Evaluation of Recommender Systems

Introduction :-

It is difficult to deny that comparison between recommender systems requires a common way for evaluating them. Nevertheless, at present, they have been evaluated in many, often incompatible, ways. We affirm this problem is mainly due to the lack of a common framework for recommender systems, a framework general enough so that we may include the whole range of recommender systems to date, but specific enough so that we can obtain solid results. In this paper, we propose such a framework, attempting to extract the essential features of recommender systems. In this framework, the most essential feature is the objective of the recommender system. What is more, in this paper, recommender systems are viewed as applications with the following essential objective. Recommender systems must: (i) choose which (of the items) should be shown to the user, (ii) decide when and how the recommendations must be shown. Next, we will show that a new metric emerges naturally from this framework. Finally, we will conclude by comparing the properties of this new metric with the traditional ones. Among other things, we will show that we may evaluate the whole range of recommender systems with this single metric.

Recommender systems were originally defined as ones in which ‘‘people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients’’. Now, a broader and more general definition is taking place in the 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’’. The last implies that current recommender systems have a clear main objective: to guide the user to useful/interesting objects. As a result, evaluation of recommender systems implies assessing how much of this goal has been achieved.

Previous metrics for evaluating recommender systems :-

Whenever a recommender system is evaluated, the metric used for it is built upon certain assumptions. Therefore, we will devote this section to a review of the most common metrics, highlighting the assumptions on which they are based and which we consider most significant. To this end, first of all, we start describing the commonly 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 be rated. In most of the recommender systems these ratings are obtained explicitly.

In some other cases, the ratings are inferred from other users’ interactions, then they are called implicit ratings. Afterwards, once the recommender system has enough ratings, it can start the process. For each recommendation, a number N 6 I of objects are chosen by the recommender, and shown to the target user. Additionally, some recommender systems also rank the marked-out objects in order to show them as an ordered list. Next, the user presumably will 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 we must measure 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, we should take 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 :-

In order to measure the closeness of predictions to users' real preferences, a numerical representation is normally used. In addition, for reasons of clarity, we will use the same notation along the current section. To this end, 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. Clearly, the function p(u,i) can never be known with absolute precision. Therefore, the values of this function are most usually estimated by means of the users' previous ratings. As we said above, these ratings can be obtained explicitly or implicitly. In some cases, both functions p(u,i) and P(u,i) will offer only two values 1 or 0, which means that a particular item i is considered 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 :-

Accuracy metrics measure the quality of nearness to the truth or the true value achieved by a system. Perhaps, accuracy is the most used and well-known metric into the field of Artificial Intelligence, and, in general terms, it can be formulated as in (1). Particularizing the metric to the recommender system's field, it can be formulated as in (2). Under this form, accuracy can be found in the evaluation of many cases.

$$accuracy = \frac{number\ of\ good\ cases}{number\ of\ cases} \quad (1)$$

$$accuracy = \frac{number\ of\ successful\ recommendations}{number\ of\ recommendations} \quad (2)$$

Now, assuming 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 may be more formal and reformulate accuracy as in (3). In this equation, we consider that p and P are binary functions. Additionally, r(u,i) is 1 if the recommender showed the item i to the user u, and 0 otherwise. Finally, R ¼ P u;i rðu; iÞ is the number of recommended items shown to the users.

$$accuracy = \frac{\sum_{(\forall u,i/r(u,i)=1)} 1 - |p(u,i) - P(u,i)|}{R} \quad (3)$$

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). Then, due to the fact that only rated items can show us each user's real preferences, we may derive the (4), 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 obviously depends on the number of items properly rated. Of course, the higher this number, the better the estimate.

$$MAE = \frac{\sum_{u,i} |p(u,i) - P(u,i)|}{N} \quad (4)$$

Several recommender systems make use of this metric for the evaluation . Also, there are some direct variations of MAE. For instance, mean squared error, root mean squared error, or normalized mean absolute error (Goldberg, Roeder, Gupta, & Perkins, 2001). Finally, notice the similarity between accuracy and MAE metrics. In fact, if we consider that p and P are binary functions, and also consider that the (MAE) number of recommender predictions is the same as the (accuracy) number of recommender recommendations (thus, N = R), then we can collapse both formulas into one: MAE = 1 – accuracy.

Presently, consider it a significant idea which will repeatedly appear during the following metrics.

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, this 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 pool. Therefore, it is not a surprise that the IR field is a good supplier of tools for the RS field. Among these tools, we find its metrics, whose key ones are: precision & recall. Also, we can find the related ROC analysis. In fact, several recommender systems have been evaluated by them.

To compute these metrics, precision, see (5), and recall, see (6), 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, to use them into the RS field, firstly we must switch the IR terms ‘‘retrieved’’ and ‘‘relevant’’ to the RS terms ‘‘recommended’’ and ‘‘successful recommendation’’ respectively. Again, notice we are working with recommender’s decisions, thus in principle no ratings are needed.

Table 1
Confusion matrix of two classes when considering the retrieval of documents

	Relevant	Non-relevant
Retrieved	*a*	*b*
Not retrieved	*c*	*d*

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.

Even though, we can always translate non-binary (rating) functions p(u,i) to binary ones (by means of thresholds).

$$precision = \frac{a}{a+b} \tag{5}$$

$$recall = \frac{a}{a+c} \tag{6}$$

The meaning of each measure is really intuitive. Thus, recall represents the coverage of useful items the recommender system can obtain. In other words, this metrics measures the capacity of obtaining all the useful items 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. Observe that we can always improve one of the metrics by declining the other (see Fig. 1). For instance, a recommender system could recommend a large number of items to the user, then the coverage would be maximal (almost all the useful items would be shown), though the precision would be as bad as the proportion of useful items present in the pool. As a result, we look for the optimization of recall and precision, both at the same time (see Fig. 2). An alternative to the last metrics is ROC (Receiver Operating Characteristic) analysis . A ROC curve represents recall against fallout (7). The objectives of ROC analysis are to return all of the relevant documents without returning the irrelevant ones. It does so (see Fig. 3) by maximizing recall (called the true positive rate) while minimizing the fallout (false positive rate).

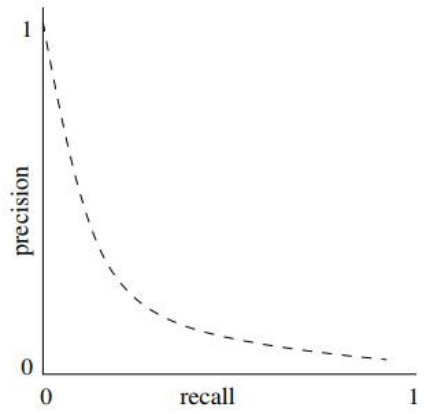

Fig. 1. A common curve representing *precision* against *recall*. Notice that the higher is one of the metrics, the less is the other one.

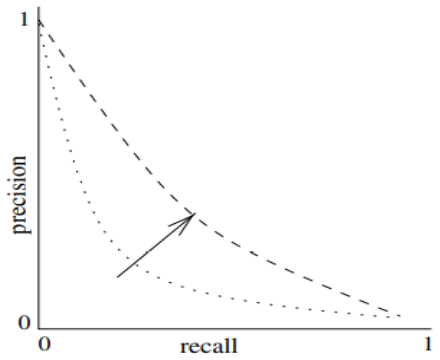

Fig. 2. The simultaneous optimization of *recall* and *precision* can be graphically represented as pushing the peak of the curve towards the right-top corner (towards the point $recall = 1$, $precision = 1$).

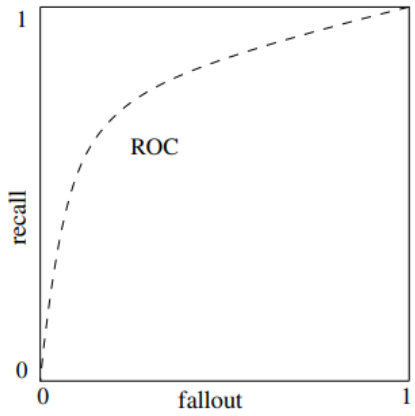

Fig. 3. Common ROC curve representing *recall* against *fallout*. Notice that ROC curve is a monotonically increasing function. Thus, the earlier it gets the top value of *recall*, the better the ROC result. In fact, the Area Under this Curve (AUC) is usually used as a measure of this result.

$$fallout = \frac{b}{b+d} \tag{7}$$

Notice that the optimization of a ROC curve is similar to the optimization of precision/recall curves (see Fig. 4). What is more, methodologically, optimizing ROC curves is equivalent to optimizing precision/recall curves . As a result, we can focus the evaluation on whatever of both analysis. Some other measures derived from precision/recall are F-measures (8), which try to grasp in a single value the behavior of both precision and recall metrics. Thus, varying b, the value of Fb weights one metric over the other. However, the most usual of the F-measures is F1 ðb ¼ 1 2 Þ, see (9), which is the harmonic mean of precision and recall.

$$F_\beta = \frac{precision\ recall}{(1-\beta)\ precision + \beta\ recall} \tag{8}$$

$$F1 = \frac{2\ precision\ recall}{precision + recall} \tag{9}$$

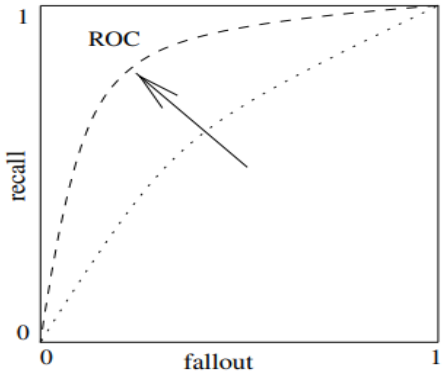

Fig. 4. ROC optimization can be graphically represented as pushing the peak of the curve towards the left-top corner (towards the point *recall* = 1, *fallout* = 0).

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 a recommender's predictions to users' real preferences, some rank metrics will measure the correlation of the rank of predictions P(u,i1) P P(u,i2) P P(u,i3) to the rank of real preferences p(u,i1) P p(u,i2) P p(u,i3) Examples of systems that apply these techniques are which applies the well-known Pearson's product-moment correlation or Fab which applies NDPM (Normalized Distance-based Performance Measure). Alternatively, other rank metrics as Half-life utility weight decreasingly this predicted/realpreference 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 known 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 and Pazzani, make use only of the first top n recommended items in order to compute accuracy. Other systems as INTRIGUE or SETA 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 :-

It cannot be denied that there is a lack of uniformity in the current metrics for the evaluation of recommender systems, which perhaps is due to the large number of them. However, 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 (stated in the beginning of Section 1) 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. 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 objective already stated, we must bear in mind that this objective is expressed in terms of the recommender system's decisions. Thus, SDMC metrics appear as the most appropriate for this task. However, note that when we refer to a useful recommendation in an SDMC metric, it is widely considered that a successful recommendation is one whose recommended item interest corresponds to the target user's real interest. In other words, if we have available the users' binary preference functions, successful recommendations are the ones that comply with the next: jP(u,i) p(u,i)j = 0. At this point, we want to extract two important assumptions that stand behind this popular belief:

Objectives and Implementation

Overall, the project aims to provide a good platform for designers to evaluate recommender systems and guide them to design better recommender systems. Based on the project's specification, there are main five objectives extracted. Blow is the brief description of each projective.

1. **A web application**. A web application is developed to provide GUI interfaces for users to conduct experiments more easily and conveniently. The implementation process goes from prototyping, to the design of pages, to coding, and finally to optimisation. Along the way, the key idea that has been bearing in mind is making the front interfaces as easily interactive as possible.
2. **Implementation of three algorithms**. The three algorithms are all collaborative filtering ones, user-based, item-based and matrix factorisation. The implementation is aided by anopen-source library named LibRec[3]. The work that need to do is to extend the library and integrate it with the web environment.
3. **Evaluation visualisation**. This objective can be said the core part of the project. First, three evaluation methodologies [32] should be integrated to the system. They are repeated random sub-sampling, leave-one-out and K-fold cross validation. Next, some basic evalua- tion metrics (here primarily accuracy metrics) are included. Last, the compare and contrast, namely metrics visualisation between multiple experiments should be finished.
4. **Session mechanism**. The system is developed to support two different modes. The first one is that a user can get access to the application as a guest which means there is no session between the user and the server. The other mode is session mode, which allows users to log in the system using a unique session code. In this mode, users are in session services provided by the server. This objective is outside of the project specification. In consideration of the fact that evaluating a recommender system is often a time-consumingtask, the session mechanism is used to manage experiments and store users' running data including the information of experiments, running times, metric details, etc. Core activities involved in the implementation include database design, management and representation of experiments.
5. **Extension and enhancement**. The objective is considered the advanced improvement for the project. The first extension part falls into the dataset. A file uploading interface is provided for users to apply dataset in a certain format. In addition, the most important extension goes to beyond accuracy metrics. This is what the project core implementation lies on. Beyond accuracy metrics that are extended and enhanced include popularity, diversity,novelty, coverage, etc.

Beyond Accuracy Metrics :-

Prediction accuracy metrics measure how close the items predicted differs from the actual ratings and top-N metrics measure how relevant the items are recommended, while beyond accuracy metrics shift focus towards other properties which significantly describe the performance of recommender systems [8]. Those properties include popularity, coverage, diversity, novelty, etc. The measurements of beyond accuracy metrics are considered difficult because they often involve in the levels associated with user subjective opinions such as user emotions or biases . Despite the difficulty, in literature, still many approaches have been proposed to measure them. In this section, the beyond accuracy metrics involved in the project are reviewed first and then some other out-of-accuracy considerations are mentioned as well.

Popularity :-

In order to better understand the performance of recommender systems, popularity has to be considered. Popularity as an important metric describes how popular the recommended items are. This implies that it is particularly important for e-commerce businesses. As described by [1], if a recommender system is only able to generate popular items, that means there is only around 20% item discovery in the catalogue space leaving the remanding 80% niche items unexplored. In real offline evaluation, a recommender system with lower popularity score is considered wellperformed in exploring potential unpopular items. One simple way to measure the popularity of items recommended to a user is averaging over the percentage of each item's popularity. The equation below shows that the popularity of item j is calculated as the number of users who rated item j (Nj) divided by the number of users in the system (N).

Coverage :-

Coverage refers to a range of products for which a recommender can make a prediction or recommendations . A recommender with a high converge will make a big difference to users. Many papers have considered it as one of the most important metrics to measure a recommender system. In literature, the coverage is often described from three perspectives. First, it can be measured simply by the percentage of target users for who at least one recommendation can be made. This type of coverage tells if all users are likely to receive recommendations in the system. Second, it can be measured by asking "what is the percentage of items that the system is capable of making a prediction or recommendation". This is so called prediction coverage or recommendation coverage. This coverage indicates how large the recommendation space is out of the item space. A simple way to calculate this type of coverage is taking the percentage of items in the dataset which appear at least once in the top-N recommendations made over all target users.

Diversity :-

As opposed to similarity, diversity is defined as a metric frequently applied to the top-N list of recommendations to measure how different the items recommended are. Recommender systems with diverse items recommended can avoid users to explore repeated and useless items, which is not appropriate for the user task in finding all good items. In relevant studies, the most commonly used approach to measure diversity is pairwise dissimilarity, where the similarities between all pairs of items in the top-N recommendation list are computed.

Novelty :-

Another important consideration into beyond accuracy metrics is novelty. It measures the ability of a system to suggest items that users are not aware of. It is an important performance indicator because recommender systems with a lack of novel recommendations make users less know the existence of some items in the catalog. Novelty is often closely related to popularity, or to say opposite to popularity. For example, an approach to measure novelty is called self-information-based novelty which assumes that popular items provide less novelty.

Taxonomy of Recommendation Algorithms

There are so many well-established recommendation algorithms. In literature, they can be gener- ally divided into the following categories.

- **Content-based.**

  Algorithms that use content metadata and user profiles to calculate recom- mendations are called content-based algorithms . It works the way that new interesting items are recommended to a user by matching up the attributes of the user's profile with the attributes of a content item . In order to achieve a content-based recommender system, the most important task is to calculate the item-item similarity or map items toprofiles. Often, the content representation plays a role in the computation, namely extract-ing important item contents and representing them into data structures. The traditional content-based algorithms use information retrieval techniques to represent items in an un- structured manner (using the raw contents of documents). However, for items that are developed by categorical features, such as commodities on e-commercial sites, an alter- native content-based approach named case-based uses a well-defined set of features and feature values to represent items in a more structured manner, allowing for fine-grained judgements about the similarity between items .

- **Collaborative filtering (CF).**

  "When one neighbour helps another, we strengthen our communities.", this quote exactly gives a general picture of how a CF system works. Further speaking, a CF system attempts to generate recommendations by combining the preferences of similar users in the system. Its characteristic of collaborative filtering makes it perform well in personalised recommender systems. As stressed by Herlocker, CF systems are based on human rather than machine analysis of content, which causes them having many significant advantages over traditional content- based techniques. The advantages are concluded in literature: (a) the ability to filter various content, (b) the ability to filter items based on taste and quality, (c) the ability to make serendipitous recommendations. Nevertheless, some unavoidable limitations are also proposed by some researchers. For example, Herlocker et al.point out that the stochastic computation processes and data sparsity in a CF system can lead to being not trusted for high-risk content domains, such as medicine recommendation. This CF techniques are what the project specifically falls into. Hence, more technical details of the three CF algorithms,user-based, item-based and matrix-factorisation in this category are revealed.

- **Hybrid.**

  As known, content-based techniques make recommendations built upon content models while collaborative filtering techniques have good performance in recommending novel and serendipitous items. In order to share the advantages of individual algorithms, hybrid approaches involving two or more recommendation algorithms has been proposed. An example of implementing a hybrid system by combining CF and content-based approaches can be found in.

Apart from the algorithms listed, there are many other different algorithms that have sprung up recent years. For example context-aware recommender systems make recommendations by taking contextual information into account. Constraint-based approaches are used to make the product selection process more effective in complex products such as financial servicesor electronic consumer goods. Conversational approaches are designed to provide an interactive process between the system and the user for collecting user feedback information . Dueto this feature, conversational algorithms are capable of overcoming the problem of insufficient understanding for user preferences in many other recommender algorithms that use the user input data only once to make recommendations. Demographic approaches user attributes classified as demographic data, such as gender, race, age, etc., to make recommendations. An example proving that demographic data can be used to enhance the CF algorithms can found in.

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, Confusion matrix for general recommender systems Followed and useful/interesting Rest of the cases Displayed a b Not displayed - d In this case, note that there is only a single confusion matrix for the whole set of sessions S.

Also, notice that we do not consider the quantity c = 0 anymore. As a mathematical detail, notice that a þ b þ d ¼ P e2evrðSÞjrwðeÞj. 802 F. Herna´ndez del Olmo, E. Gaudioso / Expert Systems with Applications 35 (2008) 790–804 recommenders are biased towards the particular techniques used to select the items to be shown, and they do not take into account the main goal of any recommender: to guide the user to useful/interesting objects. The metric P presented in this paper can be considered an step forward towards this direction, since it considers the follow-ability of the recommendations, apart from the usefulness/interest of the recommended items. In fact, to provide a common ground for evaluating recommender systems and taking into account the main goal of any recommender, we have presented a new general 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 wrapping and showing only those items that are most likely to be followed by the user.

To illustrate both subsystems, we have shown how two very different and well-known recommenders can be defined in terms of our proposed framework. We are now working on obtaining metrics derived from the metric P for evaluating each one of the subsystems individually. It would provide us with a knowledge of the recommender system inside, which we expect to be very useful for future improvements. Related to the latter, although obvious and very often neglected, it must be noticed that the function of the guide subsystem is essential for the good performance of a recommender system. For instance, when the user obtains a lot of items on the screen, she can feel overwhelmed in spite of the usefulness of them (this is usually called intrusion).

Thus, the quality of the guide subsystem is closely related to the intrusion cost of the act of recommending. This point appears really promising for obtaining better guides in future recommender systems.

- ----XXXXX --- --