Engineering Applications of Artificial Intelligence 96 (2020) 103993

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Recommender systems for sensor-based ambient control in academic facilities

Francisco Pajuelo-Holguera [a], Juan A. Gómez-Pulido [a,*], Fernando Ortega [b]

[a] Dep. Tecnología de Computadores y Comunicaciones, Universidad de Extremadura, Spain
[b] Dep. Sistemas Informáticos, Universidad Politécnica de Madrid, Spain

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
Sensors
Recommender systems
Collaborative filtering
Prediction
Ambient control
Smart facilities

ABSTRACT

Academic spaces are an environment that promotes student performance not only because of the quality of its equipment, but also because of its ambient comfort conditions, which can be controlled by means of actuators that receive data from sensors. Something similar can be said about other environments, such as home, business, or industry environment. However, sensor devices can cause faults or inaccurate readings in a timely manner, affecting control mechanisms. The mutual relationship between ambient variables can be a source of knowledge to predict a variable in case a sensor fails. Moreover, the relationship between these variables and the occupation of spaces by students over time also contains an adequate knowledge of the context for prediction. In this article we propose to predict ambient variables by means of recommendation systems based on collaborative filtering, which are fed with data from sensors over time in different academic rooms. For this purpose, we applied two different algorithms: Probabilistic Matrix Factorization and Bayesian Non-negative Matrix Factorization. The accuracy of the algorithms when comparing actual and predicted values and the performance comparison between the two collaborative filtering implementations lead us to propose Probabilistic Matrix Factorization as a good approach for supporting ambient control systems.

1. Introduction

Sensor devices are increasingly present in all types of facilities in the academic environment, providing data that are monitored for multiple purposes, such as safety, comfort, energy efficiency, etc. This data ecosystem promotes the creation of multiple innovative and specific purpose applications that make life easier for the university community.

The academic spaces used by students for learning are facilities of special interest for sensors, since the use of study and learning time depends not only on the quality of the furniture and technical equipment, but also on the conditions of ambient comfort. These spaces can be classrooms for master classes, laboratories, computer rooms, libraries, study rooms, etc. The ambient conditions of these spaces can be controlled by means of actuators and systems such as thermostats to regulate the temperature, motors to raise and lower blinds, water flow limiters, acoustic warnings of the presence of gases, etc. In any case, control decisions are based on data from sensors that measure ambient parameters such as temperature, humidity and CO_2 concentration, to mention only the three parameters covered by our study.

Sensor devices are electronic systems that can fail at specific times, failing to provide valid readings and causing incorrect decisions by the ambient control systems. A possible solution to this type of event would be to have an intelligent control system that not only detects the failure of the sensor reading, but is also able to predict the data that it would provide, according to the context in which it is set. This context is determined by two factors.

On the one hand, it is reasonable to assume that there is a relationship among the ambient parameters when faced with an external event, such as the arrival of students in the room. In that case, not only the temperature can be increased, but also the humidity and CO_2 levels can be affected. Moreover, this relationship can have a similar behavior in analogous circumstances at other times. Therefore, the correlation among the variations along the time of the ambient parameters measured by the sensors can be an important source of knowledge when predicting the value of the variation of one parameter according to the variations of the other parameters when its corresponding sensor fails.

On the other hand, the evolution or behavior over time of this relationship between the parameters in similar events of room occupation by students, also contains an adequate context knowledge to be taken into account for the prediction.

The intersection of both sources of knowledge can be exploited to design an intelligent method of prediction. For this purpose, in this article we propose the use of Recommender Systems (RS) (Jannach et al., 2011) implemented as CF (Ricci et al., 2011), by linking the times of use of spaces in different sessions over time with the relationship among the ambient parameters measured by sensors during those

* Corresponding author.
E-mail address: jangomez@unex.es (J.A. Gómez-Pulido).

https://doi.org/10.1016/j.engappai.2020.103993
Received 5 June 2020; Received in revised form 7 September 2020; Accepted 2 October 2020
Available online 6 October 2020
0952-1976/

sessions. RS represent a well-proven prediction technique because they analyze in depth the relationship between users and items, evaluated by ranking values. In our case, we consider the relationship between sessions along the time and ambient parameters evaluated by the sensor readings that a session collects from each sensor; therefore, the session ambient conditions are strongly considered in the prediction.

The remainder of this paper is structured as follows. After going over some related works in Sections 2, 3 describes the main features of the academic smart environment where the research was carried out. Next, Section 4 formulates the problem of predicting the values of ambient parameters when a sensor fails. In Section 5, we detail the RS algorithms used for prediction purposes. Next, Section 6 show the experimental design and procedure, the datasets collected from selected spaces and considered for the experiments, the results obtained as error measures when comparing real with predicted values, and a discussion of the results. Last, the conclusion of this work is left for Section 7.

2. Related works

The use of historical data from sensor devices and intelligent techniques as Machine Learning (ML) or CF for different prediction purposes is a wide research emerging area, due to the generalization of IoT devices in heterogeneous contexts and the increasing efficiency of the intelligent algorithms for data analytics. Some research efforts can be found in this line. For example, the long short-term memory (LSTM) artificial neural network was applied to uncover relations with multiple historical sensor data and mental health conditions in order to create a stress recognition model, which can implement an automatic human mental health assistant (Acikmese and Alptekin, 2019). Also, another recurrent neural network was applied to soft sensor data in dynamic chemical processes: singular value decomposition based echo state network (SVD-ESN) (He et al., 2020).

Currently, CF-based RS are intelligent techniques used for both recommendation and prediction purposes. The approach we have taken for this research is prediction, not recommendation. We can find many cases of predictive RS. For example, predicting response in online advertising (Menon et al., 2011), where the problem is estimating the probability that an advertisement is clicked when displayed on a content publisher's webpage. Also predicting of elements in social networks (He and Chu, 2010) where the system is able to predict which elements a user with the same preferences would choose according to the elements chosen before, allowing so to find the most popular trend in RS. Other example is predicting in movies (Odić et al., 2013), where the system will predict the interest of a user for a movie according to the movies seen in the past.

These approaches are based on information available on the Internet where we download the datasets and test our RS. However, we can also use information obtained through sensors. For example, AIRPA v2 (Santos et al., 2016) is an open platform to detect changes in physiological signals acquired from sensors that can be associated with stressful situations, and when this happens, it recommends the learner to relax by delivering modulated sensorial support in terms of light, sound, or vibration at a relaxation breath rate. In this way, by taking advantage of ambient intelligence, the learner can perceive the recommended action without interrupting the learning activity (in this case, practicing the oral exam of a second language). We find another example in DroidOppPathFinder app (Arnaboldi et al., 2013), which recommends users exercise routes in the city by using the local sensors placed on the smartphone, as the GPS sensor. As far as we can see, a RS can use data from heterogeneous sensor sources.

The combination of facilities controlled by sensors and intelligent software applications makes our lives easier. We find a good example in the SEPRE project, which has developed a sensor capable of detecting the position and body movements of the user in bed and, using a predictive algorithm, deducing the moment when the person intends to get up. This application allows the professionals be vigilant to help the elderly people to get up. Sensors are also applied to control failures in the construction of civil works (Bernal et al., 2014) or heterogeneous tasks in academic environments (Bernabeu et al., 2014; Fortes et al., 2019).

Many other works can be found when we consider RS and sensors working together, specially in the IoT context. For example, RS were applied: on smart homes analyzing big data from sensors (Chen et al., 2016); for personal well-being services based on health data (Nouh et al., 2019); for Quality of Service (QoS) health applications based on biometric sensors in wearable electronics and mobile phones; for supporting sensor inputs from heterogeneous data sources (Kuchař and Kliegr, 2017); for capturing the user preferences in physical stores and provide either micro-location marketing or product recommendations from sensors (Symeonidis and Chairistanidis, 2017); and on many other developments and research proposals.

Our research proposal tries to extract knowledge of the behavior of ambient parameters based on the activity of students in smart facilities. In short, the proposal tries to infer knowledge from the context. In this sense, RS have been widely used to process context information (Adomavicius and Tuzhilin, 2015). For this reason RS may be a good option to apply them to our purpose. There are several examples in the literature in this regard: context-aware photo system (Lemos et al., 2012), contextual situation as climate conditions for recommending a holiday destination (Baltrunas et al., 2012), context-aware recommendation of music (Wang et al., 2012) or news (Sotsenko et al., 2014) for smartphones, etc.

3. SmartPolitech: A sensor-based framework in academic facilities

SmartPoliTech is a project developed at the School of Technology at UEX (EPCC) of the University of Extremadura (UEX), Spain. It aims to transform the Center into a large experimental ecosystem: a living-lab for the design, implementation and validation of a wide range of systems capable of creating and managing intelligent spaces. To this end, SmartPoliTech applies different technologies to build an energy-aware space in order to facilitate the social and academic life of the users.

The School of Technology is a University Center located in the city of Cáceres (Fig. 1). It was founded in 1982 and offers undergraduate and postgraduate education in computer science, telecommunications, architecture, and civil engineering. The Center provides the ideal conditions for the implementation of this ambitious project. Several research groups in computing, communications, building and environmental technologies coexist here, as well as several technological spin-offs and start-ups. All this creative potential, with the support of hundreds of undergraduate and master's students who live together in these facilities every year, has been coordinated to develop SmartPoliTech project+5.

The project planned a gradual process of sensor deployment and automation of the different spaces of the center in order to introduce increasing levels of intelligence in each of them and as a whole. The deployment of sensors and actuators in areas delimited and controlled by intelligent elements clearly links up with the concepts developed in Robotics over the last few decades. These concepts have been adapted and extended in recently created fields such as Ambient Intelligence, Inmotics or Ubiquitous Computing. Each space is robotized according to its use and each group of spaces incorporates new intelligent elements. The goals that this intelligence must reach are multiple and sometimes opposed, so it is necessary to include automatic planning algorithms and multi-objective optimization (Lanza-Gutierrez and Gomez-Pulido, 2015), along with learning techniques that increase the knowledge of the system.

Considering all of the above, SmartPoliTech proposes an incremental approach to these objectives, iterating on three phases: sensor and actuator selection, predictive space modeling, and planning and control. In each cycle of this process, all the technologies involved will be evaluated and tested for reliability and sustainability.

Fig. 1. Wi-Fi access points in the buildings of the School of Technology.

3.1. Technology

This project poses different technological challenges to be able to walk the path that goes from a traditional, non-technological space to an intelligent environment that interacts with its inhabitants through applications and web services accessible from mobile devices. An intelligent space like SmartPoliTech is based on the philosophy of open data, which allow students, teachers and researchers to imagine and create new applications and services every day. This requires the development of technologies on five fronts: sensor connectivity, mass storage and high availability, visualization, modeling and planning.

For this purpose, it is necessary to research into highly flexible, scalable and secure open technology communication middleware that allows important features:

- Enabling hundreds or thousands of heterogeneous sensor devices to communicate with a highly accessible cloud storage system.
- Introducing NoSQL databases.
- Exploring the intersection of the IoT with Inmotics and the Open Data philosophy with Data Mining.
- Committing to decoupling data storage from its end uses.

These challenges require a coordinated participation of existing research groups with wide experience in electronics, communications, information systems, data engineering, software engineering, artificial intelligence, machine learning, statistics, acoustics, building, installations, etc., working on a common environment.

3.2. SmartPoliTech IoT and Communication Infrastructure

This section explains the information system on which SmartPoliTech is based: the installation of sensors and their programming guidelines, the databases used, the Zato service bus (ESB) used to connect the sensors, databases and users, and finally the Grafana visualization tool.

Fig. 2 shows an outline of all the components that together make up the SmartPoliTech information system and the transit of SmartPoliTech-generated data through them.

The most noteworthy aspects of this scheme are summarized next. Firstly, a section of views includes sensors. A large number of sensors are distributed throughout the EPCC, most of which have been in continuous operation for the last three years. These sensors generate data related to different variables: ambient temperature, relative humidity, carbon dioxide concentration, water consumption, electricity consumption, etc. This section includes some installation and programming guidelines for the sensors, as well as maps with their location.

The next important aspect of this scheme is the service bus. For this purpose, Zato Enterprise Service Bus (ESB) (Suchojad, 2013) was chosen. It is an open source software with commercial and community support, mainly consisting of a request server written in Python that can be used to build middleware and backend systems. SmartPoliTech uses Zato as a host for services that facilitate data integration to the databases, sending data from the sensors to the databases, requesting databases by the users, and collecting information from the Wi-Fi access points, among other outputs.

Another aspect to highlight is the persistence, which is represented by InfluxDB and Neo4J databases in Fig. 2. InfluxDB is an open source time series database that fits perfectly with the type of data generated by SmartPoliTech (mainly, they are time series) and allows integration with the Grafana visualization tool in a very simple way. Neo4J is a graph oriented database used for building a maintenance system with the operations of creating, reading, updating, and deleting on a data network model. It is used in online processing systems, so SmartPoliTech applies it to obtain an ordered structure of the facilities and the sensors deployed through them, allowing generating interactive maps available to the EPCC users.

Table 1
Sensor features.

Measure	Temperature	CO_2	Humidity
Units	° C	%ppm	%RH
Resolution	0.1	N/D	1
Range	−40 to +80	400 to 10,000	0 to 100
Accuracy	+/ −0.5	+/−3	+/−3

3.3. Sensor devices

SmartPoliTech has currently a wide network of sensors (> 200 devices) that perform measurements on heterogeneous ambient and energy parameters. Infrastructure parameters such as the state of windows and blinds are also collected, as well as occupation parameters by means of open sessions at wireless network access points.

The data considered in this work were collected from temperature and humidity sensors via Wi-Fi. The deployment of these sensor devices is simple and fast, as they are wireless and battery powered. They were designed under Arduino architecture and manufactured by the company *Ray Ingeniería Electronica S.L.* (Anon, 2020), whose headquarters are in the city of Cáceres. The Arduino architecture of the sensors allows the users to modify the firmware through the IDE and ICSP programmer, according to the users' preferences and purposes.

Table 1 lists the particular features of the sensors considered for this work, mounted on two devices: DHT22 (temperature and humidity) and RS485 (CO_2) (Fig. 3).

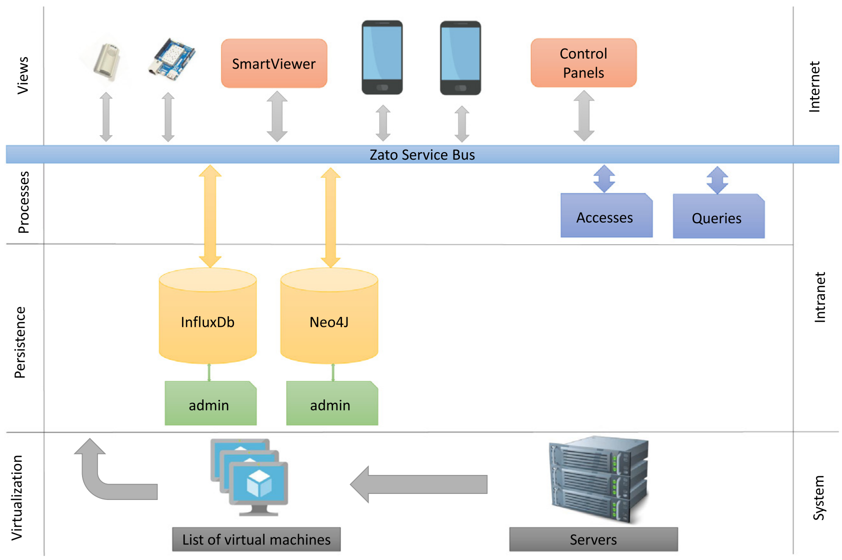

Fig. 2. SmartPoliTech infrastructure and data flow.

Fig. 3. Sensor devices deployed in SmartPoliTEch considered for this work.

3.4. Monitoring solution

Grafana tool is used for visualizing the SmartPoliTech data, because it allows an easy and fast integration with InfluxDB. It is the most commonly used tool for visualizing time-series data for infrastructure and analysis applications, but it is also used in other domains, including industrial sensors and process control. Fig. 4 shows a real screen capture of the SmartPoliTech project.

Grafana platform can detect if a sensor fails, taking the corresponding action. In this sense, the tool was configured to send a control message in case of unusual readings. For example, if a temperature sensor registers a very high value in winter, a warning message is sent in order to identify two possible causes: the sensor has broken down or there is a problem in the classroom where the sensor is located.

4. Problem formulation

Sensor devices deployed in smart academic spaces, on occasion, may have malfunctions or just stop working. Erroneous readings can lead to undesired control decisions by the actuator systems in charge of ambient conditions of comfort and safety. This is the starting point of our research: What can be done when a sensor fails so that the control system can continue to act with some reliability?

Fig. 5 shows this starting point on the left side. Let us suppose that a laboratory session takes place during a certain time interval, in which the mere presence of the students implies a change in the sensor readings with respect to their initial values. In this way, we can describe the impact of the session progress on the ambient parameters as the variation of the same, expressed as the absolute value of the difference between the initial and final readings. For example, if P is the ambient parameter (T, C or H), $PV = |P_{end} - P_{ini}|$ is the absolute variation considered (TV, CV or HV).

These sessions take place over time, on possibly different days and at different times. If a sensor failure occurs in any of these sessions, our proposal is to develop an intelligent system capable of replacing the erroneous data with another of fair value, which has been calculated taking into account two dimensions: the current context and the historical context.

- The current context has to do with the relationship among the different ambient parameters during the same session. The variations TV, CV and HV are clearly due to the presence and activity developed by the students in the room, even in the mutual influence among the parameters.
- The historical context has to do with the mutual relationship between TV, CV and HV over time, that is, their relationship throughout the different sessions. It is reasonable to assume that, if the presence of students is the main source of variations TV, CV and HV, the relationship between them is similar in the different sessions of the same subject, as they all share the same characteristic of activity (same number of students and same materials used). The exception would be if a session corresponds to a different subject or activity where other causes have to be added to the students' impact (such as the influence of different equipment).

The TV, CV and HV variations recorded for the different sessions build what we call the parameter matrix. The rows of this matrix are the sessions and the columns are the variations of the ambient parameters, and the matrix cells are the values of these variations in the corresponding sessions. Thus, if the temperature sensor fails during the session i, the value of the variation TV will be unknown. However, taking into account the current and historical contexts explained before, this TV value can be calculated from the TV values recorded in the different sessions, as well as the variations of the rest of the ambient parameters during the session i. This is the basis of collaborative filtering. Therefore, we propose to use a recommender system capable of calculating the predicted value TVP, which corresponds to the unknown variation of the temperature in the session i.

The benefits of this intelligent solution are clear: there is no need to replace a sensor in real time when it fails so that the ambient control system can continue to perform well.

5. Recommender systems as predicting tool

As we said before, RS can be applied for prediction purposes. In a classical description, a RS considers the ratings that a set of users

Fig. 4. Grafana platform running for SmartPoliTech project.

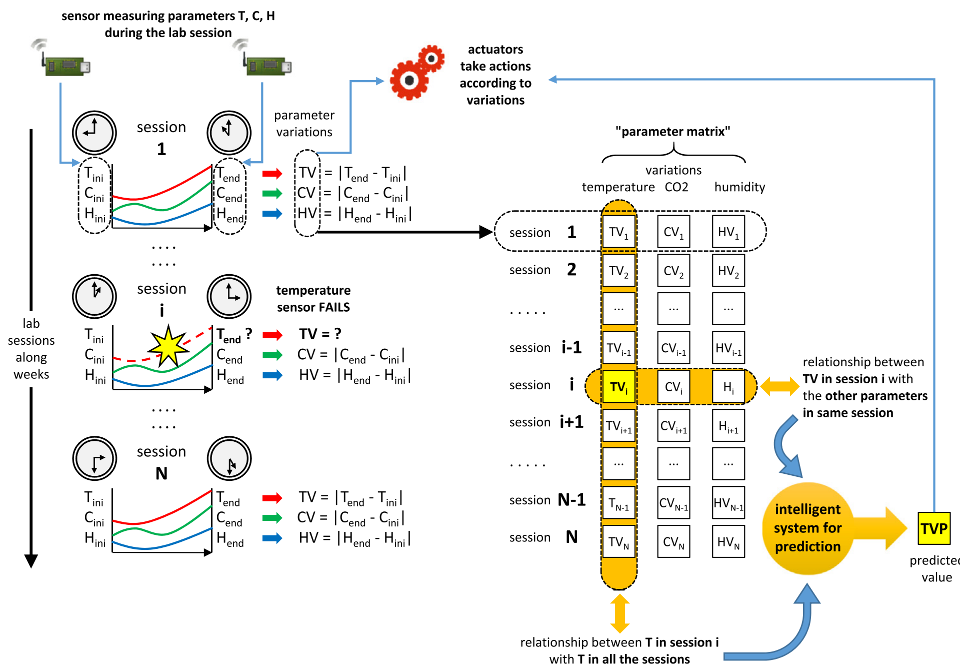

Fig. 5. Proposal for predicting parameters after failures.

have given to a set of items. Then, we have n users, m items, and a rating matrix $R \in \mathbb{R}^{n \times m}$. We follow these terms in order to explain the method, and later we map them to the problem that we have tackled in SmartPoliTech.

Matrix Factorization (MF) is a technique that splits a matrix $X \in \mathbb{R}^{n \times m}$ into two matrices $U \in \mathbb{R}^{n \times k}$ and $V \in \mathbb{R}^{k \times m}$, so that $X \approx U \cdot V$ (Lee and Seung, 1997). The factorized matrices U and V contain a compact representation of X. Particularly for CF, MF based RS factorize R that contains the set of known ratings of n users to m items (Koren et al., 2009).

MF is based on the assumption that the ratings of the users to the items are related by a subset of latent factors k intrinsic to the users and items. Therefore, the factorization generates two new matrices, $P \in \mathbb{R}^{n \times k}$ and $Q \in \mathbb{R}^{m \times k}$, where P represents the k-latent factors of the n users and Q represents the k-latent factors of the m items.

After factorizing R, the rating predictions ($\hat{r}_{ui}$) of a user u to the item i is calculated as the dot product of the row vector of the matrix P that contains the latent factors of the user u ($\vec{p}_u$) and the column vector of the matrix Q that contains the latent factors of the item i ($\vec{q}_i$) (1). Hence, the learning process finds the optimal parameters for the matrices P and Q that verifies the condition given in (2).

$$\hat{r}_{ui} = \vec{p}_u \cdot \vec{q}_i^T \tag{1}$$

$$R \approx P \cdot Q^T \tag{2}$$

This process is usually raised as an optimization problem in which the quadratic difference between the know ratings ($r_{u,i}$) of the matrix R and the predicted ones ($\vec{p}_u \cdot \vec{q}_i^T$) must be minimized (3).

$$\min_{\vec{p}_u, \vec{q}_i} \sum_{(u,i) \in R} (r_{u,i} - \vec{p}_u \cdot \vec{q}_i^T)^2 \tag{3}$$

In this work we have applied two approaches of CF algorithms in order to study the behavior of the problem with more than one option.

5.1. Probabilistic Matrix Factorization (PMF)

PMF (Mnih and Salakhutdinov, 2008) is a popular implementation of MF applied to CF. It performs the factorization thorough a probabilistic model that represents interaction between users and items.

Fig. 6 represents the probabilistic model through three elements: circles that symbolize random variables; arrows between two variables,

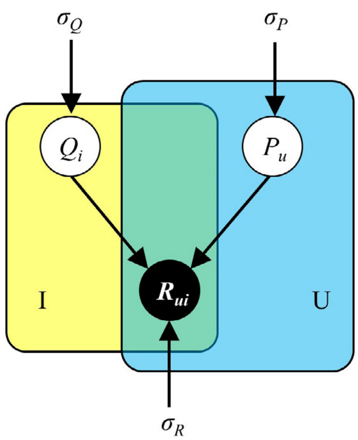

Fig. 6. Graphical representation of PMF model.

Algorithm 1: PMF algorithm

input : R, K, λ, γ
output: P, Q
Create a random matrix P with U rows and K columns
Create a random matrix Q with I rows and K columns
repeat
  for *each user u* **do**
    for *each item i rated by user u* **do**
      error = $R[u][i]$ - dotProduct($P[u]$, $Q[i]$)
      for *each factor k* **do**
        $P[u][k] += \gamma \cdot (error \cdot P[u][k] - \lambda \cdot Q[i][k])$
  for *each item i* **do**
    for *each user u that has rated the item i* **do**
      error = $R[u][i]$ - dotProduct($P[u]$, $Q[i]$)
      for *each factor k* **do**
        $Q[i][k] += \gamma \cdot (error \cdot Q[i][k] - \lambda \cdot P[u][k])$
until *convergence*
return P, Q

that indicate dependence between those random variables; and rectangles, that indicate repetitions of the random variables. Black or white circles indicate if the random variables are observed or must be learned, respectively. We see three random variables: R_{ui} (the rating of the user u to the item i), P_u (the latent factors of each user u), and Q_i (the latent factors of each item i). The arrows between P_u and Q_i with R_{ui} indicate the dependency between the rating of the user u to the item i and the latent factors of the user u and the item i. PMF assumes a Gaussian distribution for all the random variables. Finally, σ_R, σ_P and σ_Q are the model hyper-parameters.

Algorithm 1 details how PMF works. The inputs are the rating matrix R, the number of latent factors K, and the hyper-parameters λ and γ. The outputs are the latent factors matrices P and Q learned from R.

5.2. Bayesian Non-negative Matrix Factorization (BNMF)

BNMF (Hernando et al., 2016) model is an accurate factorization model designed for CF based RS, which factorizes the rating matrix in a probabilistic way. The main objective of BNMF is to provide an understandable probabilistic meaning of the latent factors space generated as consequence of the factorization process. To this end, the model has been designed for representing better the interaction between users and items. A discrete distribution is used to represent ratings, instead of assuming a continuous distribution like Gaussian.

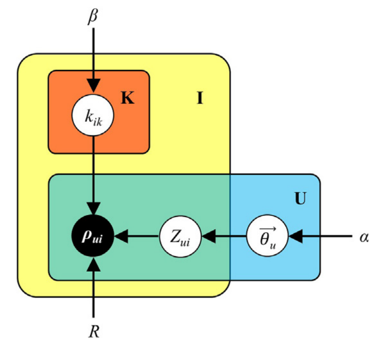

Fig. 7. Graphical representation of BNMF model.

This matches the reality of most CF systems, where users must rate items on a pre-set scale of score values.

The BNMF model, graphically represented by Fig. 7, is composed by the following random variables:

- $\vec{\theta}_u$ is a K dimensional vector from a Dirichlet distribution. This random variables are used to represent the probability that a user belongs to each group.
- κ_{ik} from the Beta distribution is used to represent the probability that a user in the group k likes the item i.
- Z_{ui} from the Categorical distribution is used to represent that the user u rates the item i as if he or she belongs to the group k.
- ρ_{ui} from the Binomial distribution is used to represent the observable rating of the user u to the item i.

The model also contains the following hyper-parameters:

- α is related to the possibility of obtaining overlapping groups of users sharing the same preferences.
- β is related to the amount of evidences required to belong to a group.
- K is related to the number of groups (i.e. number of latent factors) that exists in the dataset.
- R is related to the Binomial distribution which takes values from 0 to R.

As the first step to apply BNMF as prediction method, we must determine the conditional probability distribution of the non-observable random variables given a set of observations (i.e. the known ratings). Applying the variational inference technique (Hoffman et al., 2013), we can obtain the algorithm to perform this task. Algorithm 2 contains a detailed explanation about the training phase of BNMF model (Hernando et al., 2016).

Table 2 lists the values of the hyper-parameters applied for PMF and BNMF experiments. These values have been tuned using a Grid Search approach, in which a wide range of values for each hyper-parameter has been evaluated in order to minimize prediction error.

6. Results

In this section we describe the datasets used in the collaborative filtering experiments, the experimental procedure, and the performance comparison based on the error made in the prediction.

6.1. Datasets

Table 3 shows the main features related to the datasets used in the experiments, freely available in Gomez-Pulido et al. (2020). Six laboratory classrooms were chosen according to the use of groups of students of different subjects. Each space was used for a certain number of sessions during several weeks. The heterogeneity of spaces, number

Algorithm 2: BNMF algorithm. The algorithm returns the latent factors for each user and item. Input ratings (r_{ui}) must be normalized.

input : r_{ui}, α, β, K, R
output: p_{uk}, q_{ik}
temp : γ_{uk}, ϵ^-_{ik}, ϵ^+_{ik}, λ_{uik}, λ'_{uik}
Initialize γ_{uk}
Initialize ϵ^-_{ik}
Initialize ϵ^+_{ik}
repeat
  for *each user* u **do**
    for *each item* i *rated by user* u **do**
      for *each factor* k **do**
        $\lambda'_{uik} \leftarrow exp(\Psi(\gamma_{uk}) + r^+_{ui} \cdot \Psi(\epsilon^+_{ik}) + r^-_{ui} \cdot \Psi(\epsilon^-_{ik}) - R \cdot \Psi(\epsilon^+_{ik} + \epsilon^-_{ik}))$
      for *each factor* k **do**
        $\lambda_{uik} \leftarrow \frac{\lambda'_{uik}}{\lambda'_{ui1} + \cdots + \lambda'_{uiK}}$
  for *each item* i **do**
    $\epsilon^+_{ik} \leftarrow \beta$
    $\epsilon^-_{ik} \leftarrow \beta$
  for *each user* u **do**
    $\gamma_{uk} \leftarrow \alpha$
    for *each item* i *rated by user* u **do**
      for *each factor* k **do**
        $\gamma_{uk} \leftarrow \gamma_{uk} + \lambda_{uik}$
        $\epsilon^+_{ik} \leftarrow \epsilon^+_{ik} + \lambda_{uik} \cdot R \cdot r_{ui}$
        $\epsilon^-_{ik} \leftarrow \epsilon^-_{ik} + \lambda_{uik} \cdot R \cdot (1 - r_{ui})$
until *convergence*
for *each factor* k **do**
  for *each user* u **do**
    $p_{uk} \leftarrow \frac{\gamma_{uk}}{\sum_{f=1..K} \gamma_{uf}}$
  for *each item* i **do**
    $q_{ik} \leftarrow \frac{\epsilon^+_{ik}}{\epsilon^+_{ik} + \epsilon^-_{ik}}$

Table 2
Hyper-parameters applied in the experiments.

PMF		BNMF	
Number of factors	4	Number of factors	5
Number of iterations	50	Number of iterations	20
Learning rate β	0.2	Overlapping rate α	0.2
Regularization factor λ	0.05	Number of evidences β	10

of students and subjects allows introducing a variety of behaviors very useful to evaluate the response of a solution based on collaborative filtering.

During all the sessions recorded in these spaces there were no sensor failures or unusual readings, which is a requirement for the further estimation of the prediction error.

6.2. Experimental procedure

Our proposal is to predict variations of ambient parameters in case of sensor failures, as we pointed out in Fig. 5. We think that this goal can be tackled by CF as prediction tool, considering the motivations described before about current and historical contexts. To this end, we map the corresponding terms as Table 4 describes. Note that, according to these definitions, variations TV, CV and HV (for PMF) correspond to item(1), item(2) and item(3), respectively.

When building the ranking matrix, one important consideration must be taken into account: PMF can work with rankings corresponding

Table 3
Characteristics of the datasets considered for the experiments.

Space	Sessions	Students	Subject	Cells
L1	36 (2h each)	20	Data structures	72
L2	15 (3h each)	15	Computer foundations	
L2	7 (4h each)	15	Processor design	
L2	12 (2h each)	12	Design of OS	
L2	11 (4h each)	14	Parallel computing	
L2	45			135
LD	18 (6h each)	12	Computer structure	
LD	28 (4h each)	12	Computer foundations	
LD	46			138
C3	24 (5h each)	40	Informatics-I	
C3	26 (5h each)	40	Informatics-II	
C3	50			150
T3	30 (5h each)	20	Telecomm.-III	
T3	22 (5h each)	20	Telecomm.-IV	
T3	52			156
C2A	40 (2h each)	15	English	
C2A	33 (4h each)	10	Master technology	
C2A	73			219

Table 4
Terms mapping between CF and our research proposal.

CF	Proposal
Ranking matrix R	Parameter matrix
n users u	N sessions
m items i	Variations of three ambient parameters
Rating r	Sensor measure
Predicted rating $\hat{r}$	Predicted variations

to continuous variables (real values), while BNMF can only work with discrete values. In our case, the ranking matrix is built from the variations of the ambient parameters (for example, from 0.0 to 5.8); therefore, for BNMF these values should be discretized to give, for example, rankings with only 10 possible values that go from "very small variation" to "very large variation".

To explain the experimental procedure we use the nomenclature defined in Table 5. To simplify the description, we identify P from now on with the ambient parameter, which can be T, C or H.

Fig. 8 summarizes the experimental procedure designed for our proposal. Let us suppose a classroom where students attend a subject during a certain period of time. For this session, we record the values of the ambient parameter P at the beginning and at the end, by means of the readings collected from the corresponding sensor.

To build the rating matrices, we have to take into account that PMF can handle continuous variables while BNMF can only handle discrete variables. Thus, for PMF we consider as rating the absolute variation of P during the session, PV, whereas BNMF needs a discretization mechanism.

The discretization mechanism starts by calculating PP as the relative absolute variation of P along the session. Then, knowing the minimum and maximum values of PP for all sessions, we normalize PP between 0.0 and 1.0. Finally, the normalized result is rounded up to the nearest first decimal dot and then multiplied by 10 to obtain the rating PD. This rating may correspond to one of the 11 possible values from 0 (*no variation*) to 10 (*maximum possible variation*). For example, Fig. 9 shows the ratings TD for the temperature column in the BNMF rating matrix, obtained after applying the discretization mechanism for the six datasets considered.

We assume as many prediction cases as there are cells in the ratings matrix. For example, for cell P_i that represents the rating experienced by the ambient parameter P during the session i, we assume that it is an unknown value because of the failure of the corresponding

Table 5
Nomenclature for the experiments.

Ambient parameter	P (T, C, H)	
Measure at the start of the session	P_{ini}	
Measure at the end of the session	P_{end}	
	PMF	BNMF
% of absolute variation along the session		$PP = \lvert \frac{P_{end}}{P_{ini}} - 1 \rvert$
Min/max PP in all sessions		PP_{min}, PP_{max}
Rating matrix	PV[a]$= \lvert P_{end} - P_{ini} \rvert$	PD[b]$= round.1(\frac{PP-PP_{min}}{PP_{max}-PP_{min}}) \times 10$
Predicted value	PVP	PDP
Prediction error	$PVP_e = \lvert PVP - PV \rvert$	$PDP_e = \lvert PDP - PD \rvert$
% of prediction error	$PVP_{ep} = \lvert \frac{PV}{PVP} - 1 \rvert$	$PDP_{ep} = \lvert \frac{PD}{PDP} - 1 \rvert$

[a]Absolute variation along the session.
[b]% Absolute variation, normalized and discretized.

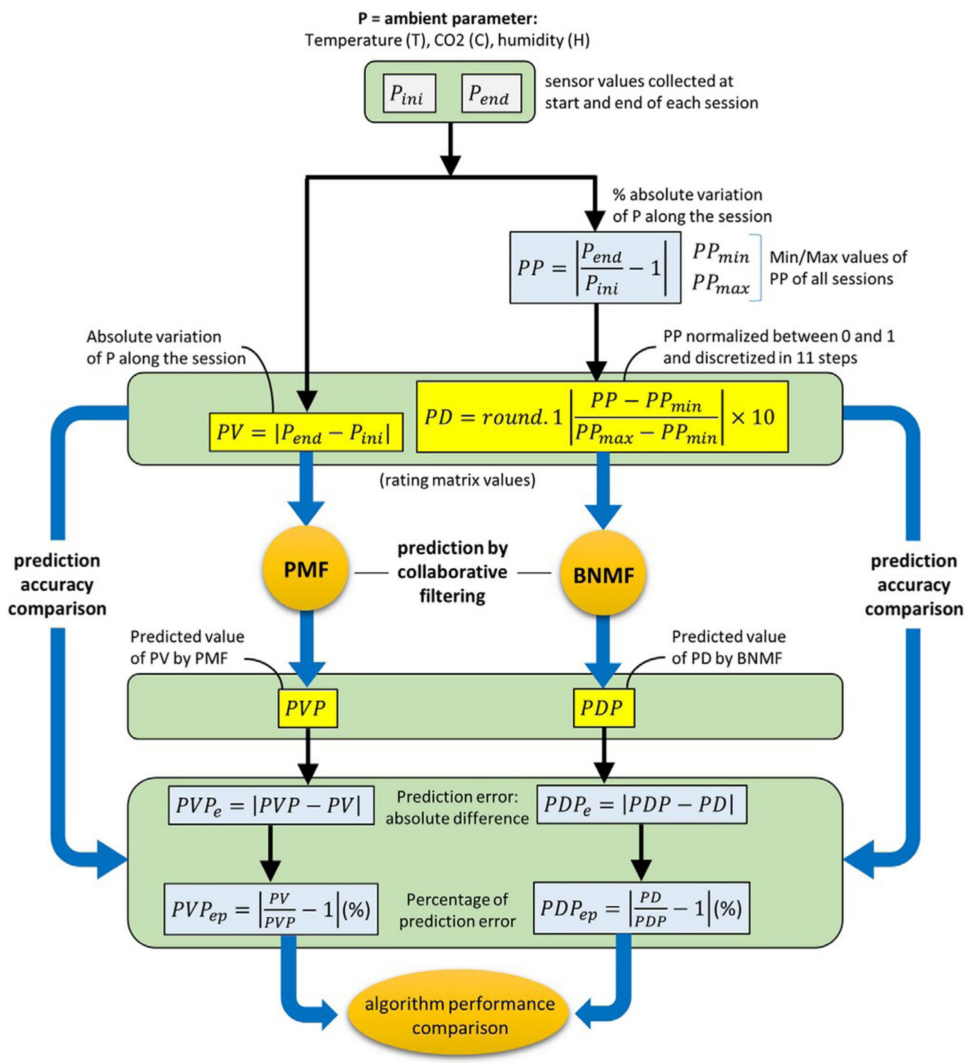

Fig. 8. Experimental procedure.

sensor. Therefore, we obtain the predicted value PVP_i and PDP_i after applying PMF and BNMF respectively. As we know the real values PV_i and PD_i, we can calculate the absolute errors $PVPe_i$ and $PDPe_i$, which can also be represented as a percentage errors of the predicted ratings with respect to the real $PVPep_i$ and $PDPep_i$. Please, note that here you can replace P by any of the ambient parameter variations T, C or H.

As PMF and BNMF are stochastic algorithms, the predicted values may differ in each run. For this reason, each value chosen as predicted was the mean of 41 runs, where the minimum, maximum, mean, median and standard deviation values were recorded.

Another aspect to highlight is that the CF algorithms perform the prediction using training and a test datasets. We considered that the training dataset is composed of all the data of the ranking matrix, while the test dataset was chosen following a diagonal strategy. This method chooses the test data following a zigzag path that goes across the rows and columns of the matrix uniformly, as Fig. 10 shows for L1 dataset. The reason of proposing this method to build the test dataset is that

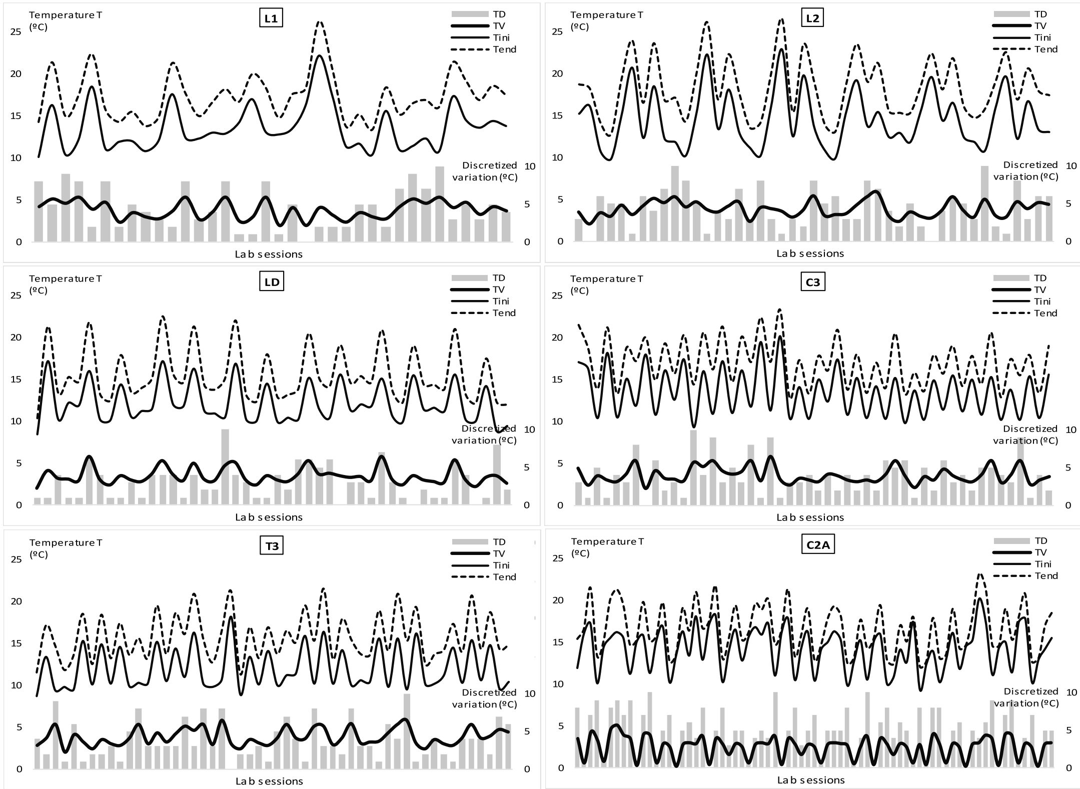

Fig. 9. Continuous and discretized variations for temperature.

sessions	PMF rating matrix (L1) TV	CV	HV	BNMF rating matrix (L1) TD	CD	HD
1	4.2	3	3.3	8	5	2
2	5.1	4	2.2	5	7	1
3	4.6	3	3.6	9	3	3
4	5.2	3	2.6	8	4	2
5	3.9	2	2.9	2	1	4
6	4.7	3	2.2	8	4	1
7	2.4	2	2.4	2	1	2
8	3.5	2	3.6	5	0	4
9	3	2	1.9	4	0	0
10	2.8	4	2.2	3	7	2
11	3.7	3	1.8	2	3	3
12	5.3	2	1.9	8	0	1
13	2.7	5	3.3	3	8	3
14	3.6	2	2.8	4	1	2
15	5.3	4	1.8	8	7	0
16	2.5	3	2.2	1	3	1
17	3	3	1.9	1	3	5
18	5.3	2	2.2	8	1	1
19	2	3	2.4	1	4	2
20	4.1	4	3.6	5	7	5
21	2	2	2.4	0	0	4
22	4.1	2	2.8	2	0	10
23	3.2	3	1.9	2	3	3
24	2.4	2	2.2	2	0	1
25	3.5	4	3.8	5	5	5
26	3	3	3.6	5	3	2
27	2.8	2	3.2	2	0	4
28	4.2	5	2.8	7	10	1
29	5.1	2	2.9	9	1	2
30	4.6	2	2.2	7	1	1
31	5.3	3	2.8	10	4	1
32	4.1	4	2.7	3	5	3
33	4.7	3	2.2	5	3	1
34	3.3	2	2.4	3	0	1
35	4.2	3	2.1	5	4	1
36	3.7	4	3.8	4	8	4

Training data Test data

Fig. 10. Strategy for selecting training and test data.

we guarantee that the test data represent most of sessions and ambient parameters with a similar weight.

6.3. Discussion

Once the predicted matrix has been generated after applying both PMF and BNMF for all the variations of the ambient parameters in the different facilities along the time, we can check not only the accuracy of the algorithm when comparing actual and predicted values, but the performance comparison between the two CF implementations.

Fig. 11 shows the actual variations (solid line) and predicted variations (dashed line) obtained by PMF along the time (defined as laboratory sessions), for of each ambient parameter (T, C, and H) and classroom.

At first glance, the prediction curves look very similar to the real curves in all the cases. The first conclusion is that the prediction proposal based on collaborative filtering works reasonably well. This good behavior of the prediction is not only based on the degree of similarity between the curves, but also by the absence of outliers, which points out to the reliability of the proposed method.

Notwithstanding the good behavior of the prediction according to the appearance of these curves, it is more meaningful to quantify the prediction accuracy by considering the rates of the prediction error: $PVPep$ for PMF and $PDPep$ for BNMF. This way, we can compare the results obtained by both CF implementations, since they handle different measures of the variations of the ambient parameters (in the continuous and discrete domains), as Table 5 shows. Thus, Fig. 12 reports the percentages of prediction errors when applying PMF and BNMF. Analyzing this figure, we can draw some interesting conclusions.

First, PMF reaches much better results than BNMF, which is also checked by calculating the mean of these rates, as Table 6 shows. The good behavior of PMF is widespread, except in one case, when the temperature was monitored in the C2A classroom. However, there could have been some anomalous circumstances with the temperature sensor in this classroom, observing the behavior of the temperature variations drawn in Fig. 11. Even without considering this classroom, the temperature prediction error rate would fall to 8%. In any case, we clearly bet on PMF as the CF implementation to be applied for sensor-based ambient control.

We also observe that not all the ambient parameters have the same prediction accuracy. As we can see in Table 6, the humidity prediction is more accurate than temperature and CO2.

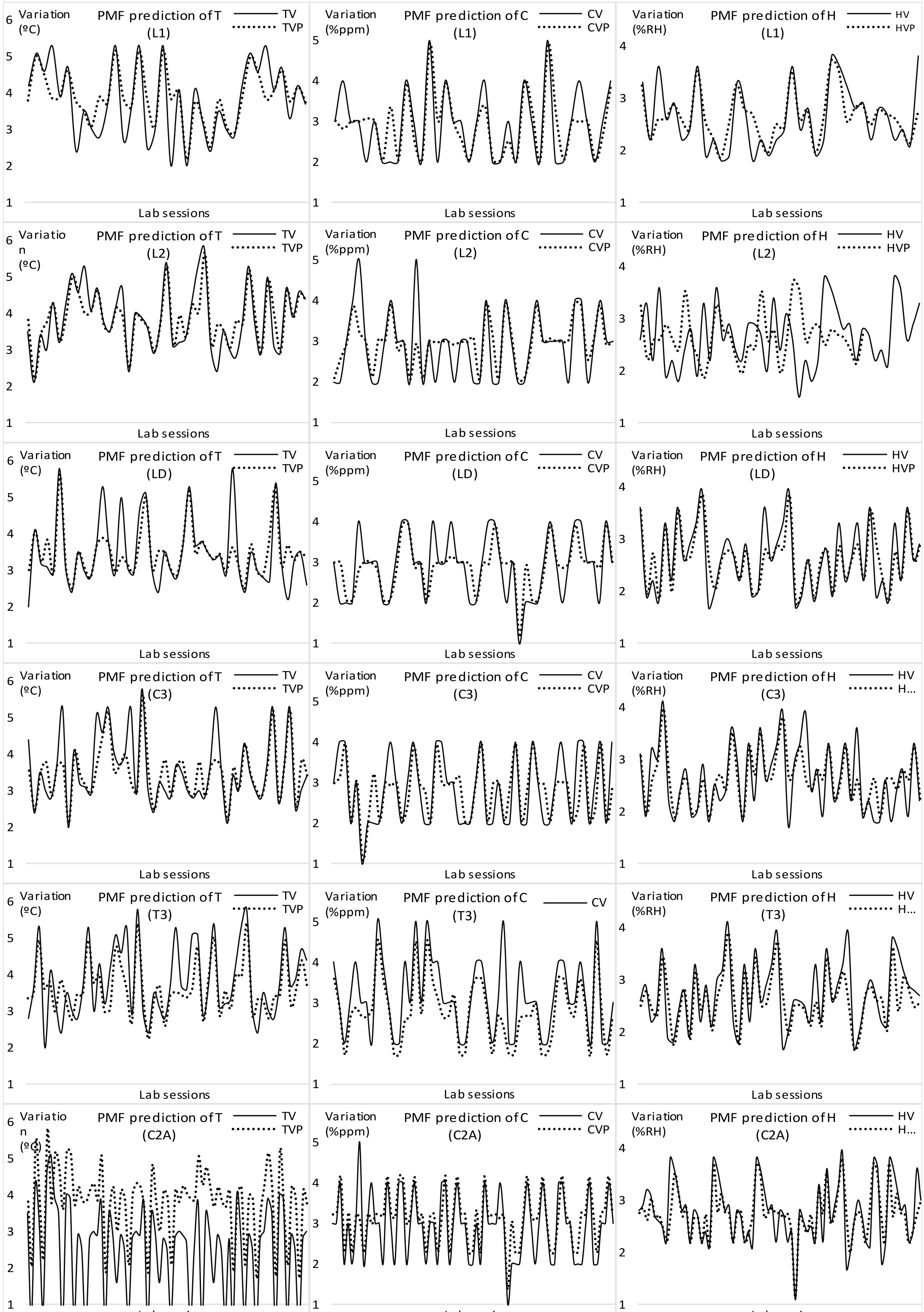

Fig. 11. Actual and predicted variations of the ambient parameters by PMF.

7. Conclusion

As a final conclusion, we can say that our proposal for predicting the variation of ambient parameters by means of CF-based RS represents a valid approach to the problem of covering the lack of valid readings to ambient control systems when a sensor fails.

The novelty of this proposal lies in the fact that the prediction of an ambient parameter takes into account its behavior in all the registered sessions, and in the behavior of the remaining ambient parameters in the same session, following a collaborative hypothesis. Although the results obtained by PMF are reasonably good, we highlight the reliability of the proposal as there are no cases of anomalous predictions.

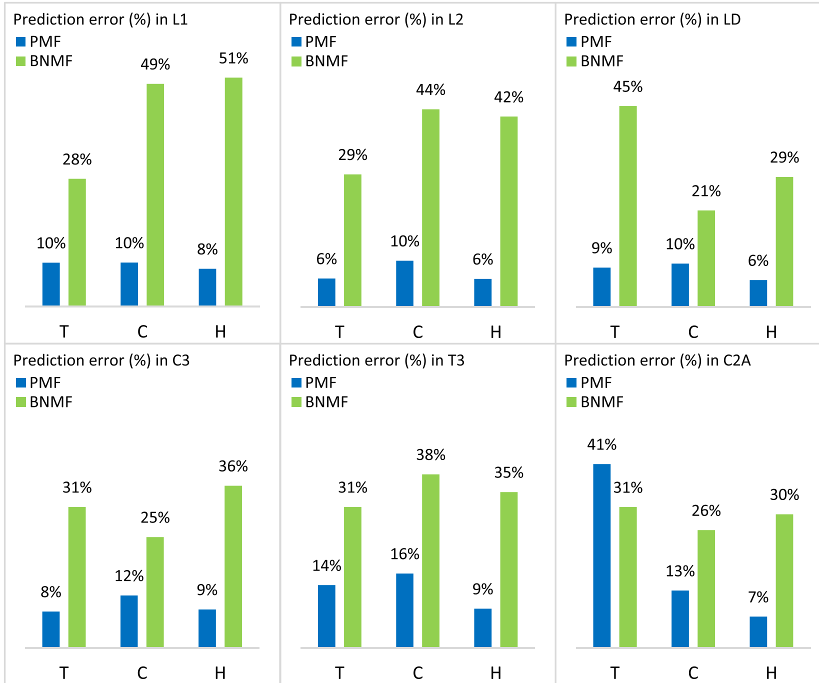

Fig. 12. Prediction error (%) of the ambient parameters by PMF and BNMF.

Table 6
Mean of the prediction error rates.

	PMF	BNMF
Temperature	15%	33%
CO_2	15%	33%
Humidity	15%	33%

We consider that PMF is a good starting point for future improvements. We think that better results could be reached by tuning efficiently the hyper-parameters and by considering more historical data of the sensor readings, since they allow us a better understanding and modeling of the relationship between the students' behavior and the variations of the sensor readings.

CRediT authorship contribution statement

Francisco Pajuelo-Holguera: Investigation, Data curation, Software, Visualization, Validation. **Juan A. Gómez-Pulido:** Conceptualization, Methodology, Writing - original draft. **Fernando Ortega:** Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank to the entities that funded this research project: the Government of Extremadura (Spain) under the grant IB16002, and by the ERDF (European Regional Development Fund, EU) and the Spanish State Research Agency under the contract TIN2016-76259-P.

References

Acikmese, Y., Alptekin, S.E., 2019. Prediction of stress levels with LSTM and passive mobile sensors. In: Knowledge-Based and Intelligent Information and Engineering Systems: Proceedings of the 23rd International Conference KES2019. Procedia Comput. Sci. 159, 658–667. http://dx.doi.org/10.1016/j.procs.2019.09.221, URL http://www.sciencedirect.com/science/article/pii/S187705091931405X.

Adomavicius, G., Tuzhilin, A., 2015. Context-aware recommender systems. In: Recommender Systems Handbook. Springer, pp. 191–226.

Anon, 2020. Ray Ingeniería Electrónica. URL https://www.ray-ie.com/.

Arnaboldi, V., Conti, M., Delmastro, F., Minutiello, G., Ricci, L., 2013. DroidOppPathFinder: A context and social-aware path recommender system based on opportunistic sensing. pp. 0–2. http://dx.doi.org/10.1109/WoWMoM.2013.6583363.

Baltrunas, L., Ludwig, B., Peer, S., Ricci, F., 2012. Context relevance assessment and exploitation in mobile recommender systems. Pers. Ubiquitous Comput. 16 (5), 507–526. http://dx.doi.org/10.1007/s00779-011-0417-x.

Bernabeu, J.M., Berna-Martinez, J., Maciá Pérez, F., 2014. Smart sentinel: Monitoring and prevention system in the smart cities. Int. Rev. Comput. Softw. 9, 1554–1559. http://dx.doi.org/10.15866/irecos.v9i9.2972.

Bernal, D., Ma, Q., Castro-Triguero, R., Gallego, R., 2014. Sensor placements for damage localization with the SDLV approach. In: Conference Proceedings of the Society for Experimental Mechanics Series, Vol. 6. pp. 347–353. http://dx.doi.org/10.1007/978-3-319-04729-4_30, cited By 0.

Chen, H., Xie, X., Shu, W., Xiong, N., 2016. An efficient recommendation filter model on smart home big data analytics for enhanced living environments. Sensors 16, 1706. http://dx.doi.org/10.3390/s16101706.

Fortes, S., Santoyo Ramón, J., Palacios Campos, D., Baena, E., Mora-Garcí a, R.o., Medina, M., Mora, P., Barco, R., 2019. The campus as a smart city: University of Málaga environmental, learning, and research approaches. Sensors 19, 1349. http://dx.doi.org/10.3390/s19061349.

Gomez-Pulido, J.A., Pajuelo-Holguera, F., Ortega, F., 2020. Data for: Recommender systems for sensor-based ambient control in academic facilities. http://dx.doi.org/10.17632/7j845nz5wh.2, Mendeley Data, V2.

He, J., Chu, W.W., 2010. A social network-based recommender system (SNRS). In: Memon, N., Xu, J.J., Hicks, D.L., Chen, H. (Eds.), Data Mining for Social Network Data. Springer US, Boston, MA, pp. 47–74. http://dx.doi.org/10.1007/978-1-4419-6287-4_4.

He, Y.-L., Tian, Y., Xu, Y., Zhu, Q.-X., 2020. Novel soft sensor development using echo state network integrated with singular value decomposition: Application to complex chemical processes. Chemometr. Intell. Lab. Syst. 200, 103981. http://dx.doi.org/10.1016/j.chemolab.2020.103981, URL http://www.sciencedirect.com/science/article/pii/S0169743919308184.

Hernando, A., Bobadilla, J., Ortega, F., 2016. A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. Knowl.-Based Syst. 97, 188–202.

Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J., 2013. Stochastic variational inference. J. Mach. Learn. Res. 14 (1), 1303–1347.

Jannach, D., amd A. Felfernig, M.Z., Friedrich, G., 2011. Recommender Systems. An Introduction. Cambridge University Press, Cambridge, United Kingdom.

Koren, Y., Bell, R., Volinsky, C., 2009. Matrix factorization techniques for recommender systems. Computer (8), 30–37.

Kuchař, J., Kliegr, T., 2017. InBeat: JavaScript recommender system supporting sensor input and linked data. Knowl.-Based Syst. 135, 40–43. http://dx.doi.org/10.1016/j.knosys.2017.07.026, URL http://www.sciencedirect.com/science/article/pii/S0950705117303428.

Lanza-Gutierrez, J.M., Gomez-Pulido, J.A., 2015. Assuming multiobjective metaheuristics to solve a three-objective optimisation problem for Relay Node deployment in Wireless Sensor Networks. Appl. Soft Comput. 30, 675–687. http://dx.doi.org/10.1016/j.asoc.2015.01.051, URL http://www.sciencedirect.com/science/article/pii/S1568494615000721.

Lee, D.D., Seung, H., 1997. Unsupervised learning by convex and conic coding. In: Advances in Neural Information Processing Systems. pp. 515–521.

Lemos, F., Carmo, R., Viana, W., Andrade, R., Towards a context-aware photo recommender system, in: CEUR Workshop Proceedings, Vol. 889.

Menon, A.K., Chitrapura, K.-P., Garg, S., Agarwal, D., Kota, N., 2011. Response prediction using collaborative filtering with hierarchies and side-information. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '11, Association for Computing Machinery, New York, NY, USA, pp. 141–149. http://dx.doi.org/10.1145/2020408.2020436.

Mnih, A., Salakhutdinov, R.R., 2008. Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems. pp. 1257–1264.

Nouh, R., Lee, H.-H., Lee, W.-J., Lee, J.-D., 2019. A smart recommender based on hybrid learning methods for personal well-being services. Sensors 19, 431. http://dx.doi.org/10.3390/s19020431.

Odić, A., Tkalčič, M., Tasič, J.F., Košir, A., 2013. Predicting and detecting the relevant contextual information in a movie-recommender system. Interact. Comput. 25 (1), 74–90. http://dx.doi.org/10.1093/iwc/iws003, arXiv:https://academic.oup.com/iwc/article-pdf/25/1/74/2304742/iws003.pdf.

Ricci, F., Rokach, L., Shapira, B., 2011. Introduction to recommender systems handbook. In: Recommender Systems Handbook. Springer, pp. 1–35.

Santos, O.C., Uria-Rivas, R., Rodriguez-Sanchez, M., G. Boticario, J., 2016. An open sensing and acting platform for context-aware affective support in ambient intelligent educational settings. IEEE Sens. J. 16, 1. http://dx.doi.org/10.1109/JSEN.2016.2533266.

Sotsenko, A., Jansen, M., Milrad, M., Using a rich context model for a news recommender system for mobile users, in: CEUR Workshop Proceedings, Vol. 1181, 2014, pp. 13–16.

Suchojad, D., 2013. Zato—Agile ESB, SOA, REST and cloud integrations in Python. Linux J. 2013.

Symeonidis, P., Chairistanidis, S., 2017. CheckInShop.Eu: A sensor-based recommender system for micro-location marketing. In: Proceedings of the Eleventh ACM Conference on Recommender Systems. RecSys '17, Association for Computing Machinery, New York, NY, USA, pp. 351–352. http://dx.doi.org/10.1145/3109859.3109977.

Wang, X., Rosenblum, D., Wang, Y., 2012. Context-aware mobile music recommendation for daily activities. In: Proceedings of the 20th ACM International Conference on Multimedia. MM '12, Association for Computing Machinery, New York, NY, USA, pp. 99–108. http://dx.doi.org/10.1145/2393347.2393368.