we recorded the MAE obtained by each system and the time spent doing ten queries to the instance pool. The experiments were designed with small seeds as we realised that the dataset was learned quickly.

We can observe when comparing the Tables 4, 5 and 6 that the minimums are given in the Top10 experiment. We may conclude that this is because the previous filter that the HiR model does to the dataset. Another observation done is that SiP with fewer data as the seed is capable of reducing the MAE more with the same number of queries than the conventional algorithms. Query strategies implemented in the system allow it to gather the best points to reduce its MAE.

Table 5 Summary of the MAE and time obtained with Top10 data

	Active learning						BR		SVM	
	Density		Uncertainty		Random					
	Time	Density	Time	MAE	Time	MAE	Time	MAE	Time	MAE
1%	8.273	**0.695**	0.037	0.695	0.036	0.711	0.001	0.721	0.002	0.718
5%	7.66	**0.698**	0.035	0.71	0.035	0.705	0.001	0.712	0.003	0.701
10%	6.97	0.685	0.034	0.686	0.034	**0.685**	0.001	0.702	0.003	0.706
15%	6.33	0.699	0.033	**0.697**	0.032	0.698	0.001	0.711	0.004	0.71

Entries in bold represent the lowest (in these tables, the best score) score in each row of the table

Table 6 Summary of the MAE and time obtained with Top5 data

	Active learning						Bayesian regressor		SVM	
	Density		Uncertainty		Random					
	Time	Density	Time	MAE	Time	MAE	Time	MAE	Time	MAE
1%	2.439	0.784	0.03	0.759	0.029	0.788	0.001	0.754	0.001	**0.736**
5%	2.27	**0.702**	0.033	0.706	0.027	0.711	0.001	0.738	0.001	0.716
10%	2.119	**0.713**	0.032	0.723	0.028	0.719	0.001	0.727	0.003	0.729
15%	1.932	0.706	0.037	**0.698**	0.031	0.701	0.001	0.712	0.006	0.709

Entries in bold represent the lowest (in these tables, the best score) score in each row of the table

9 Related work

Even though there has not been relevant works concerning the recommendation of persuasive strategies to enhance energy saving, the research in RSs is very large. In this section, we will introduce works done in some of the techniques presented in this manuscript.

9.1 Preference-based recommendation systems

There are several RS which aim to predict the preferences of a user without taking into account the impact of those preference in the user. These systems aim to create a ranking of the existing items taking into account which are the user preferences and offering them the whole ranking to choose. There are several works done in this field. In [17], the authors present two novel methods to improve state-of-the-art ranking-based VSRank algorithm. The first contribution is to add the negative similarity between users to calculate the relative preference for items. Through this method, the authors are capable of recommending items based on preferences from a user with strongly opposite interest. The second contribution is to add information from social networks to infer the relationship between two users. Authors state that depending on the relationships users are more likely to rate the same items with similar ratings. They finally expose several benchmarks in which their proposal outperforms the base algorithm and other methods usually employed in RS.

Regarding applications that use this type of recommendations, Wu et al. [30] designed and implemented a recommendation system based on user preferences to recommend business partners (buyers and suppliers). In the proposed approach, the recommendation system processes the user preferences and items in the dataset as trees. A tree is used for every user in which each leaf in the tree represent an item in the dataset and the preference level for that specific user is also represented in each leaf. A tree-structured data matching algorithm constructs a map to identify which parts of both trees correspond the most and therefore generate a ranking of the best items for a user. With this algorithm, they propose a system called SMART BIZSEEKER which is able to propose a ranking of partners for a business user.

Liu et al. [18] propose a hybrid recommendation system for books based in learning-to-rank model. In the manuscript, the authors propose an enhanced collaborative ranking method that employs two different types of latent features, derived features and content features. Derived features are extracted from the rating information of either users and books. However, content features are collected from the explicit information available in users profiles or books. Combining these two features, the authors are capable of creating a new vector that a regression or classification model will use to train the ranking model. Finally, they present the evaluation of their model with successful results.

9.2 Active learning

Active learning in RSs has been used to reduce the CSP in systems that needed an initial test to gather information from users before starting to recommend items to them. Many authors have used this method to enhance their RSs by creating efficient query strategies that adapt to their problems.

The paper presented by Pozo et al. [23] focuses on solving new users' cold start problem by using an initial questionnaire to acquire past users' interest and generate better recommendations. To make initial questions more efficient, the authors proposed to use AL methods. The main idea is to train an accurate collaborative filtering model to generate the best recommendations. To accomplish this objective, users should answer a questionnaire when they use the system the first time. AL methods are used to select the best ten questions to make to each user. In this questionnaire, the questions are not predefined; instead, each question is selected depending on the answer given to the last one. To test their approach, the authors used the well-known Movielens (1M and 100k) datasets. Their techniques improve the state of the art in terms of root-mean-square error (RMSE) and the number of questions needed to achieve the desired RMSE. However, they state that their models are more efficient in small dataset context, where low interactions do not allow to create big decision trees.

Khenissi et al. [14] develop two new query strategies to increase explainability of the algorithm and the accuracy of matrix factorisation method. In the poster, the authors create two strategies called ExAL-min and ExAL-max, which comes from “explainable active learning”. They present explainability as the value that represents how explainable the next recommendation is. The ExAL-min strategy picks the item that is expected to have the least estimated mean absolute error with taking into consideration explainability. Instead, the ExAL-max strategy learns the real rating of the items that are expected to increase the test error and the newly selected items will provide new information to help improve the predictions done by the system. In the experimentation, the authors used the MoviLens dataset as well. Authors used precision, recall and *F*-score to measure the explainability and MAE to measure the accuracy loss. The conclusions for the paper are that ExAL-min is the optimal method for increasing explainability and ExAL-max helps to increase the accuracy at a faster pace. Finally, they propose a combination of both of them in which the trade-off between accuracy and explainability can be controlled.

9.3 Cold start problem

CSP can be reduced by different methods but we will focus on the introduction of features from both users and items in the system as is the one we have employed in our approach.

In [7], the authors present an advanced algorithm for collaborative filtering recommendation system that learns the mapping between the item attribute space and a high dimensional space commonly used in these approaches. This algorithm is called LearnAROMA as is a variant of the AROMA algorithm proposed in [8]. The proposed algorithm learns how to represent the item attributes in high-dimensional spaces and proposes them for new items so the system is able to recommend them without having to wait for the rate of a user. As these representations may not be very accurate, the authors have developed a dynamic algorithm that adapts the vectors as the user ratings are introduced in the system. Finally, they state that the algorithm presents an improvement in the state of the art and the algorithm that they propose to update the model dynamically achieves the same accuracy as a model fully trained with all the ratings.

Chamoso et al. [5] proposed a context-aware recommendation system that aims to recommend user-user and user-job relationships. The proposed algorithm relates the users with job offers by calculating their affinity value assigning weights to a set of attributes extracted from the user profile and the job information. Users are related among them using the same method with different attributes. These weights are updated after every interaction with the system. In order to reduce the cold start problem, the system relates new users to ones with similar profiles. With this method, the authors state that the acceptance rate of jobs has been increased and the refuse of them decreased.

10 Conclusions and future work

In this paper, we have proposed a novel recommendation system that selects appropriate persuasive strategies for increasing environmental awareness towards reducing energy consumption in shared spaces such as the workplace alleviating the cold start problem. By conducting a review of RS literature and the related state of the art, we have identified that we should use systems that combine information about user and item feature to reduce the cold start effect.

To measure the impact that the persuasive strategies have on each user, we implemented an AL system. Instead of following the conventional approach of tuning the query strategies to make the AL model better, we add it a previous filter. This filter is implemented by a ranking model designed to be the first input of the whole model. As explained in the manuscript, AL techniques rely on quality data to generate better recommendations, so this filter improves the query strategy by previously filtering the bad strategies for each user.

Whereas we have conducted a thorough analysis and evaluation of the solution presented providing conclusive evidence of its effectiveness, we deem that the core contribution of this paper is to provide, from the best of our knowledge, the first persuasion-driven RC. The system is currently dependent and tightly coupled to the context of use (i.e. environmental awareness and energy efficiency at the workplace). However, with the right data, it could be extrapolated to other contexts. As has been explained in the paper, the aim of our approach is not to provide the user with the item he or she is waiting for, but elucidating which would be the best persuasive strategy to use when suggesting new items for users. For example, if we want a person to buy a pair of headphones, instead of offering the headphones directly, we could use a persuasive strategy to make him/her realise how a pair of headphones could improve his/her lifestyle. Hence, based on the experience acquired, we argue that interested researchers could create a similar RS which may work with other existing RSs such as those of Amazon, Netflix or Booking. In this latter case, ethical considerations may appear. This is not the case of our application context, where the objective is providing the right hints for people to behave in a pro-environmental way.

Regarding future work, we identified that changing the size and domain of the dataset and the way we conduct the experiments could allow us to create a comparison benchmark in which we could see how our model behaves with different data sources. Furthermore, we also realised that conducting an online test in a closed environment would allow us to identify how real users react to the different persuasive strategies that our system recommends and if our system is achieving the same results as in offline tests.

Funding information This research was funded by the European Commission through the project HORIZON 2020 - RESEARCH and INNOVATION ACTIONS (RIA)-696129-GREENSOUL. We gratefully acknowledge the support of the Ministry of Economy, Industry and Competitiveness of Spain under Grant No.: TIN2017-90042-R.

References

1. Bennett J, Lanning S et al (2007) The netflix prize. In: Proceedings of KDD cup and workshop, vol 2007. New York, p 35
2. Ċano E, Morisio M (2017) Hybrid recommender systems: a systematic literature review. Intell Data Anal 21(6):1487–1524
3. Casado-Mansilla D, Borges CE, Kamara O, Sanchez-Corcuera R, Manterola A, López de Ipiña D, Tsolakis A, Papageorgiou D, Krinidis S, Moschos I (2019) Socio-economic and cultural dataset in relation to persuasive strategies to boost energy efficiency and in the uk, spain, greece and austria. https://doi.org/10.5281/zenodo.2596067
4. Chai T, Draxler RR (2014) Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. Geoscie Model Dev 7(3):1247–1250
5. Chamoso P, Rivas A, Rodríguez S, Bajo J (2018) Relationship recommender system in a business and employment-oriented social network. Inf Sci 433:204–220
6. Cialdini RB, Cialdini RB (2007) Influence: the psychology of persuasion. Collins New York
7. Cohen D, Aharon M, Koren Y, Somekh O, Nissim R (2017) Expediting exploration by attribute-to-feature mapping for cold-start recommendations. In: Proceedings of the 11th ACM conference on recommender systems, pp 184–192. ACM
8. Crammer K, Kulesza A, Dredze M (2009) Adaptive regularization of weight vectors. In: Advances in neural information processing systems, pp 414–422
9. Fogg BJ, Soohoo C, Danielson DR, Marable L, Stanford J, Tauber ER (2003) How do users evaluate the credibility of web sites?: a study with over 2,500 participants. In: Proceedings of the 2003 conference on designing for user experiences, pp 1–15. ACM
10. Guo S, Sanner S, Bonilla EV (2010) Gaussian process preference elicitation. In: Advances in neural information processing systems, pp 262–270
11. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inform Syst (TOIS) 22(1):5–53
12. Kaptein M, Markopoulos P, De Ruyter B, Aarts E (2015) Personalizing persuasive technologies: explicit and implicit personalization using persuasion profiles. Int J Human-Comput Stud 77:38–51
13. Karapanos E (2016) Designing for different stages in behavior change. arXiv:1603.01369
14. Khenissi S, Abdollahi B, Sun W, Sagheb P, Nasraoui O. (2017) New explainable active learning approach for recommender systems
15. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems, vol 42
16. Kula M (2015) Metadata embeddings for user and item cold-start recommendations. arXiv:1507.08439
17. Liu Y, Yang J (2015) Improving ranking-based rec ommendation by social information and negative similarity. Procedia Comput Sci 55:732–740
18. Liu Y, Yang J (2017) A novel learning-to-rank based hybrid method for book recommendation. In: Proceedings of the international conference on web intelligence, pp 837–842. ACM
19. Mishra P A layman's introduction to principal components. https://hackernoon.com/a-laymans-introduction-to-principal-components-2fca55c19fa0
20. Muhammad Abdullahi A, Orji R, Oyibo K (2018) Personalizing persuasive technologies: do gender and age affect susceptibility to persuasive strategies? In: Adjunct publication of the 26th conference on user modeling, adaptation and personalization, pp 329–334. ACM
21. Oinas-Kukkonen H, Harjumaa M (2018) Persuasive systems design: key issues, process model and system features. In: Routledge handbook of policy design. Routledge, Evanston, pp 105–123
22. Orji R, Mandryk RL, Vassileva J (2015) Gender, age, and responsiveness to cialdini's persuasion strategies. In: International conference on persuasive technology. Springer, Berlin, pp 147–159
23. Pozo M, Chiky R, Meziane F, Métais E (2018) Exploiting past users' interests and predictions in an active learning method for dealing with cold start in recommender systems
24. Rubtsov V, Kamenshchikov M, Valyaev I, Leksin V, Ignatov DI (2018) A hybrid two-stage recommender system for automatic playlist continuation. In: Proceedings of the ACM recommender systems challenge 2018, p 16. ACM
25. Sanchez-Corcuera R, Casado-Mansilla D, Borges CE, López-de ipiña D. (2019) "persuade me!" a user-based recommendation system approach. In: Proceedings of The 5th IEEE international conference on internet of people (IoP 2019)
26. Shaltout NA, El-Hefnawi M, Rafea A, Moustafa A (2014) Information gain as a feature selection method for the efficient classification of influenza based on viral hosts. In: Proceedings of the world congress on engineering, vol 1
27. Smith B, Linden G (2017) Two decades of recommender systems at amazon. com. IEEE Internet Comput 21(3):12–18
28. Willmott CJ, Matsuura K (2005) Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. Clim Res 30(1):79–82
29. Wold S, Esbensen K, Geladi P (1987) Principal component analysis. Chemometr Intell Lab Syst 2(1-3):37–52
30. Wu D, Zhang G, Lu J (2015) A fuzzy preference tree-based recommender system for personalized business-to-business e-services. IEEE Trans Fuzzy Syst 23(1):29–43
31. Yao Y (1995) Measuring retrieval effectiveness based on user preference of documents. J Am Soc Inf Sci 46(2):133–145