Information Sciences 642 (2023) 119227

Contents lists available at ScienceDirect

Information Sciences

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

Users' photos of items can reveal their tastes in a recommender system

Pablo Pérez-Núñez, Jorge Díez *, Oscar Luaces, Beatriz Remeseiro, Antonio Bahamonde

Artificial Intelligence Center, Universidad de Oviedo, Campus de Gijón, 33204, Gijón, Spain

A B S T R A C T

Recommender Systems (RS) are based on the generalization of the observed interactions of a population of users with a collection of items. Collaborative Filters (CF) give good results, but they degrade when there are few interactions to learn from. The alternative would be to observe some features of the users that could be linked to their tastes. However, specific information on users or items is often not available. In this research work, we explore how to exploit the photos of items taken by users. Our aim is to assign similar meanings to the photos of items with which the same group of users interacted. For this purpose, we define a multi-label classification task from images to sets of users. The classifier uses a general-purpose convolutional neural network to extract the basic visual features, followed by additional layers necessary to accomplish the learning task. To evaluate our proposal we compared it with CFs, using two tourism datasets that include: restaurants of six cities and points of interest of three locations. According to the experimentation carried out, the poor results achieved by CFs are outperformed by our proposal, which takes into account the visual and taste semantics of the available photos.

1. Introduction

The datasets that constitute a learning task for a Recommender System (RS) include references to users, items, and a relationship that must be learned to extend. Usually, this relationship is an evaluation (implicit or explicit) of the users towards the items. So the idea is to suggest an item i to a user u as long as other users with similar tastes to u also like i. This is the approach of the so-called *collaborative filters* (CF). To implement this idea, it is necessary to have a sufficient number of matches of user-item interactions. In practice, this method cannot be used with low densities of the user-item interaction matrix.

On the other hand, in general, there is no other information about users or items beyond their interaction. It is uncommon to have user data such as age, gender, profession, or a systematic description of their tastes. For this reason, CF somehow learn (latent) features from users to build an RS. This implies that it is necessary to have extensive information on the behavior of each user to make non-trivial recommendations. This problem, known as *cold start*, is another challenge when the number of interactions from each user is small.

In this paper we deal with a problematic situation like the one described above. The density of user-item interactions is very low (on average, each user had contact with only 1.9 items) and, additionally, we do not have specific features of either users or items. However, people use social networks to try to find out if some items may or may not be to their liking. We do it because we have available a very special relationship between users and items: the photos of some items that some users take and share.

The kernel of our proposal is a mapping function (semantics) from a set of photos to a Euclidean space. Let us recall that *semantics* studies the meaning of the expressions that we use to communicate. While these expressions are typically linguistic, the human

* Corresponding author.
E-mail address: jdiez@uniovi.es (J. Díez).

https://doi.org/10.1016/j.ins.2023.119227
Received 23 October 2020; Received in revised form 11 March 2022; Accepted 22 May 2023

Available online 26 May 2023

Fig. 1. Two images of pizza that elicit different reactions from users.

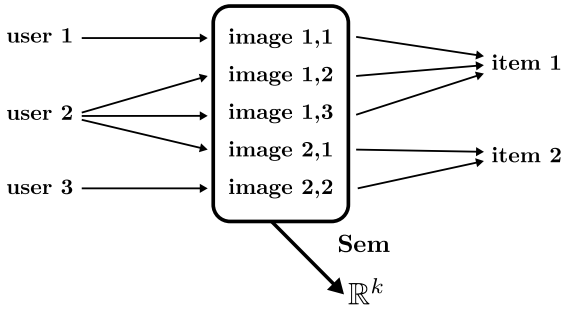

Fig. 2. The general scenario of the data used in this paper.

experience also relies on signs beyond words such as visual stimuli. In this paper, we will see how images can be given meaning in the context of an RS. In particular, a set of items' photos shared by users will be analyzed. Notice that photos contain very valuable information both about users and the items they interact with, as people only take photos of what is relevant to them. Our goal is to understand what these photos mean and what they can reveal about users' tastes.

This photo mapping function is what we refer to as *semantics*, but we could simply call it embedding or projection. The key point is that this semantic function must translate photos into vectors in a useful way. Images are usually encoded by the convolutional base of a general-purpose convolutional neural network (CNN), typically pre-trained on ImageNet [7]. In this manner, the meaning of an image is linked to the visual features that allow to detect its contents. Unfortunately, this is not very useful for taking into account users' tastes. To illustrate this problem, let us consider the two photos of pizzas depicted in Fig. 1. Based on their content, these photos are similar; however, they do not provoke similar reactions from users. Therefore, they should not have similar semantics (i.e., they will not be *synonymous* or near neighbors in the map). In other words, the user reactions go beyond the labels included in the photos themselves; in fact, we do not have ingredient labels of the photos because they are not necessary for the problem at hand.

From a formal point of view, the aim of this paper is to show how to learn our semantic function, *Sem* (see Fig. 2). The goal is to grasp the following idea: the photos of items that were interacted by the same group of users should have the same semantics. For this purpose, we try to assign a subset of users to each photo, those who have interacted with it. Notice that *Sem* uses a general-purpose CNN to encode the photos of items. However, the novelty here does not lie in the encoding itself, but in the use of this representation to define photo mappings through a multi-label learning task.

The rest of the manuscript is organized as follows. Section 2 includes an overview of some relevant publications related to the topics covered here. Section 3 describes in depth the way we understand semantic mapping in this context. Next, in Section 4, we formally introduce the approach proposed in this paper; that is, how to learn the function *Sem*, which was implemented with the network architecture described in Section 4.1. The experiments carried out are described in Section 5, along with two tourism datasets that include: restaurants of six cities and points of interest (POIs) of three locations. The experimental study includes a comparison of our method with a collaborative filter and a straightforward image encoding provided by a general-purpose CNN. The results show that the proposed mapping method is necessary to define the desired semantics. Section 6 closes the manuscript with the main conclusions.

2. Related work

When we face problems with images as input data, it is very common to use pre-trained CNNs as generic feature extractors [23], obtaining the so-called *deep features*. Unlike traditional hand-crafted features that represent basic image properties, deep features represent semantic information for a given learning task (e.g., semantic similarity in image classification refers to the target class). In the context of RS, the concept of image semantics must be defined for the problem at hand. This is the case of the work presented by Guo et al. [13], who define the semantics of an image through all the objects within it, against most approaches that use a single

object. For this purpose, the authors proposed two attention networks that combine the feature embeddings of all the fine-grained image objects to provide a recommendation.

There are other interesting works in the recent literature that take advantage of visual data to provide recommendations. He and McAuley [15] proposed a visual Bayesian personalized ranking, which uses the deep features extracted from product images by means of a pre-trained CNN. This work was improved by Kang et al. [19], who proposed to learn the image embedding along with the RS. More recently, Neve and McConville [24] proposed a method for reciprocal RS, which are those based on social platforms that connect people with people. The method consists of a Siamese CNN, with an ad-hoc architecture, trained to identify images that fit users' preferences. As different convolutional backbones can be used as image feature extractors as part of a visual-based RS, Deldjoo et al. [5] analyzed three popular CNNs along with four visual-based RS. The experimental results proved that a deeper CNN ensures high recommendation performance.

Despite the common adoption of CNNs to represent visual data in RS tasks, Ferwerda and Tkalcic [11] suggested to use traditional visual features, such as hue and saturation. These visual features were combined with content features in order to predict the users' personality in the social network Instagram. Kawattikul [20] also opted to use traditional features rather than the deep features extracted by CNNs. The proposed method uses a simple weighting technique to combine both images and text descriptions. In particular, a shape-based representation is extracted from product images and a LSTM representation is obtained from product descriptions.

Focused on other characteristics of RS, Dominguez et al. [8] performed a comparative study about the impact of several algorithms on relevant aspects such as explainability and users' trust. In particular, they compared black algorithms, such as the deep neural networks, with other transparent but less reliable methods. For their part, Díez et al. [10] proposed a framework to estimate the authorship probability of photos, which can be used to visually explain the recommendations of an RS.

In the context of the hospitality sector, we can highlight the approach available on TripAdvisor, the popular platform in which the most appealing pictures are selected to be shown when looking for restaurants and hotels [2]. The method is based on preference learning and uses the convolutional base of a ResNet50 [14]. Standard embeddings computed by CNNs are also used in [4] to provide restaurant recommendations based on visual features. Yang et al. [28] designed their own CNN to learn the similarity distance metric between food images. Their food preference learning approach can be used as part of a restaurant RS.

Smart tourism destination is the topic considered by Figueredo et al. [12], who proposed a solution capable of detecting tourist preferences using images from social media networks along with CNNs and fuzzy logic. Sertkan et al. [26] presented an approach based on fine-tuned CNNs to represent travel behavioral patterns. As a result, they are able to determine a tourist profile from a collection of user's pictures. More approaches related to travel RS can be found in the review provided by Chaudhari and Thakkar [3].

To delve further into the field, we refer the interested reader to a recent survey focused on recommender systems that take advantage of multimedia data in different domains, such as tourism or food [6].

As can be seen, in previous works the usage of semantics in RS is characterized by the use or learning of traditional semantic representations, only based on content or visual features, which is not a bad strategy since the content of an image partly represents the user's tastes or preferences. Another point in common between those works is the creation of user profiles that summarize the system's knowledge of them. The quality of these profiles increases as the number of user interactions increases; however, in cold start situations these systems may not have enough information to have reliable profiles.

The semantics that we present in this article goes beyond what has been seen in the works we have just discussed, since our semantics will take into account both the visual content of the images and the interactions that users have had with the items, which is a novelty. In addition, the semantics defined will allow us to work in cold-start scenarios where no previous user information is available because it can work with just a photo.

3. Semantics

A common strategy for building an RS is to project both users and items into a common Euclidean space $\mathbb{R}^k$. These projections, embeddings or mappings, which must be learned as functions, require an initial vector representation for both items and users. When no representation is available, artificial encodings are used. In particular, one-hot codification is one of the most prevalent artificial encodings due to its simplicity. Let us recall that one-hot codification for an object of index i (in the set of users or items) is represented by a binary vector where all the components are zero except the one with index i, which is 1.

The situation described above is found in matrix factorization methods, which are a class of collaborative filters. In this case, the RS is constructed by estimating the affinity between users and items as the inner product between the projections of their one-hot representation in a vector space $\mathbb{R}^k$. The main disadvantage of this type of method is that it is not possible to make recommendations for users or on items that have not been included in the learning process: they do not have an initial one-hot representation and, therefore, they do not have a projection in $\mathbb{R}^k$.

The alternative to this approach is to have an *external* vector representation for both users and items. This is the case of *content-based* RS, which commonly use not only the features of the items, but also a representative profile of the users that can be built with enough user data.

In contrast to these classical approaches, we propose a method that: (1) does not depend on any initial artificial encoding, being able to work in cold-start situations; and (2) does not require item features or user profiles. More specifically, we assume that, for each user, we have a small set of photos that capture their tastes. In this scenario, we can calculate the semantic mapping function of their photos. As a result, whenever we intend to summarize the user tastes, we could use the centroid in $\mathbb{R}^k$ of their photos codified by this mapping function.

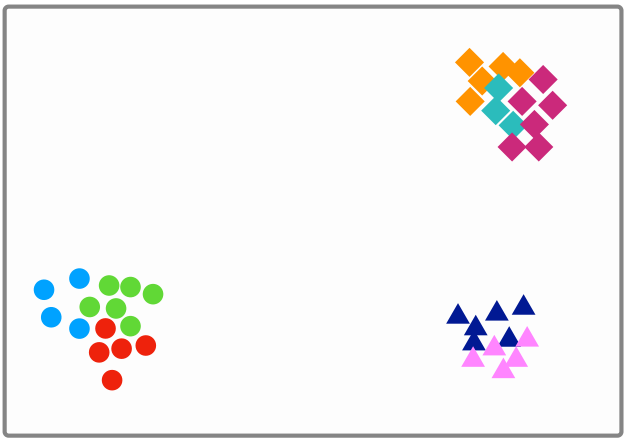

Fig. 3. Example of an ideal semantics of photographs expressed as two-dimensional vectors. Notice that the points of the same color and shape correspond to the semantics of photos of the same item.

$M =$

	item 1			**item** 2			**item** n
	$\mathbf{img}_{1,1}$	$\mathbf{img}_{1,2}$	$\mathbf{img}_{1,3}$	$\mathbf{img}_{2,1}$	$\mathbf{img}_{2,2}$	...	$\mathbf{img}_{n,1}$
user 1	1	1	1	0	0	...	1
user 2	0	0	0	1	1	...	1
...	...	...	...	...	...	...	...
user m	1	1	1	0	0	...	0

Fig. 4. Matrix $\boldsymbol{M}$ with the relationships between users, items, and images, provided $m = |\mathcal{U}|$ and $n = |Items|$.

On the other hand, for each item, we have a set of points in $\mathbb{R}^k$ formed by the mapping of the photos that users took of it. Correspondingly, we intend for the Euclidean representations of these photos to be close. Therefore, the core point of this research is the definition of a semantic mapping function that makes everything work correctly.

As stated in Section 1, the fundamental idea of this article is that the photos of items with which the same group of users interacted should have the same semantics. This principle leads us to the following:

- The photos of the same item should have the same or very similar semantics, regardless of their content.
- The photos of two items, A and B, should have similar semantics when almost all users who interacted with A also interacted with B.

In practical terms, we aim at having a semantics of photos like those shown in Fig. 3. As can be observed, there is a group of photos of three items, represented by circles (bottom left), with very similar semantics (i.e., the same group of users interacted with these three items). Another group of users seems to have interacted with other three items, represented by squares (top right). Finally, there is another group of users who interacted with two items, represented by triangles (bottom right). Summarizing, we assume that similar photos are the ones that are close using the Euclidean distance. However, we will see that this definition of similarity is not necessarily the best.

Next section describes our proposed method to learn a semantic mapping function and a similarity measure for semantics. Finally, notice that the nature of the interaction between users and items has not been detailed. This concept is discussed in Section 5.

4. Formal framework

Let us consider a set of users $\mathcal{U}$, a set of *Items*, and a set of images $\mathcal{I}$, photos that users took from items. We register their relationships in a matrix $\boldsymbol{M}$ with binary components with one row per user and one column per image (see Fig. 4). We assume that $\boldsymbol{M}(u,i) = 1$ whenever user u had an interaction with the item photographed (by u or by another user) in image i. In all other cases, $\boldsymbol{M}(u,i) = 0$.

Notice that the columns $\boldsymbol{M}(\cdot,i)$ are equal for all the images of the same item. Therefore, we can encode items by groups of equal columns of $\boldsymbol{M}$. On the other hand, these columns can also be understood as a set of users.

From the point of view of an RS, two items A and B with equal (or almost equal) column representations, are items with the same (or almost the same) interactions with users. Thus, A and B would play the same role in a recommendation environment.

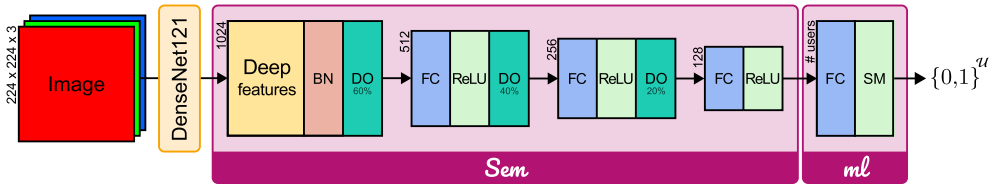

Fig. 5. *Sem* architecture.

Having all this into account, we define

$$h : \mathcal{I} \longrightarrow \{0,1\}^{\mathcal{U}},\ i \longmapsto h(i) = \boldsymbol{M}(\cdot, i). \tag{1}$$

That is, for each image i, $h(i)$ is an encoding of the item photographed in i. We would like to emphasize that

$$\langle h(i), h(j)\rangle = \langle \boldsymbol{M}(\cdot, i), \boldsymbol{M}(\cdot, j)\rangle = |\boldsymbol{M}(\cdot, i) \cap \boldsymbol{M}(\cdot, j)|. \tag{2}$$

In other words, the inner product of the projections of the images i and j is the number of common users who interacted with the items of these two images.

This is exactly the idea that we want to formalize when defining a semantic (mapping) function. To make it operative, we factorize (using a deep network presented in Section 4.1) the function h into a function from images to a Euclidean space (the continuous part) followed by a linear multi-label classifier; that is, a binary classifier for each component indexed by a user. In this way, we arrive to the key definition of this paper.

The *Semantics of Images* (*Sem*) is given by a Euclidean mapping from which h can be obtained using a multi-label classifier (*ml*). In symbols,

$$h : \mathcal{I} \xrightarrow{Sem} \mathbb{R}^k \xrightarrow{ml} \{0,1\}^{\mathcal{U}}. \tag{3}$$

The *semantics* of an image i is defined by $Sem(i)$ and, according to equation (2), the *similarity* is defined by the inner product.

It should be noted that, as the number of users available in some recommendation scenarios can be large, in order to simplify the multi-label classifier complexity, we are going to select a percentage of the *top active* users (those with more reviews available in the training set) to act as target labels in the learning task.

4.1. Network architecture

This section presents the network employed to solve the multi-label learning task previously described, see Fig. 5.

Sem receives as input an image i, which is first encoded by means of a densely connected CNN (DenseNet) [17]. In particular, we used the convolutional base of a DenseNet-121 pre-trained on ImageNet,[1] thus obtaining a general-purpose 1024-dimensional embedding. Once the input image is encoded, we pass it through a batch normalization layer (BN) [18] in order to improve the speed and stability of the learning process.

Then, the output of the BN is passed through three processing blocks of sizes 512, 256, and 128. Each block is composed of a hidden fully connected layer (FC) and a rectified linear unit (ReLU) [22] as activation function to achieve the non-linearity. Notice that the output of the BN layer and the first two processing blocks are followed by a dropout layer (DO) [27] in order to reduce the amount of overfitting.

Finally, to solve the multi-label problem that we are facing, the output block of our network is an FC layer, with size $|\mathcal{U}|$, followed by a sigmoid (σ) activation function to achieve a probability value for each user $u \in \mathcal{U}$.

The model was trained by minimizing the binary cross-entropy loss, using the Adam optimizer [21]. With the aim of increasing the performance, we added weights to the loss function making five times more relevant to fail a 1 than a 0. This makes the model more accurate in predicting the correct users in each image and it will also have more leeway to predict other users that it also considers relevant.

By removing the last two layers of the trained model, we can take the 128-feature vector ($k = 128$ in equation (3)) obtained in the third block to represent the semantics captured from input images. These are the so-called *Sem* features.

The entire process described in this section is reflected in Fig. 6, in which all the images of the same item will have the same output (users who interacted with it) in the multi-label classification problem.

5. Case study: restaurants and POIs

The adequacy of the approach presented in this paper was assessed through several datasets from different domains. In each domain, two kinds of experiments were performed in order to, first compare our system with a traditional collaborative filter, and

[1] https://keras.io/api/applications/densenet/#densenet121-function.

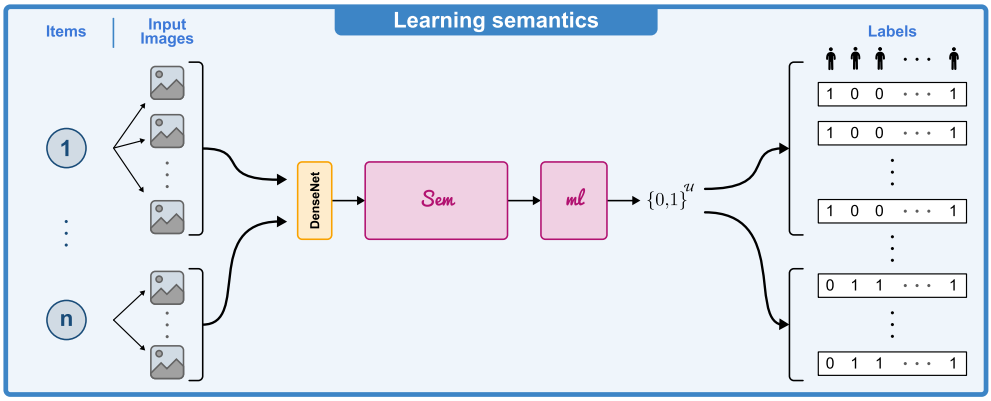

Fig. 6. Illustration of the formal framework to learn the semantics of the input images, which in this example are the photos of a set of items. Each input image is processed to obtain a binary vector of dimension the number of users, with 1 in the positions corresponding to the users who interacted with the item and 0 otherwise.

Table 1
Statistics of the two datasets and six cities used in the experiments, including their population in millions of inhabitants.

Dataset		Population	Total figures per dataset			
		(millions)	Reviews	Images	Users	Items
Restaurants	Gijón	0.3	6871	14964	4382	553
	Barcelona	1.6	52889	112083	26925	5315
	Madrid	2.1	69561	149004	34998	6042
	NYC	8.3	80188	138673	45662	6675
	Paris	8.9	96500	183574	50345	11019
	London	3.2	175542	315546	103835	12574
POIs	Barcelona	1.6	42330	109760	23192	674
	NYC	8.3	74471	180837	36693	808
	London	8.9	78245	190989	42587	1690

second to analyze the performance of our system recommending items to previously unseen users. The latter was tackled to check the answer to the *cold start* problem, where the systems should try to recommend items to users without any previous information.

With the aim of performing these two experiments, we need to split each dataset into training/dev/test partitions keeping in mind that, in the first experiment (collaborative filter) all users and items must be in the training set. However, the second one (cold start) requires all restaurants in training, but not the users, who must be distributed among the partitions keeping a group of unseen ones for the test set. The reason behind these restrictions is explained in Sections 5.5 and 5.6.

5.1. Restaurant reviews

The first domain focuses on restaurants (items), customers (users) and photographs of dishes taken by them.[2] Each customer on the set uploaded one or more *reviews*, each in a different restaurant, including some photographs of their experience (notice that we only consider the reviews with pictures). Since our objective is to acquire the semantics of images in terms of users' tastes, we filtered out all the images whose main content is not food. In this case, we interpret that users had a positive interaction with the restaurants they visited. We could have used another criterion but, to defend this point of view, we can see recommendations as suggestions of restaurants that should be visited.

The restaurants section of Table 1 shows the main characteristics of these datasets without performing any partitioning. Reviews are from restaurants in six quite different cities. In fact, the size of cities ranges from the smallest one, with 0.3 million inhabitants, up to the most populated one with almost 9 million inhabitants.

5.2. Points of interest

The second domain focuses on points of interest (POIs) in cities. In this domain, visitors (users) go to one or more POIs (items) and take pictures of them. As in the restaurants case, we consider that users had a positive interaction with the items they visited. The POI rows of Table 1 show the statistics of the three cities available for this domain.[3]

[2] The datasets for the restaurant domain are available for download at [25].
[3] The datasets for the POI domain are available for download at [1].

Table 2
Multiple simulated examples, for a user i, of visited restaurants (y_i), Top-3 of restaurants that a system could predict as most related to this user ($\hat{y}_i$) and the behavior of Top-3 accuracy, Prec@3 and Rec@3 metrics in those situations.

y_i	$\hat{y}_i^3$	$\lvert y_i \cap \hat{y}_i^3 \rvert$	$\lvert y_i \rvert$	Top-3 accuracy	Prec@3	Rec@3
r_{71}, r_{43}	r_{34}, r_{54}, r_{129}	0	2	0	0	0
r_{311}, r_3, r_{198}	r_{92}, r_{198}, r_{311}	2	3	1	0.67	0.67
r_{203}	r_{48}, r_{203}, r_7	1	1	1	0.33	1
r_{57}, r_{32}	r_{286}, r_{30}, r_{37}	0	2	0	0	0
r_{47}, r_8, r_{93}	r_{143}, r_1, r_{93}	1	3	1	0.33	0.33
r_{81}, r_{32}	r_{81}, r_{327}, r_{32}	2	2	1	0.67	1
$r_{27}, r_9, r_{111}, r_{41}, r_8$	r_{41}, r_9, r_{111}	3	5	1	1	0.6

Although we might think that all the photos of the same POI are the same or very similar, not all visitors take the same photographs because their interests are not always the same. For example, when someone visits a cathedral they may be more interested in the exterior architecture of the building, the interior stained glass windows, the holy images, or the frescoes painted on it. Therefore, the perception of the same POI by several visitors may be different. It is also worth noting that all the photographs taken in a POI, in most cases, can be unambiguously related with only one item, whereas in the restaurant domain, very similar food pictures can be taken in more than one item. That means that you will not find a picture of the Eiffel Tower on another POI, but you will find almost the same picture of pizza in multiple restaurants.

5.3. Evaluation method

Once the semantic (mapping) function is defined (Section 4), we must devise a method to evaluate it. For this purpose, let us assume that a test user shows a small set of photos capturing their preferences. Then, we calculate the semantics of each photo, which presumably should be similar. Next, we compute the centroid in $\mathbb{R}^k$ with the intention of summarizing the preferences of the user. If the semantics is correctly estimated, in the restaurant domain, for example, the most *similar* photo to the centroid should be one taken in a restaurant that the user would like to visit. Notice that, in this semantic context, the most similar will be the one with the greatest dot product with the centroid, see equation (2). This procedure will yield a list of items ordered from highest to lowest affinity. Then, if the first item of the list is one the user has actually interacted with, we count a hit (Fig. 7); doing this for all the test users will return us the Top 1 accuracy. Following the same procedure, but with the first five items of the list, we will obtain the Top 5 accuracy and so on, applying the formula

$$\text{Top-N accuracy}(Y, \hat{Y}) = \frac{1}{m} \sum_{i}^{m} 1_{y_i \cap \hat{y}_i^N \neq \emptyset}, \tag{4}$$

where Y and $\hat{Y}$ are the sets of visited and recommended items, respectively, for all the m users of the dataset, y_i is the set of visited items for user i, $\hat{y}_i^N$ refers to the Top-N recommended items for user i, and, finally, 1_p will be 1 when the predicate p is true and 0 otherwise.

We will also calculate the Precision and Recall measures in order to further analyze the quality of the recommendation of these rankings using the following formulas:

$$\text{Prec@N}(Y, \hat{Y}) = \frac{1}{m} \sum_{i}^{m} \frac{\lvert y_i \cap \hat{y}_i^N \rvert}{\lvert \hat{y}_i^N \rvert} = \frac{1}{m} \sum_{i}^{m} \frac{\lvert y_i \cap \hat{y}_i^N \rvert}{\lvert N \rvert} \tag{5}$$

$$\text{Rec@N}(Y, \hat{Y}) = \frac{1}{m} \sum_{i}^{m} \frac{\lvert y_i \cap \hat{y}_i^N \rvert}{\lvert y_i \rvert} \tag{6}$$

Table 2 shows, for a user i, multiple simulated examples of visited restaurants (y_i), Top-3 of restaurants that the system could predict as most related to this user ($\hat{y}_i$) and the behavior of Top-3 accuracy, Prec@3 and Rec@3 metrics in those situations.

- Prec@N shows the proportion of recommended restaurants which are relevant for the user, that is, if 3 restaurants are recommended to a user, the only values that can be obtained are 0, 0.33, 0.66, or 1, depending, respectively, on whether the user actually visited 0, 1, 2, or 3 of the recommended restaurants. However, the vast majority of users have written reviews in only 1 or 2 restaurants. Therefore, even if the system outputs the best recommendation possible, in some cases would be impossible to obtain a perfect result for this metric, as it happens, for example, in the third and sixth case of Table 2.
- Rec@N computes the proportion of relevant restaurants which have been predicted. If 3 restaurants are recommended for a user who actually likes 5, the values that can be obtained are 0, 0.2, 0.4, or 0.6, depending on whether the user actually visited 0, 1, 2, or 3 of them. In other words, even if the best possible recommendation is given, it is not possible to reach the maximum in Prec@3 as can be seen in the last example of Table 2.
- Top-N accuracy is a metric independent of the number of restaurant predictions and the number of visited restaurants by the users, so for each case a 1 will be obtained if any of the restaurants visited for the user is among the recommended ones and a 0 in the opposite case.

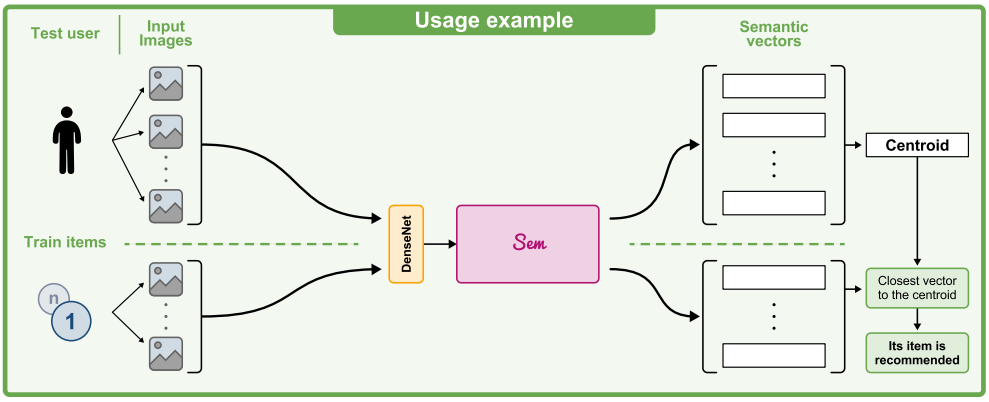

Fig. 7. Illustration of a usage example of the validation procedure, in which the semantic vectors of the images of a new user are computed with the network trained in Fig. 6. The centroid of these semantic vectors is calculated and compared to the semantic vectors of the photos considered during the training phase. The closest photo to the user's centroid corresponds to the item that will be recommended.

Thus, each of these measures will have a more appropriate scope of application: i) Prec@N and Rec@N are more suitable for comparing the performance of different recommender systems with each other, since the particular characteristics of each dataset affect all the systems equally, and ii) Top-N accuracy is more suitable for assessing the quality of recommendations in a given dataset.

Let us recall that the semantics learned does not have to simply focus on the content of the photo; convolutional neural networks do that very well, as explained in Section 1. In our case, the semantics learned goes further, since it not only takes into account the content of the photographs, but also the set of restaurants or points of interest that a user visited. For example, in the points of interest domain, there are many tourists visiting Notre Dame Cathedral and the Eiffel Tower when they go to Paris. If we have a system that obtains semantics based only on the content of the photographs (such as a CNN), it will not be able to recommend the Eiffel Tower to a user who has visited the Notre Dame Cathedral, since their photographs are not similar. However, our system will be able to do so because the semantics it learns also takes into account the places visited by the users.

5.4. Experimental settings

Once the case study and the evaluation have been defined, we can start with the experimentation. As said at the beginning of the section, we want to perform two experiments: i) comparing our system with a traditional collaborative filter and ii) analyzing the performance in a cold start scenario. In both experiments we are going to test, not only the embedding obtained by our system, but also that produced by the pre-trained convolutional base of a DenseNet-121 (used as input in our model). This will allow us to test whether the semantics we are trying to learn actually performs better than traditional content-based encoding. To do so, we only need to remove our system (*Sem*) from the evaluation procedure showed in Fig. 7, using directly the encoding of the DenseNet as semantic vectors. To find the closest vector to the centroid, in this case, we need to change the dot product for the Euclidean distance due to the nature of the ImageNet problem that this type of pre-trained CNNs usually solve.

The first experiment also requires the creation of a collaborative filtering system to compare with. For this purpose, we have created a model based on the projection of users and items in a new latent space in which the probability of interaction between them is computed. This model will receive users and items encoded as one-hot vectors and, by calculating embeddings, it projects them in a common 64-dimensional space. It then concatenates both vectors, which are fully connected to another 64-dimensional layer (using a ReLU activation function). This layer is then connected to a single cell output-layer that predicts the probability of interaction between users and items by applying a sigmoid function. This particular method of solving collaborative filtering tasks is the so called *Neural network-based Collaborative Filtering* [16].

We cannot use the same evaluation procedure explained above since we do not have images so, in order to obtain the affinity sorted list of items, we will use the CF model to predict, for each test user, the probability of interaction for all the items. The rest of the evaluation procedure stays intact.

It is worth noting that we use the same experimental procedure for all the systems we are going to compare (*Sem*, DenseNet, and a collaborative filter). First, the dataset (restaurants or POIs) is divided into training, dev, and test partitions as appropriate (following the constraints of each of the two experiments). Then, using the training and dev sets of Barcelona (as it is a medium-sized city), we performed a grid-search of hyperparameters and architectures for each system and in both experiments, in order to find the optimal combination for the problem to be solved. Finally, with the best combination for each system, we trained all the systems joining the training and dev subsets. The test subset was not seen in the training procedure and will only be used to perform the final evaluation described in the preceding section.

5.5. Experiment 1: comparison against a collaborative filter

In this first experiment we want to evaluate the performance of the semantics learned by our *Sem*, following equation (3), against a traditional recommender system such as a collaborative filter. We also incorporate to the comparison the embedding generated by

Table 3
Dataset division in training/dev/test for the first experiment. Notice that Training + Dev contains all the users and items of the entire dataset (Table 1) due to the nature of the experiment. Density represents the percentage of 1s in the user-item interaction matrix.

Dataset		Training + Dev		Test
		Reviews	Density	Reviews
Restaurants	Gijón	6341	0.26	530
	Barcelona	47403	0.03	5486
	Madrid	62353	0.03	7208
	NYC	72725	0.02	7463
	Paris	87014	0.02	9486
	London	160288	0.01	15254
POIs	Barcelona	38009	0.24	4321
	NYC	65345	0.22	9126
	London	70425	0.10	7820

Table 4
Top N accuracy results (in percentage) between the Collaborative Filter, and the embeddings learned by DenseNet and *Sem*. Best results are in bold.

Dataset		Top 1			Top 5			Top 10		
		CF	DNet	*Sem*	CF	DNet	*Sem*	CF	DNet	*Sem*
Restaurants	Gijón	7.79	12.31	**24.12**	12.56	26.13	**33.42**	15.58	37.94	**41.46**
	Barcelona	1.93	7.80	**14.43**	4.90	15.99	**24.49**	6.70	20.44	**30.73**
	Madrid	2.41	10.62	**12.38**	5.00	20.21	**21.59**	6.71	26.31	**27.10**
	New York	4.57	15.76	**16.56**	7.92	**28.00**	**28.00**	10.03	**34.14**	33.11
	Paris	0.21	7.46	**9.98**	2.55	15.28	**20.40**	4.46	19.68	**26.25**
	London	0.69	7.55	**10.77**	3.64	16.84	**23.90**	6.29	21.99	**29.44**
POIs	Barcelona	62.49	80.89	**81.83**	74.00	**94.06**	91.52	86.55	**96.66**	93.95
	New York	7.28	69.61	**88.28**	76.41	91.80	**93.30**	86.85	**95.59**	95.01
	London	14.99	59.56	**70.32**	46.82	**81.96**	81.79	57.72	**88.04**	84.72

a convolutional neural network in order to show that taking into account only the content of the photographs yields worse results. Recall that the semantics learned by our system considers not only the content of the photographs, but also the places visited by the users. In this comparison, the embeddings generated by the DenseNet network will be used to generate recommendations in the same way as those generated by *Sem* (see Fig. 7).

To carry out this first experiment, the datasets were separated into training and test sets, ensuring that all users and items were present in the training set. This requisite is essential for the operation of collaborative filters, although it is not necessary for our system. To perform this division, the reviews of users who have interacted with just one or two items are forced to belong to the training set (this is necessary since the training set will be further subdivided to obtain a validation set); for the remaining users, those who have reviewed three or more items, their interactions are divided between the training and test sets. The characteristics of the resulting subsets obtained after this split are shown in Table 3. As can be observed, the number of reviews used as test is approximately 10% of the total number of reviews. If we focus on the densities of the user-item matrix, in the restaurant domain the densities are extremely low, between 0.01% and 0.03% (with the exception of Gijón, a city that is also very different in size with respect to the rest of the cities), while in the points of interest domain the densities vary between 0.10% and 0.24%.

Top N accuracy results obtained for this experiment can be seen in Table 4. If we focus on the accuracy when recommending a single item (Top 1) we see that *Sem*, the system presented in this paper, obtains the best performance on all datasets in both domains. In contrast, the collaborative filter obtains the worst results, which is not a surprise since the user-item matrix hardly presents any interactions. The performance of the collaborative filter in the POI domain for Barcelona may be striking, since it obtains a 62.49% when in the rest of the cases it is barely close to 10%. Analyzing this result, we have observed that this high percentage is due to the fact that, unlike the rest of the cities, there is a POI (the Sagrada Familia) that has a much higher number of reviews than the rest of the POIs of that city, which facilitates the success in the recommendation.

If we ask our system to give us a list with five recommendations (Top 5), it achieves the best result in all the cities in the restaurant domain. Regarding the POI domain, there is a remarkable improvement when using the DenseNet embeddings in the city of Barcelona, while *Sem* is the best in New York and both systems obtain similar results in London. There is also a considerable improvement in the performance of CF when using POIs, although with considerably worse results than the other two systems (DenseNet and *Sem*). We believe that this improvement is due to the fact that the POI problem is a bit simpler for several reasons: 1) the number of POIs is smaller than the number of restaurants in the same city (in Table 1, it can be seen that cities have approximately 10 times more restaurants than POIs); 2) the density of the user-item matrix in the POI domain is higher than in the restaurant domain, so there are more data available during the training stage; and 3) when a tourist visits a new city, there are certain points of interest that are almost always visited.

Table 5
Precision results (in percentage) between the Collaborative Filter, and the embeddings learned by DenseNet and *Sem*. Best results are in bold.

Dataset		Prec@1			Prec@5			Prec@10		
		CF	DNet	*Sem*	CF	DNet	*Sem*	CF	DNet	*Sem*
Restaurants	Gijón	7.79	12.31	**24.12**	2.56	5.23	**6.73**	1.66	3.84	**4.27**
	Barcelona	1.93	7.80	**14.43**	0.98	3.20	**4.94**	0.67	2.05	**3.12**
	Madrid	2.41	10.62	**12.38**	1.00	4.04	**4.33**	0.68	2.63	**2.73**
	New York	4.57	15.76	**16.56**	1.59	5.60	**5.61**	1.01	**3.42**	3.32
	Paris	0.21	7.46	**9.98**	0.51	3.06	**4.10**	0.45	1.97	**2.65**
	London	0.69	7.55	**10.77**	0.73	3.37	**4.79**	0.63	2.20	**2.96**
POIs	Barcelona	62.49	80.89	**81.83**	14.83	**19.19**	18.55	8.76	**10.07**	9.62
	New York	7.28	69.61	**88.28**	15.33	18.97	**19.22**	8.87	**10.15**	10.01
	London	14.99	59.56	**70.32**	9.40	16.67	**16.75**	5.83	**9.10**	8.78

Table 6
Recall results (in percentage) between the Collaborative Filter, and the embeddings learned by DenseNet and *Sem*. Best results are in bold.

Dataset		Rec@1			Rec@5			Rec@10		
		CF	DNet	*Sem*	CF	DNet	*Sem*	CF	DNet	*Sem*
Restaurants	Gijón	7.47	11.89	**22.82**	12.02	24.21	**30.84**	14.57	34.97	**38.22**
	Barcelona	1.63	7.59	**12.76**	4.20	15.38	**21.69**	5.64	19.49	**27.20**
	Madrid	2.19	10.32	**11.43**	4.48	19.38	**19.52**	5.86	**25.02**	24.19
	New York	4.21	15.49	**15.79**	7.32	**27.31**	26.41	9.32	**33.09**	31.12
	Paris	0.15	7.29	**8.99**	2.13	14.70	**17.96**	3.77	18.77	**23.02**
	London	0.61	7.40	**9.90**	3.22	16.35	**21.84**	5.51	21.17	**26.74**
POIs	Barcelona	59.28	76.97	**77.30**	69.64	**88.47**	85.64	81.59	**91.02**	87.92
	New York	6.40	64.97	**80.79**	70.57	85.16	**85.56**	80.11	**89.06**	87.70
	London	14.32	56.61	**64.76**	43.27	**76.57**	75.42	53.12	**81.96**	78.21

If we focus on the performance when 10 items are recommended (Top 10): our system obtains the best results in all the datasets of the restaurant domain except one (New York); and in the POIs domain, DenseNet obtains the best result in Barcelona and London, while both approaches obtain a very similar result in New York.

If we want to see which representation is more effective when trying to answer the question *how many of the recommended items are relevant?*, we have to look at Table 5 showing the results in Precision. We can see in the table that the semantics learned by *Sem* is the best in the vast majority of the tests performed, so we can conclude that our semantics is more accurate in the recommendations than the other methods. The Precision values are quite high in the POI domain when only one place is recommended; this reinforces the previous statement that this domain is simpler than the restaurant domain. It is also observed that the Precision values in both domains decrease for all systems as the number of recommended items increases. This is because the average number of interactions per user is very low (around 1.9) and as the list of recommended items increases, it is inevitable that items that are not relevant are introduced in the recommendation.

If we want to answer the question *how many of the relevant items have been recommended?*, then we have to look at Table 6 where we show the results on the Recall measure. In this case we see that when only one item can be recommended, *Sem* is the one that obtains the best results. As the number of recommended items increases, the embedding obtained by DenseNet is equal in performance to that obtained by *Sem*. This happens mainly in the POI domain and we believe, as we mentioned before, that this may be due to the fact that it is a simpler domain.

In summary, *Sem* presents in general the best performance, although when a large list of recommendations is made in the simplest domain, the embeddings provided by a DenseNet are comparable and even better in some cases. What is clear after seeing these results is that the performance of the collaborative filter is quite far from the performance of our system, being significantly worse (using 0.05 as p-value) in all cases applying the Bonferroni-Dunn test [9].

5.6. Experiment 2: performance in cold start situations

In this second experiment we intend to analyze the performance of our system when asked to make recommendations for users who were not present in the training set. Collaborative filters are not able to make recommendations to such users and, for this reason, this system is not used in this experiment.

To carry out an experimental evaluation, we split the datasets into training, dev and test sets. The division was made in two steps. First, we randomly selected half of the users for the training set and the other half for the dev and test sets; and then, we moved to the training set those users in the other two sets with reviews on items that only appear in them. Thus, we guarantee that there are no references to unknown items in the dev and test sets. In Table 7 we show a description of the sets after performing the split.

Table 7
Dataset division in training/dev/test for the second experiment. The test set is showed in terms of the number of users with 1, 2, 3, or $>=4$ items reviewed. Notice the popularity relevance in the POIs dataset.

Dataset		Training + Dev			Test			
		Users	Avg it/usr	Popularity	=1	=2	=3	>=4
Restaurants	Gijón	3330	1.61	8.65	833	136	37	46
	Barcelona	20527	2.04	1.52	4683	873	336	506
	Madrid	26670	2.05	2.54	5990	1196	442	700
	NYC	34746	1.81	3.39	8015	1583	619	699
	Paris	38511	1.99	0.57	8876	1528	558	872
	London	78685	1.73	0.83	19557	3078	1035	1480
POIs	Barcelona	17411	1.83	41.96	3906	955	398	522
	NYC	27558	2.04	21.51	5802	1560	691	1082
	London	32022	1.86	14.69	7450	1623	616	876

Notice that we show the users of the test set broken down by the number of reviews they have since we will focus the analysis of the results on seeing the performance of the different algorithms based on the number of registered user interactions. We also show the percentage of users who visited the most popular item in the training set. For example, in the set of POIs of Barcelona, 41.96% of users visited the Sagrada Familia (something that, as mentioned in the previous section, affects the results obtained to a certain extent). It can be seen that the most popular items in the restaurant domain have a much lower percentage than the most popular items in the POI domain.

Fig. 8 depicts the Top N accuracy performance of the embeddings (*Sem* and DenseNet), broken down by the number of reviews of the tests users. We also include under the name *Popularity* the results obtained by recommending on the basis of the number of reviews, i.e., ordering items (restaurants or POIs) from the most to the least reviewed (popular). Provided we have already shown in the previous experiment that the three methods compared present a similar behavior in Precision and in Recall to that of Top N accuracy, in this experiment we will only show the graphs obtained in the latter measure, in order to ease the reading of the paper. Moreover, both Precision and Recall scores are not bounded as usual in these problems (e.g., it is impossible for a user interacting with less than N items to achieve a maximum Prec@N, as well as for a user interacting with more than N to achieve a maximum Rec@N), but the Top N accuracy does not present this drawback, which makes it easier to compare the performance obtained between the different datasets.

The most relevant aspect that can be appreciated in these graphs is that, for the representation obtained by *Sem*, the greater the number of user interactions (or number of reviews), the better the performance, outperforming the other two systems.

With only one interaction, it looks like the DenseNet encoding is enough to obtain a slightly better result than *Sem*, but not significantly. If we increase the number of interactions, the DenseNet encoding performance becomes worse and worse in most cases. This is because when you have a single image, for example of a pizza, if you look only at the content (as DenseNet does), it is normal to recommend pizza restaurants. The problem starts when the user has taken photos in more than one restaurant (pizza an sushi for example). In this case, when the centroid is computed using both image embeddings, the obtained vector represents an average of two contents, resulting in a bad recommendation. Our system, however, takes into account the other places the user has visited in addition to the content, making it capable of knowing that the users who ate pizza and sushi, for example, also tend to go to Mexican restaurants. This additional knowledge makes *Sem* capable of producing better recommendations as can be seen in the graphs.

Another behavior shown in the graphs is that recommending the most popular items is rarely a good option. It is true that its performance increases with the number of user interactions, but this is explained by the fact that, with the increase of interactions, the probability of visiting a popular item (restaurant or POI) also increases. In the POI domain, this method achieves some good results, being some of them at the same level as our system. This is because of the already commented peculiarities of the domain in junction with, as shown in Table 7, the immense popularity of some places that are always visited by many tourists.

In short, when we have a situation where the users provide only images of one item, the *Sem* or DenseNet encodings can be used almost indistinctly. If the user has interacted with more than one item in the past, using the embeddings from *Sem* is the best option, while the most popular item should not be recommended under any circumstances. Fig. 9 shows the significantly better performance of *Sem* against the other methods, using a Bonferroni-Dunn test. Therefore, we can conclude that *Sem* is the most competitive in cold start situations and can also benefit from the scarce information in these situations.

6. Conclusions

This paper presents a method to learn a photo mapping (embedding or projection) in $\mathbb{R}^k$. In particular, we deal with photographs taken by users on RS items. From a practical point of view, this means that we have users, items, and relationships between them. First, we have a binary relationship that we can understand as an interaction between users and items. The second relationship that we contemplate is the one given by the photos that users take of items.

The projections we are looking for must be such that, the photos of items with which the same set of users interacted, lead us to very similar vectors in $\mathbb{R}^k$. With the set of photos taken by a user we can build their profile in a RS. Furthermore, this can be done every time a new user appears because their photos lead us to assign them a place in $\mathbb{R}^k$ that represents their tastes. The same happens with items: the photos that users take of them allow defining a vector with their features. This is a key point as photos

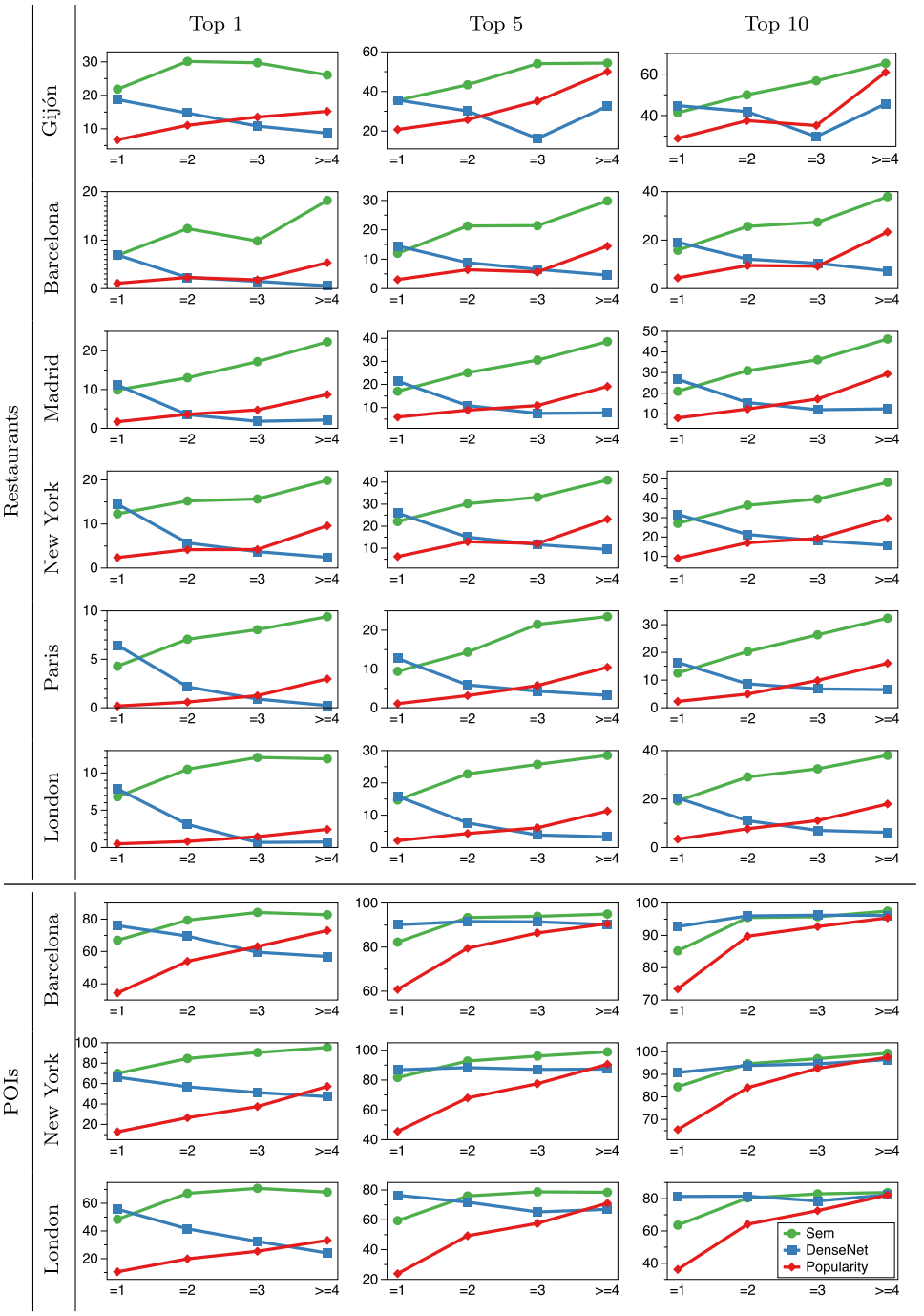

Fig. 8. Top N accuracy of the three methods analyzed in the second experiment. The horizontal axis represents the number of reviews of each user.

allow us to treat the original dataset as a *content-based* RS. Remember that we do not normally have personal information about users beyond their interactions. With no age, job, gender, or any other characteristic related to their tastes, a CF is the only possible approach to building a RS. In this way we can overcome the so-called *cold start* problem. Also, when the number of interactions is low, collaborative filters cannot provide acceptable returns, but content-based RS can.

To evaluate our proposed method, we used two tourism datasets with different domains: restaurants (six cities) and points of interest (three cities). In both cases, we have verified that the performance of the collaborative filters is dramatically low while we achieve good results with our proposal. Additionally, we compared our semantic approach against a general-purpose embedding given by a pre-trained CNN. The empirical results support our hypothesis: a CNN embedding is not enough to reflect the users' tastes. The method proposed in this paper, *Sem*, take advantage of users to learn a multi-label classification task involving their tastes.

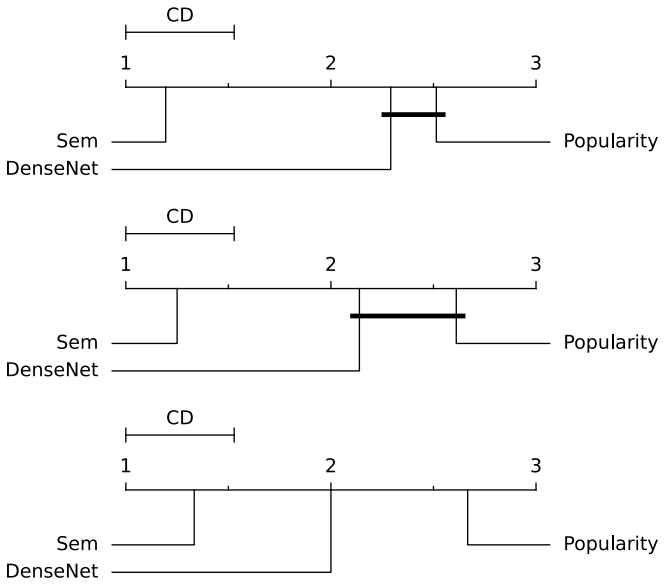

Fig. 9. Cold start Bonferroni-Dunn tests for Top 1 (up), Top 5 (middle) and Top 10 (bottom). Our system performs significantly better than the others (p-value of 0.05) with a Critical Difference (CD) of 0.53. The other two systems only differ significantly in Top 10.

CRediT authorship contribution statement

Pablo Pérez-Núñez: Conceptualization, Methodology, Software, Writing – review & editing. **Jorge Díez:** Conceptualization, Methodology, Writing – review & editing. **Oscar Luaces:** Conceptualization, Methodology, Writing – review & editing. **Beatriz Remeseiro:** Conceptualization, Methodology, Writing – review & editing. **Antonio Bahamonde:** Conceptualization, Methodology, 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.

Acknowledgements

This work was funded under grants PID2019-109238GB-C21 and TIN2015-65069-C2-2-R from the Spanish Ministry of Science and Innovation, and IDI-2018-000176 from the Principado de Asturias Regional Government, partially supported with ERDF funds. Pablo Pérez-Núñez acknowledges the support of the Principado de Asturias Regional Government under *Severo Ochoa* predoctoral program (ref. BP19-012).

We are grateful to NVIDIA Corporation for the donation of the Titan Xp GPUs used in this research.

References

[1] C. Alonso, P. Pérez-Núñez, O. Luaces, J. Díez, B. Remeseiro, A. Bahamonde, TripAdvisor points of interest, https://doi.org/10.5281/zenodo.5749846, 2021.
[2] G. Amis, Improving TripAdvisor photo selection with deep learning, https://www.tripadvisor.com/engineering/improving-tripadvisor-photo-selection-deep-learning/, 2017.
[3] K. Chaudhari, A. Thakkar, A comprehensive survey on travel recommender systems, Arch. Comput. Methods Eng. 27 (2020) 1545–1571.
[4] W.T. Chu, Y.L. Tsai, A hybrid recommendation system considering visual information for predicting favorite restaurants, World Wide Web 20 (6) (2017) 1313–1331.
[5] Y. Deldjoo, T. Di Noia, D. Malitesta, F.A. Merra, A study on the relative importance of convolutional neural networks in visually-aware recommender systems, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3961–3967.
[6] Y. Deldjoo, M. Schedl, P. Cremonesi, G. Pasi, Recommender systems leveraging multimedia content, ACM Comput. Surv. 53 (5) (2020) 1–38.
[7] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[8] V. Dominguez, P. Messina, I. Donoso-Guzmán, D. Parra, The effect of explanations and algorithmic accuracy on visual recommender systems of artistic images, in: 24th International Conference on Intelligent User Interfaces, 2019, pp. 408–416.
[9] O.J. Dunn, Multiple comparisons among means, J. Am. Stat. Assoc. 56 (293) (1961) 52–64.
[10] J. Díez, P. Pérez-Núnez, O. Luaces, B. Remeseiro, A. Bahamonde, Towards explainable personalized recommendations by learning from users' photos, Inf. Sci. 520 (2020) 416–430.
[11] B. Ferwerda, M. Tkalcic, Predicting users' personality from Instagram pictures: using visual and/or content features?, in: 26th Conference on User Modeling, Adaptation and Personalization, 2018, pp. 157–161.

[12] M. Figueredo, J. Ribeiro, N. Cacho, A. Thome, A. Cacho, F. Lopes, V. Araujo, From photos to travel itinerary: a tourism recommender system for smart tourism destination, in: IEEE 4th International Conference on Big Data Computing Service and Applications, 2018, pp. 85–92.
[13] G. Guo, Y. Meng, Y. Zhang, C. Han, Y. Li, Visual semantic image recommendation, IEEE Access 7 (2019) 33424–33433.
[14] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[15] R. He, J. McAuley, VBPR: visual Bayesian personalized ranking from implicit feedback, in: 30th AAAI Conference on Artificial Intelligence, 2016, pp. 144–150.
[16] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 173–182.
[17] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[18] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.
[19] W.C. Kang, C. Fang, Z. Wang, J. McAuley, Visually-aware fashion recommendation and design with generative image models, in: IEEE International Conference on Data Mining, 2017, pp. 207–216.
[20] K. Kawattikul, Product recommendation using image and text processing, in: International Conference on Information Technology, 2018, pp. 1–4.
[21] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: 3rd International Conference on Learning Representations, 2015, pp. 1–15.
[22] V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, in: 27th International Conference on Machine Learning, 2010, pp. 807–814.
[23] L. Nanni, S. Ghidoni, S. Brahnam, Handcrafted vs. non-handcrafted features for computer vision classification, Pattern Recognit. 71 (2017) 158–172.
[24] J. Neve, R. McConville, ImRec: learning reciprocal preferences using images, in: Fourteenth ACM Conference on Recommender Systems, 2020, pp. 170–179.
[25] P. Pérez-Núñez, O. Luaces, J. Díez, B. Remeseiro, A. Bahamonde, TripAdvisor restaurant reviews, https://doi.org/10.5281/zenodo.5644892, 2021.
[26] M. Sertkan, J. Neidhardt, H. Werthner, Eliciting touristic profiles: a user study on picture collections, in: 28th ACM Conference on User Modeling, Adaptation and Personalization, 2020, pp. 230–238.
[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[28] L. Yang, Y. Cui, F. Zhang, J.P. Pollak, S. Belongie, D. Estrin, Plateclick: bootstrapping food preferences through an adaptive visual interface, in: 24th ACM International on Conference on Information and Knowledge Management, 2015, pp. 183–192.