A Common Framework and Metric for Recommender Systems: A Proposal

Felix H. del-Olmo and **Eduardo H. Martin** and **Elena Gaudioso**
Universidad Nacional de Educacion a Distancia
C/ Juan del Rosal 16, 28040 Madrid, Spain
e-mail: {felixh,eduardoh,elena}@dia.uned.es

Abstract

Comparing recommender systems requires common means for evaluating them. Nevertheless they have been so far evaluated through many, often incompatible, ways. This problem is mainly due to the lack of a common framework for recommender systems. In this paper, we propose such a framework. Within this framework, recommender systems are applications with the following composed objective: (i) to choose *which* (of the items) are going to be shown to the user and, (ii) to decide both *when* and *how* the recommendations will be shown. Starting from this framework, a natural metric emerges. We will complete the framework proposal by studying the proporties of this metric, along with a comparison of it with the traditional ones.

Introduction

Recommender systems were originally defined as ones in which "people provide recommendations as inputs, which the system then aggregates and directs to appropiate recipients" (Resnick & Varian 1997). Nowadays, a broader and more general definition extends in this field, referring to recommender systems as those systems that "have the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options" (Burke 2002). This fact implies that current recommender systems have a clear main objective: to *guide* the user towards *useful/interesting* objects.

Although the definition (then the goal) of these systems has evolved through the years, it can hardly be affirmed the same on their metrics. From the early first systems to date, a majority of the published empirical evaluations have focused on measuring how close recommender system predictions are to the user's true preferences (Herlocker *et al.* 2004).

Despite their popularity, these measures do not match the above stated general goal of recommenders. Moreover, some metrics are enforcing recommender systems to follow particular policies, limiting their possibilities. Instead of a common goal for any recommender system, a fuzzy objective prevails at present, which is supplied *de facto* by a diversity of current metrics. To check this, we devote the next section to review the traditional metrics and to discuss some of the problems related to them. Thereafter, we will propose a new framework as a way to overcome them.

A Review of the Metrics for Evaluating Recommender Systems

In this section, we will survey and discuss the most common metrics of the field highlighting the most significant assumptions on which they are based. To this end, we start describing the usually accepted framework followed in the field to define the general recommendation process.

In this current framework, a recommender system is embedded in another system, which contains a number I of items available to be recommended. In order to start the recommendation process, some of those items must have been previously rated. In most of the recommender systems these ratings are obtained explicitly. In some other cases, the ratings are inferred from other users' interactions. In this case they are called implicit ratings.

Once the recommender system has collected ratings enough, the process can start. For each recommendation, a number $N \leq I$ of objects are chosen by the recommender, and shown to the target user. Additionally, some recommender systems also rank the marked-out objects to display them as an *ordered list*. After this, the user will *presumably* investigate these items starting at the top of this list.

Finally, in order to evaluate the performance of the recommender system, for each object shown to a particular user it is estimated how *close* the utility of the *shown* object is with respect to the preferences of the user. In the case of an ordered list, additionally, it should be taken into account the place that each recommended object has in this list. Now, we will have a quick view on how this evaluation has been carried out to date.

First considerations

Let us call $P(u, i)$ the predictions of a recommender system for every particular user u and item i, and $p(u, i)$ the real preferences. The function $p(u, i)$ must be either explictly expressed by de user or implicitly calculated by taking into account (generally) some relevant past actions of such user in this environment.

Sometimes, both functions $p(u, i)$ and $P(u, i)$ will offer only two values 1 or 0, so that a particular item i is consid-

ered *useful* or *useless*, respectively, for a particular user u. For this singular case, we will say that p and P are *binary* functions.

Accuracy metrics

Particularizing to the recommender system's field, accuracy can be formulated as in (1). Under this form, $accuracy$ can be found in many works (Pazzani, Muramatsu, & Billsus 1997; Armstrong *et al.* 1995; Burke 2002).

$$accuracy = \frac{number\ of\ successful\ recommendations}{number\ of\ recommendations} \tag{1}$$

Additionally, most authors assume that a "successful recommendation" is equivalent to "the usefulness of the recommended object is *close* to the user's real preferences", and using the functions p and P introduced previously, we can reformulate $accuracy$ as in (2). In this equation, p and P are considered *binary* functions. Additionally, $r(u,i)$ is 1 if the recommender showed the item i to the user u, and 0 otherwise. Finally, $R = \sum_{u,i} r(u,i)$ is the number of recommended items shown to the users.

$$accuracy = \frac{\sum_{(\forall u,i/r(u,i)=1)} 1 - |p(u,i) - P(u,i)|}{R} \tag{2}$$

Also common in the recommender systems' field is the metric *mean absolute error* (MAE). This metric measures the average absolute deviation between each predicted rating $P(u,i)$ and each user's real ones $p(u,i)$. We may derive (3), where i must have been rated by u (to obtain $p(u,i)$). In this, we consider N as the number of observations available, which depends on the number of items properly rated.

$$MAE = \frac{\sum_{u,i} |p(u,i) - P(u,i)|}{N} \tag{3}$$

Several recommender systems make use of this metric for the evaluation (Breese, Heckerman, & Kadie 1998; Herlocker *et al.* 1999; Shardanand & Maes 1995). Also, there are some direct variations of MAE. For instance, *mean squared error*, *root mean squared error*, or *normalized mean absolute error* (Goldberg *et al.* 2001).

Information retrieval measures

Information Retrieval (IR) is a consolidated discipline whose objectives are somehow related to the ones of the recommender systems (RS) field. Moreover, IR research is focused on the *retrieval* of *relevant* documents from a pool, which is not far from the related RS task of recommending *useful/interesting* items "from a large space of possible options". Among the IR tools of interest for the RS field, we find its metrics, specially: $precision$ & $recall$ (Baeza-Yates & Ribeiro-Neto 1999). Furthermore, we can find the related ROC (Receiver Operating Characteristic) analysis (Haley & Mcneil 1982). Several recommender systems have been evaluated so far by them (Billsus & Pazzani 2000; Herlocker *et al.* 1999).

	relevant	non-relevant
retrieved	a	b
not retrieved	c	d

Table 1: Confusion matrix of two classes when considering the retrieval of documents. Diagonal numbers a and d count the correct *decisions*: retrieve a document when it is relevant, do not retrieve it when it is non-relevant. The numbers b and c count the incorrect cases.

To compute these metrics, $precision$, see (4), and $recall$, see (5), a confusion matrix is expected such as the one in table 1. This table reflects the four possibilities of any *retrieval decision*. In order to work with *recommendation decisions* and translate them into the RS field, the IR terms "retrieved" and "relevant" are usually assumed to be switched to the RS terms "recommended" and "successful recommendation" respectively. Again, notice we are working with recommender's *decisions* where in principle no ratings are needed. The goodness of recommendations can also be evaluated in a binary way (by means of thresholds) in the sense of assigning 1 to a good recommendation and 0 to those insufficient ones.

$$precision = \frac{a}{a+b} \tag{4}$$

$$recall = \frac{a}{a+c} \tag{5}$$

The meaning of each measure is very intuitive. $Recall$ measures the capacity of obtaining the most useful items as possible present in the pool. On the other hand, $precision$ shows the recommender's capacity for showing only useful items, while trying to minimize the mixture of them with useless ones.

As a result, we look for an optimization of $recall$ and $precision$, both at the same time.

An alternative to the last metrics is ROC analysis. A ROC curve represents $recall$ against $fallout$ (6). The objectives of ROC analysis are to return both the most (ideally all) of the relevant documents and the minimum of irrelevant ones (ideally none) at the same time. It does so by maximizing $recall$ (called the true positive rate) while minimizing the $fallout$ (false positive rate).

$$fallout = \frac{b}{b+d} \tag{6}$$

Notice that the optimization of a ROC curve is similar to the optimization of $precision/recall$ curves. What is more, methodologically, optimizing ROC curves is equivalent to optimizing $precision/recall$ curves (Fisher, Fieldsend, & Everson 2004; Davis & Goadrich 2006). As a result, we can focus the evaluation on whatever of both analysis.

Some other metrics derived from $precision/recall$ are *F-measures* (7), which try to grasp in a single value the behavior of both $precision$ and $recall$ metrics. Thus, varying β, the value of F_β weights one metric over the other. However, the most usual of the *F-measures* is $F1$ ($\beta = \frac{1}{2}$, see (8)), which is the harmonic mean of $precision$ and $recall$.

$$F_\beta = \frac{precision\, recall}{(1-\beta)precision + \beta\, recall} \quad (7)$$

$$F1 = \frac{2\, precision\, recall}{precision + recall} \quad (8)$$

Rank metrics

These metrics are used in the case of recommenders based on the display of an ordered list of elements. These systems provide a ranked list of recommendations where those that rank highest are predicted to be the most preferred.

In the spirit of quantifying the *closeness* of recommender's predictions to users' real preferences, some rank metrics measure the *correlation* of the rank of predictions $P(u, i_1) \geq P(u, i_2) \geq P(u, i_3) \ldots$ to the rank of real preferences $p(u, i_1) \geq p(u, i_2) \geq p(u, i_3) \ldots$. Examples of systems that apply these techniques are Hill et al. (Hill *et al.* 1995) which applies the well known Pearson's product-moment correlation or Fab (Balabanovic & Shoham 1997) which applies NDPM (Normalized Distance-based Performance Measure).

Alternatively, other rank metrics as Half-life utility (Breese, Heckerman, & Kadie 1998; Heckerman *et al.* 2001) weight decreasingly this predicted/real-preference closeness. To this end, they postulate that each successive item in the ordered list is likely to be viewed by the user with an exponential decay.

Other metrics

In the life of the recommender systems field a lot of ad-hoc measures have appeared. However, most of them are not far from the accuracy metrics $accuracy$ or MAE explained above. In fact, we will see that they are mostly related to one of both.

For instance, some systems, such as (Billsus & Pazzani 1998), make use only of the first top n recommended items in order to compute $accuracy$. Other systems as INTRIGUE (Ardissono *et al.* 2003) or SETA (Ardissono & Goy 2000) use a metric named *satisfaction score*, which measures the degree of matching between an item and a group of users by analyzing the preferences of a group of users and the properties of the item. The assumptions are similar to those made by MAE, but using groups of users (stereotypes) instead of single users.

Discussion

Nowadays, there is a lack of uniformity in the current metrics for the evaluation of recommender systems. We will attempt to classify them into one of the next three categories, depending on the way they quantify the good behavior of the recommender system.

1. *Rating prediction*. These metrics are focused on measuring the capacity of the recommender system for predicting the rating a user will give to an item before she does it.
2. *Ranking prediction*. These metrics are focused on measuring the capacity of the recommender system for predicting the rank a user will set on a set of items before she does it.
3. *Successful Decision Making Capacity (SDMC)*. These metrics are focused on measuring the capacity of the recommender system for making successful *decisions* (recommendations).

Bearing this classification in mind, we should classify MAE (and related metrics) into the first class, ranking metrics into the second class, and $accuracy$ (and related metrics) and IR metrics into the third class.

Now, if we bring forward the main goal of a recommender system and we observe what the first and second class of metrics try to measure, we could think of some kind of "over-particularized" metrics. In fact, we should not make more assumptions than the ones actually required. However, there is no mention to any rating or rank in the definition of a recommender system's goal. Moreover, even though assuming that a useful item might be one whose $p(u, i)$ is high enough, it is really arguable that we can derive an exact $p(u, i)$ function only by means of users' ratings (Cosley *et al.* 2003). In conclusion, if our desire is to be strict while building up a metric that measures just the real objective of *any* recommender system, we must be cautious while making assumptions that could set apart some proper recommender systems from being measured.

Therefore, if we want to keep the methodology general enough to include as recommender whatever system with the already stated objective, we must bear in mind that this goal is expressed in terms of the recommender system's decisions. Thus, *SDMC* metrics appear as the most appropriate for this task.

Nevertheless, even bearing the latter in mind, notice some problems behind the use of SDMC metrics. In fact, it is widely considered that a *successful recommendation* is that one whose interest/usefulness corresponds to the target user's real interest. In other words, a *successful recommendations* is that one which comply $|P(u, i) - p(u, i)| = 0$. At this point, we will highlight the important assumptions which stand behind this popular belief:

1. It is widely assumed that a recommendation is successful if and only if the *recommended item* is useful. However, a recommendation purpose is also to *guide* the user. Moreover, if a recommendation is not "opportune" nor "attractive" enough for guiding the user to the recommended item, the whole recommendation will be of no use, in spite of the usefulness of this item. In other words, it is a requirement to choose *when* and *how* to recommend, apart from the common decision of *what* to recommend.
2. It is widely assumed that a recommended item is useful if and only if it matches the target user's preferences. However, this is not always the case. For instance, many e-commerce systems consider a recommended item to be useful whenever it involves a transaction. Actually, it might be a requisite. Naturally, the latter could have nothing to do with user's preferences. Therefore, the usefulness of a recommended item must be reconsidered and generalized.

For the above mentioned reasons, we claim there is an "over-specification" in the current metrics for recommender

systems. As a solution, we will provide a framework general enough to liberate recommender systems from following stipulated policies, but particular enough to obtain concrete results. To this end, we will presume *only* the objective that features recommender systems: to *guide* the users to *interesting/useful* objects, and assuming nothing about the target (traditionally the user, but could be not) of this interest/usefulness.

Proposal: A General Framework for Recommender Systems

In this section we will develop a new framework for recommender systems. We call it *general* because this framework will introduce no more assumptions than the common objective of recommender systems. The goal consists in achieving a concrete framework with a common terminology.

In order to build up such a framework we cannot expect to work with individual recommenders. Instead of that, we will work with categories of recommender systems. Through the years, several initiatives for classifying these systems have arisen. The most familiar of them consists of categorizing them into two classes: *collaborative filtering* and *content-based filtering* (Adomavicius & Tuzhilin 2005). Furthermore, hybrids of these can also be found (Burke 2002; Balabanovic & Shoham 1997).

Despite its popularity, this classification is extremely dependant on the type of algorithm used by the recommender system, and it could leave proper recommender systems away from consideration. In fact, Burke had to extend this categorization by adding two more classes: *demographic* and *knowledge-based* classes (Burke 2002). In addition, this categorization could need to be extended again in the future.

In order to grasp the whole range of recommender systems in a categorization from the beginning, we must focus on their most essential feature: again, their objective. Focusing on this objective, we may separate it into two sub-objectives: (i) to *guide*, and (ii) to *filter* useful/interesting items. The first part has to do essentially with an interactive, dynamic and very temporal behaviour; while the second part has to do with a somewhat opposite behaviour, more permanent and less directly interactive.

This encourages us to introduce a new categorization based on which sub-objective a particular recommender system is most centered on. In fact, in the next section we will check that most of the current recommender systems are extremely biased over one of the two sub-objectives, leaving almost unattended the other one. Because of that, as a first and strict division, we will classify all possible recommender systems as *interactive* or *non-interactive* respectively, depending on which part of the objective they are more focused on. This concept will be extended in the following subsection.

Interactive and non-interactive recommender systems

We will proceed to develop the concept in an inductive way. To this end, we will use two characteristic examples of *interactive* and *non-interactive* recommender systems. Two popular but opposite systems in terms of interaction are Syskill&Webert (Pazzani, Muramatsu, & Billsus 1996; 1997) and WebWatcher (Armstrong *et al.* 1995; Mladenic 1996). However, they both are usually included together into a same category as content-based recommenders.

Content-based recommenders try to suggest items similar to those considered *interesting/useful* for a given user in the past. As a result, they need to create a user profile for solving this task. In both cases, the items considered are web hyperlinks. In the first case, Syskill&Webert builds its user profile by means of the collected (explicit) ratings given by each user for some visited web pages. In the second case, WebWatcher *assists* each user "looking over her shoulder" by highlighting as *recommendations* some of the hyperlinks present in every shown page. This method is the so-called *annotation in context*. In WebWatcher, the user may click on the recommendation or not, but, whatever is done, the user's action is logged and its user profile updated.

It must be strongly pointed out that Syskill&Webert could have obtained the users' feedback before any recommendation had taken place, whereas in WebWatcher the collected data depend strongly on the previous activity of the recommender over the user. Therefore, to distinguish between these two types of recommender systems, we propose calling *interactive recommenders* those systems similar to WebWatcher, leaving the name *non-interactive recommenders* to the rest. Note that we do not mean that *non-interactive* recommenders do not need users' interactions, because any adaptive system does require them. However, in the second case, the users' interaction data are collected from an external system which might be not part of the recommender system. In other words, in *non-interactive* recommender systems, users' interaction data can be collected before any user's interaction with the recommender system has ever taken place.

Now, a significative question arises: can *interactive* and *non-interactive* recommender systems be evaluated by means of the same metrics? At a first sight, both of the examples, Syskill&Webert and WebWatcher, were evaluated by the same metric *accuracy*. However, their metrics are not that similar (we will come back to these ideas later). In fact, we can see that in the case of Syskill&Webert, the evaluation objective is to "determine whether it is possible to learn *user preferences* accurately" (Pazzani, Muramatsu, & Billsus 1996). On the other side, in the case of WebWatcher, the evaluation objective is "How well (accurate) can WebWatcher learn to *advise the user*?" (Armstrong *et al.* 1995).

We could be tempted to say that these two systems are different, because they have different objectives. However, we claim they only have one common objective, even though each system pays attention to a different part of it. In fact, in the case of Syskill&Webert, its attention is focused on the second part of the common objective: *determine user preferences accurately* to provide *useful/interesting* items to the user. Though, in the case of WebWatcher, its attention is on the first part of the common objective: *learn to advise the user* to *guide* him correctly.

In conclusion, we propose a framework in which any recommender system is a composition of two different sub-

systems: an interactive and a non-interactive subsystem. Each subsystem will be in charge of its own subobjective respectively: (i) to guide the user and (ii) to provide useful/interesting items. Traditionally, recommender systems have had one of the two subsystems more active than the other, and its class (interactive or non-interactive) depends on this. Finally, notice that the closer both subsystems are to their own subobjective at the same time, the closer the whole recommender system is to its global objective. The next section is devoted to detail these two subsystems.

The guide and the filter subsystems

In our framework we propose that any recommender system is composed of two subsystems:

- The *interactive* subsystem is in charge of *guiding* the user. Thus, we will call it *the guide*. In other words, *the guide* must answer *when* and *how* each recommendation must be shown to the user.
- The *non-interactive* subsystem is in charge of *choosing* the interesting/useful items among the large quantity of them. Thus, we will call it *the filter*. In other words, *the filter* must answer *which* of the items are useful/interesting candidates to become recommended items.

Note the subtle use of the terms "recommendation" and "recommended items" in the last two definitions. The point is that a full *recommendation* contains more information than a simple *recommended item*. In fact, a recommendation is composed of two parts: (a) the item to be recommended (*what/which*), plus (b) the way and the context in which the recommender must recommend (*how* and *when* respectively).

From this point of view, the process of performing a recommendation can be seen as a two steps policy: (1) election of relevant items from a (large) pool of them (*what* to recommend), and (2) generation of "frames" which contain the portrayal of the already chosen items answering both *how* and *when* they must be shown to the user. According to this, *the guide* and *the filter* subsystems of a recommender system are differentiated parts of it, which are responsible for reaching their own objectives by themselves.

In summary, a general recommendation process can be defined as follows. First, the filter collects a number of items from the pool. Then, this subsystem is the main responsible for the *usefulness* of the items. Second, the guide shows the recommendations to the user. As a result, this subsystem is the main responsible for the recommendations being *followed*. In the next section we will define this process and its components in a more formal way.

A formalized recommendation process for a general recommender system

In this section we will formalize the building blocks of the framework. We will divide the exposition in three parts: basic elements, subsystems goals and recommendation process.

Basic elements For each recommendation, we define the following elements:

- A *recommendation* is a discrimination of information in the sense of "what", "when" and "how" it must emerge. In addition, its feedback presents two faces: (i) the usefulness, which reveals the quality of "what" (ii) the followability, which uncovers the quality of "when" and "how".
- We call *recommendation event* to the single event that induces the recommender system to perform a recommendation. Likewise, we can define ev_r and $|ev_r|$ as the whole *set* and *number* respectively of the recommendation events considered.
- I: full set of items available to be recommended during a recommendation.
- *recommendation window* rw: subset of I created after every new *filtering process*. We will call every item belonging to a recommendation window *candidate item to be recommended* $i \in rw$.
- *filter*: subsystem in charge of *creating* and *filling* a new recommendation window rw.
- *guide*: subsystem in charge of creating and displaying to the user the most "followable" recommendations. It does so (1) by collecting the opportune[1] items from the recommendation window rw and (2) by choosing the best portrayal[2] for the final display to the user.

Subsystem goals

Definition 1 *A recommendation is called a followed recommendation if and only if a user has made use of it. This fact is independent of the usefulness or interest provided by this recommendation to this user or any other agent.*

Definition 2 *The goal of the guide subsystem of a recommender system consists of displaying the highest possible number of recommendations that are going to be followed.*

Note that in the last definition there is no reference to the *usefulness* of the recommendation. Actually, it is not the matter of the *guide* subsystem. Instead, the usefulness of the recommendation is an issue to be solved by the *filter* subsystem.

Definition 3 *The goal of the filter subsystem consists of achieving that most/all of the followed recommendations are useful/interesting.*

Once the basic elements of the common recommendation process have been introduced, we may define its mechanism.

The recommendation process Formally, the steps of the recommendation process are enumerated as follows:
For each *recommendation process*

1. From the set I of all objects available, the *filter* subsystem chooses a *useful* subset of them. We defined this subset as *recommendation window* rw.

[1] Opportune in terms of the appropiate moment: *when* the recommendation is going to be followed.

[2] A best portrayal in terms of *how* to display the recommendation.

	useful	useless
belong to rw	a	b
out of rw	c	d

Table 2: Confusion matrix for *non-interactive* recommender systems. Notice that there must be a new confusion matrix for each filtering, because of the creation of a new rw.

2. The *guide* collects the *opportune* (see above) items from rw and, by choosing the *best portrayal* (see above), transforms the items into *followable* recommendations.
3. Finally, the guide displays the *recommendations*.

Measures for Recommender Systems

Once we have the formal elements, we can safely start building on top of them. We begin by reconsidering the traditional metrics so that we can apply them over the framework developed in the previous section. Afterwards, we will develop a new metric. We will conclude the section by studying the advantages of this new metric over the traditional ones.

Traditional metrics

First of all, we observe the difference of considering both interactive and non-interactive subsystems while constructing a metric in this framework. Actually, it is not an usual thing, and present metrics have not been developed with this idea in mind. Thus, whenever we are working with a non-interactive recommender system, we will say that its *guide* subsystem is *forced*, so that we can avoid this system from consideration. In a similar way, we will avoid the *filter* subsystem in an *interactive* recommender system by defining it as forced as well. However, we claim that when any one of those subsystems are avoided from being measured, the real fact is that they are implicitly incorporated into the measure, but as if they were perfect subsystems.

In the case of a non-interactive recommender system, the forced guide will display every recommended item present in the recommendation window rw. Therefore, we will obtain a confusion matrix as the one in table 2. Note the similarity with table 1. Consequently, for non-interactive recommender systems, we arrive at the same traditional SDMC metrics $recall$, (5), $precision$, (4), and $accuracy$, (2).

In the case of an interactive system, embodied into this recommender system we will find a forced filter which only presents useful/interesting items to the guide. As a result, only the follow-ability of the recommendations is to be considered. Here, the most critical decision the guide must determine is whether the recommendation will or will not be displayed to the user. Therefore, we may see it as if the guide had two main decisions: (i) to display the item as a recommendation, (ii) not to display the recommendation. Next, the user has two possibilities: (i) to follow the recommendation, (ii) not to follow it. All this facts can be summarized in a confusion matrix as the one in table 3.

Now, if we compare the table 3 with the table 2 we will observe some similarities. However, in this case, a noticeable fact appears: $c = 0$. In other words, a never displayed

	followed	not followed
displayed	a	b
not displayed	c	d

Table 3: Confusion matrix for *interactive* recommender systems.

recommendation can never be followed. Due to this singular fact, it is usual to measure these recommenders taking into account just the first row of the table. This leads us to a metric as the one in (9). Notice the similarity of this equation with (4). In fact, it is not a surprise that in the context of interactive recommender systems "accuracy" and "precision" terms are considered almost synonyms (Mladenic 1996).

$$accuracy = \frac{a}{a+b} \tag{9}$$

Observe that even when interactive and non-interactive systems can use similar metrics for their evaluation, the terms involved in their equations are completely different. Particularly, as an example, c will never be the same for interactive and non-interactive recommender systems.

In conclusion, in this section we have reformulated the SDMC traditional metrics in the terms of our proposed framework. Here, different confusion matrices appear. Consequently, different metrics are applied to these two types of recommender systems, though sometimes with equivalent names. In the next section, we will develop a new metric which takes these differences into account and, among other things, this will allow to measure both types of recommender system by the same metric.

A Metric Proposal: Performance $\mathcal{P}$ of a General Recommender System

In this section we will develop a new metric that emerges naturally from our framework. We will call this metric *performance* $\mathcal{P}$ of a general recommender system. This name expresses clearly what we are looking for: an evaluation of the *global* execution of a general recommender system. As we have repeatedly mentioned, the main goal of a recommender system consists of *guiding* the user to *useful* items. Also, we confirm that the goal is achieved in every single recommendation whenever it complies with the next two rules at the same time: (i) the recommendation is *followed*, (ii) the recommendation is *useful/interesting*.

With all these previous ideas in mind, we are ready to develop the new metric $\mathcal{P}$. To this end, we will take similar steps as those we took to reconsider the traditional metrics in the previous section. In fact, we will base the new measure on a mixture of the confusion matrixes that appear in tables 2 and 3. In this case, the quantities of this matrix will be primarily computed by counting the number of both *followed* and *useful* recommendations that the recommender system has had to consider. This confusion matrix can be seen in table 4. Note the extreme similarity with the confusion matrix of table 3. This fact looks natural if we bear in mind that the guide is the external face of the recommender system.

	followed and useful	rest of the cases
displayed	α	β
not displayed	–	δ

Table 4: Confusion matrix for *general* recommender systems. Notice that we do not consider the quantity $c = 0$ anymore.

As a first try, a good metric could consist of just counting the number of *followed* and *useful* recommendations that the recommender system has displayed. Why not, at a first sight it seems clear that the higher this number, the better the recommender. However, the problem appears whenever we try to compare between recommender systems that have endured different lengths of time. In other words, we need to compare recommender systems regardless of the number of recommendation events $|ev_r|$ or recommendation opportunities they have experienced. To this end, we define the metric $\mathcal{P}$ (10).

$$\mathcal{P} = \frac{\alpha}{|ev_r|} \tag{10}$$

Note that this metric is not only an average, but an estimator of the number of *good* recommendations we *expect* to find in the following recommendation events. Obviously, the higher the number of recommendation events $|ev_r|$, the better will be the estimation. In addition to this single but significative advantage, we will enumerate several ones not less important.

1. This metric has been directly derived from the general objective of recommender systems. Therefore, there is no assumption about the source of knowledge about the usefulness/interest of the items in any recommendation. Needless to say, this usefulness/interest can still be derived from previous ratings over the item, however it is not a demand anymore.
2. Unlike traditional metrics for the evaluation of *non-interactive* recommender systems, $\mathcal{P}$ do not require the knowledge of the usefulness of any other item than those which belong to the *followed* recommendation. This is a very important issue. For instance, imagine an e-commerce system whose main source of knowledge about the usefulness/interest of its items come from the transactions they generate for a particular user. In this case, the usefulness/interest of most of the items present in the system will be unknown. Even more, with high probability the usefulness/interest of many items into this system will change over time. In fact, there are several collaborative filtering based recommender systems whose ratings can evolve over time (Cosley *et al.* 2003). Consequently, some recommender systems require to detect and to take into account the usefulness/interest of the item just in the precise instant in which the recommendation has been displayed and followed by the user.
3. Unlike traditional metrics, $\mathcal{P}$ does not require any additional computation of averages.
4. Unlike traditional metrics, $\mathcal{P}$ does not have to be "translated" whenever we evaluate different recommender systems as an *interactive* or a *non-interactive* recommender system. Moreover, this metric, like our proposed framework, is applicable regardless of the recommender system type, because they are as general as the main objective of the recommender systems. As a result, it allows, among other things, to evaluate the whole range of recommender systems by means of a single framework/metric.
5. As a consequence of the last point, $\mathcal{P}$ gathers the performance quantification of both internal subsystems (the guide and the filter) together in a single value. Naturally, unlike traditional metrics, this fact occurs even if the other subsystem is not forced.
6. Unlike traditional metrics, $\mathcal{P}$ does not have a supreme value. This fact matches the intuitive idea that there is always a better possibility for recommending more and better. In other words, there is not a best recommender, but better and better ones.

Finally, someone could claim that $\mathcal{P}$ does not address usual problems that traditional metrics do. As an example, with $\mathcal{P}$, it seems that if the guide displayed a million recommendations of which the user followed ten (which were indeed useful), this is as good as if the guide displayed ten recommendations all of which the user followed (and all of which were useful).

To clarify this question note that $\mathcal{P}$, in contrast to traditional ones, is focused on measuring the *final objective* of a general recommender system. Therefore, the metric cannot measure *how* this objective must be achieved, but only if it is really achieved. All things considered, displaying a huge number of recommendations could be fine, depending on the context and the kind of application considered. Thus, we claim that $\mathcal{P}$ does not get involved in the details of *how*, but just focus on *what* must be obtained by a general recommender system.

Conclusions and Future Work

Evaluation of recommender systems is a challenging task due to the many possible scenarios in which such systems may be deployed. Traditional evaluation metrics for recommenders are over-specified. In fact, they are usually biased towards the particular techniques used to select the items to be shown, and they do not take into account the global goal of a general recommender system: to guide the user to useful/interesting objects.

The metric $\mathcal{P}$ presented in this paper can be considered an step forward towards this direction, since it considers the followability of the recommendations and the usefulness/interest of the recommended items. Additionally, $\mathcal{P}$ assumes nothing about the target (traditionally the user, but could be not) of this interest/usefulness.

In fact, to provide a common ground for evaluating recommender systems and taking into account the global goal of a general recommender system, we have presented a new framework that considers each recommender as being composed of a guide subsystem and a filter subsystem. While the

filter subsystem is in charge of selecting useful/interesting items, the guide subsystem is the responsible for the display of followable recommendations.

As future work, we have work in progress for separating the global metric $\mathcal{P}$ into two submetrics. By this, we expect to evaluate both the guide and the filter subsystems individually. In our opinion, this submetrics will provide significative knowledge about the recommender system inside, and this fact will be crucial for the modular development of each subsystem alone. For instance, the quality of the *guide* subsystem, as the external face of the recommender system, is closely related to the intrusion cost of the act of recommending (del Olmo, Gaudioso, & Boticario 2005). This point appears really promising for obtaining better guides in future recommender systems.

References

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

Ardissono, L., and Goy, A. 2000. Tailoring the interaction with users in web stores. *User Modeling and User-Adapted Interaction* 10(4):251–303.

Ardissono, L.; Goy, A.; Petrone, G.; Segnan, M.; and Torasso, P. 2003. Intrigue: Personalized recommendation of tourist attractions for desktop and handset devices. *Applied Artificial Intelligence* 8-9(17):687–714.

Armstrong, R.; Freitag, D.; Joachims, T.; and Mitchell, T. 1995. Webwatcher: A learning apprentice for the world wide web. In *AAAI Spring Symposium on Information Gathering*, 6–12.

Baeza-Yates, R., and Ribeiro-Neto, B. 1999. *Modern Information Retrieval*. Essex.: Addison Wesley.

Balabanovic, M., and Shoham, Y. 1997. Fab: Content-based, collaborative recommendation. *Communications of the ACM* 40(3):66–72.

Billsus, D., and Pazzani, M. J. 1998. Learning collaborative information filters. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, 46–54. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Billsus, D., and Pazzani, M. J. 2000. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction* 10:147–180.

Breese, J.; Heckerman, D.; and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Uncertainty in Artificial Intelligence. Proceedings of the Fourteenth Conference*, 43–52. Morgan Kaufman.

Burke, R. 2002. Hybrid recommender systems. *User Modeling and User-Adapted Interaction* 12(4):331–370.

Cosley, D.; Shyong, K.; Istvan, A.; Konstan, A.; and Riedl, J. 2003. Is seeing believing?. how recommender interfaces affect users' opinions. In *Proceedings of CHI 2003*.

Davis, J., and Goadrich, M. 2006. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*.

del Olmo, F. H.; Gaudioso, E.; and Boticario, J. G. 2005. Evaluating the intrusion cost of recommending in recommender systems. In *User Modeling 2005, 10th International Conference*, 342–346.

Fisher, M. J.; Fieldsend, J. E.; and Everson, R. M. 2004. Precision and recall optimisation for information access tasks. In *First Workshop on ROC Analysis in AI. European Conference on Artificial Intelligence (ECAI'2004)*.

Goldberg, K.; Roeder, T.; Gupta, D.; and Perkins, C. 2001. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval* 4(2):133–151.

Haley, J., and Mcneil, B. 1982. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology* 143:29–36.

Heckerman, D.; Chickering, D.; Meek, C.; Rounthwaite, R.; and Kadie, C. 2001. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machince Learning Research* 1:49–75.

Herlocker, J. L.; Konstan, J. A.; Borchers, A.; and Riedl, J. 1999. An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 230–237. New York, NY, USA: ACM Press.

Herlocker, J. L.; Konstan, J. A.; Terveen, L. G.; and Riedl, J. T. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22(1):5–53.

Hill, W.; Stead, L.; Rosenstein, M.; and Furnas, G. 1995. Recommending and evaluating choices in a virtual community of use. In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 194–201. ACM Press.

Mladenic, D. 1996. Personal webwatcher: Implementation and design. Technical Report IJS-DP-7472, Department of Intelligent Systems, J.Stefan Institute, Slovenia.

Pazzani, M. J.; Muramatsu, J.; and Billsus, D. 1996. Syskill & webert: Identifying interesting web sites. In *Proceedings of the National Conference on Artificial Intelligence, Vol. 1*, 54–61.

Pazzani, M. J.; Muramatsu, J.; and Billsus, D. 1997. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning* 27:313–331.

Resnick, P., and Varian, H. R. 1997. Recommender systems. *Communications of the ACM*.

Shardanand, U., and Maes, P. 1995. Social information filtering: Algorithms for automating 'word of mouth'. In *CHI'95: Proceedings of the Conference of Human Factors in Computing Systems*. ACM Press.