# Preference-based Search and Machine Learning for Collaborative Filtering: the "Film-Conseil" Movie Recommender System*

Patrice Perny and Jean-Daniel Zucker

LIP6 - Paris 6 University
4 Place Jussieu
75252 Paris Cedex 05 - France

**Résumé**

Cet article présente une nouvelle approche pour l'aide à la décision sur Internet. Elle se caractérise par une méthode de filtrage basée sur les préférences et reposant sur l'intégration de principes de filtrage par le contenu et de principes de filtrage collaboratif. Les algorithmes présentés ont une particularité importante qui était absente dans les premiers systèmes : l'aptitude à expliquer et justifier les recommandations. Nous proposons ici d'intégrer des méthodes de modélisation des préférences et d'apprentissage pour l'aide collaborative à la décision et l'évaluation active. Cette intégration pose à la fois un problème d'efficacité et de stokage pour s'adapter à une large communauté d'utilisateurs. A titre d'illustration, nous présentons les principales caractéristiques du système "Film-Conseil" qui met en oeuvre l'approche proposée pour des tâches de recommandation de films sur Internet.

**Abstract**

This paper introduces a new approach for decision support on the internet. It is characterized by a preference-based filtering relying on the integration of content-based and collaborative filtering principles. We present algorithms that support a key aspect of recommender systems that was absent in early systems: the ability to explain and justify recommendations. A tight integration of preference modeling and machine learning is proposed to support both collaborative decision making and active rating. This integration addresses the problem of both efficiency and storage to scale up with large community of users. For the sake of illustration, we present the main features of the "film-conseil" system which implements the proposed approach for movie recommendation tasks on the internet.

---

*This project was partially supported by a grant from the LIP6 (University Paris 6 Computer Science Laboratory) as part of the WebConseil Project.

**Mot-clés:** Filtrage collaboratif, modélisation des préférences, analyse multicritère, apprentissage, systèmes de recommandation, relations floues.

**Keywords:** Collaborative Filtering, Preference Modeling, Multicriteria analysis, Machine Learning, Recommender Systems, Fuzzy Relations.

# 1 INTRODUCTION

During the last decade, the development of information systems and services on electronic supports, the massive usage of personal computers and the multiplication of web-access providers have established the internet as a major information and communication tool. Due to this success, an exponentially growing amount of information is available on the web and an ever larger number of users are browsing web pages, seeking useful information, documents and products. In this context, the development of tools allowing the content of thousands of electronic resources to be extracted and efficiently stored, and the elaboration of intelligent search tools and automatic filtering systems become a key issue. In this direction, multiple search engines and recommendation systems have been proposed, based on machine learning, classification and aggregation techniques aiming at selecting (or possibly ranking) relevant documents or items for a given user [35, 1]. In the same time, multiple useful techniques based on fuzzy set theory have been developed for the management of imperfect information in a database and the processing of approximative queries (see [6, 8, 5, 9, 22]).

Besides these works concerning information retrieval and data mining, the recent development of the electronic commerce has open new spaces for preference modeling and decision support activities. Nowadays, the internet is not only seen as a new media for advertising and sales enhancement, but also as a powerful tool for improving our knowledge of consumer's preferences and aspirations. Web-advisor systems are not only used to provide users with relevant recommendations for an item but also to collect individual preference information, to model the collective opinion and to anticipate the market trend (electronic marketing). In this respect, the basic research problems to be studied are linked to knowledge and belief representation, individual and collective preference modeling and preference aggregation. Such preoccupations motivate a new body of activities oriented towards information sciences (see [25]) and relying both on artificial intelligence techniques and decision theory.

As an illustration of such new activities, we are interested here in the conception of *collaborative*[1] decision support systems based on the implicit sharing of preferences and experiences between different individuals facing similar decision or search problems. More precisely, "Collaborative decision support" (CDS) is basically concerned with a new category of decision or search problems where *any individual seeks recommendation for his personal choices, the other individuals being only considered as possible advisors*. Despite the multiplicity of possible advisors, the problem addressed in CDS is not a matter of group decision making or negotiation between individuals. Indeed, in CDS, the problem is to find a recommendation that best fits the preferences of a single decision maker, rather than to seek a compromise solution satisfying all the users. The recommendation provided to the decision maker (also called the *active user*) is based on other users experiences without any explicit communication or cooperation. The individuals that have contributed to the recommendation might not be even aware of their role of advisors. On the other hand, despite the fact that each individual is making a decision, CDS is neither an individual decision making problem nor a group decision making. Indeed, there are several decision problems (one per individual) to be addressed simultaneously. Moreover, each individual has an influence on the recommendation provided to other individuals. Therefore the set of decision problems as a whole *cannot* be viewed as mere collection of independent decision problems.

In this paper we assume that several independent individuals called *users* have been connected (*e.g.* through a given web site) to a given decision support system and have expressed their preferences (*e.g.* using grades of satisfaction) about some universal items (*e.g.* movies, CDs, books, pictures...). The problem we address is to provide each individual with relevant recommendation regarding items that s/he has not yet graded. After introducing a general formalism to express different filtering principles, we introduce a set of interpretable content-based and collaborative principles for preference-based filtering (section 2). Then we discuss the main issues linked to the development of an effective recommender system based on these principles (section 3). Finally, we present the main features of the "film-conseil" movie recommender system developed during the past years (section 4).

[1]The term "collaborative" refers to the use of "collaborative filtering" algorithms [40] also called "social filtering" algorithms [42].

# 2 FORMALIZING INFORMATION FILTERING

## 2.1 Basic concepts and notations

Formally, a collaborative decision problem is characterized by a set of users $U$ that make use of the recommander system, a set of items $I$ (possibly relevant for users) and a set of grades assigned to some items by some users. More precisely, assuming that any individual $u \in U$ has graded a subset of items $G(u) \subseteq I$, we want to assess the value of every item in $I \setminus G(u)$ for every user $u$, so as to provide him, with a relevant selection of items.

Let $Z$ be a gradation scale, bounded and completely ordered and let $z_{iu} \in Z$ denote the grade given to $i$ by user $u$, for all $u \in U$ and all $i \in G(u)$. Within $Z$ we distinguish 3 particular grades, namely the maximal grade $z^*$, the minimal grade $z_*$ and a neutral grade $z^0$ dividing the scale $X$ into two parts: the part $[z^0, z_*]$ corresponds to *positive* evaluations whereas the part $[z_*, z^0]$ corresponds to *negative* evaluations. Because people use grades differently, we let the possibility of defining a different neutral grade to each individual. This individual reference point will be denoted $z_u^0$ for any user $u \in U$. It is a priori fixed to the a default value $z^0 \in Z$ for every new user $u \in U$ but may evolve when new grades are entered.

These grades correspond to absolute satisfaction levels expressing the subjective evaluation of individuals with respect to items. The grade $z^*$ means that the user likes very much the item and would strongly recommend it to other users whereas $z_*$ means that the user strongly rejects the item and wishes to advice other users strongly against it. The neutral point $z_u^0$ corresponds to a situation where user $u$ is neither positive nor negative about the item. Several additional reference points corresponding to intermediary levels of attractiveness could be considered as well, so as to facilitate the elicitation of user preferences.

As mentioned in the introduction, we want to complete the collaborative approach with a recommendation principle based on the multiattribute analysis of items. This is only possible when a database is available, providing information about the characteristics of the different items. In this paper, we assume that every item $i \in I$ is described by a tuple $(d_1(i), \ldots, d_n(i))$ of attribute values representing the image of $i$ within a multiattribute space $X = X_1 \times \ldots \times X_n$. Such attribute values may be directly obtained from a database describing items or extracted from a complex document. Usually the attributes refer to very different features of the item and are of different types. Basically, the possible types for attribute values are booleans, integers, reals, strings, and lists.

The elaboration of a recommendation system requires the construction of a relational structure linking users to items, but also users to one another and items to one another. Because these relations may be more or less well established, we have chosen to formalise these notions using the language of fuzzy sets (see [23, 7]). We introduce now a fuzzy relational system which is the basis of our filtering methods. Basically, we can distinguish three types of relations:

- The *Preference* relation $P$ defined on the space $I \times U$ and characterized by the signed membership function $p : I \times U \longrightarrow [-1, 1]$ where $p(i, u)$ must be interpreted as the degree of attractiveness of item $i$ for user $u$. This attractiveness is not necessarily proportional to the grade $z_{iu}$. Moreover, one can readily imagine two different users $u$ and $v$ giving the same grades to an item $i$ may have different preference intensities with respect to $i$. Conversely, two different grades $z_{iu}$ and $z_{iv}$ may represent the same level of satisfaction for users $u$ and $v$. For this reason, it seems preferable to define $p(i, u)$ as a function of quantities $z_{iu}$ and $z_u^0$. For example, let us define $p(i, u) = f_u(z_{iu})$ where $f_u : Z \longrightarrow [-1, 1]$ is a non-decreasing function such that $f_u(z_*) = -1$, $f_u(z_u^0) = 0$ and $f_u(z^*) = 1$. Function $f_u$ reflects the preference intensities for user $u$ and may differ from a user to another (e.g. depending on the gradation interval used by $u$).

  This relation can be split into two fuzzy relations $P^+$ and $P^-$ corresponding to the positive and negative parts of $P$ respectively. These relations are defined as fuzzy subsets of the space $I \times U$ and characterized by the membership functions $p^\sigma : I \times U \longrightarrow [0, 1]$, $\sigma \in \{+, -\}$, where $p^+(i, u)$ (resp. $p^-(i, u)$) must be interpreted as the degree of attractiveness (resp. repulsion) of item $i$ for user $u$. The membership values $p^+(i, u)$ and $p^-(i, u)$ are defined by:

$$
\begin{aligned}
p^+(i, u) &= \max\{p(i, u), 0\} \\
p^-(i, u) &= \max\{0, -p(i, u)\}
\end{aligned}
$$

  Hence, we have $\min\{p^+(i, u), p^-(i, u)\} = 0$ for all pairs $(i, u)$ such that $i \in G(u)$. Moreover $p(i, u) = p^+(i, u) - p^-(i, u)$.

- The fuzzy *resemblance* between users $R$ is defined as a fuzzy subset of the space $U \times U$ and characterized by the membership function $r : U \times U \longrightarrow [0,1]$ where $r(u,v)$ must be interpreted as the subjective resemblance between the two users. Depending on the application, this subjective resemblance can be defined as a measure of closeness between preference judgements or alternatively as an influence relation measuring to which extend a preference judgement should be transferred from a user to another.

- The fuzzy *similarity* between items $S$ is defined as a fuzzy subset of the space $I \times I$ and characterized by the membership function $s : I \times I \longrightarrow [0,1]$ where $s(i,j)$ represents the multiattribute similarity between items, defined from the descriptors $d(i)$ and $d(j)$.

For the sake of illustration, we will use Figure 1 representing relations $P$, $R$ and $S$ on a subset of three users denoted 1, 2, 3 and four items denoted $a, b, c, d$.

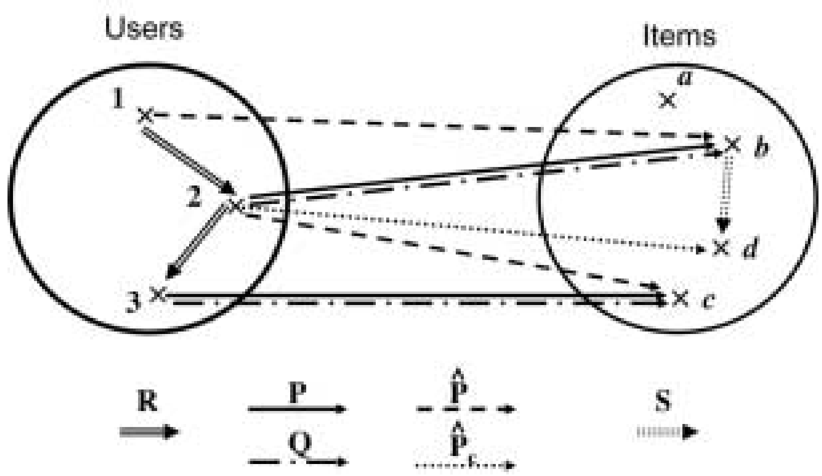


Figure 1: The relational system

These three relations form the core of the recommendation system. We will show in the next section how recommendations can be derived from such relations. In order to formalise the recommendation principle, we will also consider the following relations:

- The expected[2] preference relation $\hat{P}$ defined as a fuzzy subset of the space $I \times U$ and characterized by the membership function $\hat{p} : I \times U \longrightarrow [0,1]$ where $\hat{p}(i,u)$ must be interpreted as the expected degree of attractiveness of item $i$ for user $u$. The relation $\hat{P}$ is not given a priori. It must be derived from the other relations and used by the recommander system to provide users with a relevant selection of items. In the next subsection, we will introduce two different ways of constructing the $\hat{P}$ relation. This is the reason why two different relations denoted $\hat{P}$ and $\hat{P}_c$ are represented on Figure 1.

- The fuzzy *Qualification* relation $Q$ is defined as a fuzzy subset of the space $I \times U$ and characterized by the membership function $q : I \times U \longrightarrow [0,1]$ where $q(i,u)$ must be interpreted as the qualification of user $u$ in its evaluation of the item $i$. This qualification can be given by the user $u$ itself (self-evaluation of its expertise level on the subject) or by other users (peer-evaluation often given with respect to the usefulness of the comments associated to the evaluation of user $u$) or may be assessed by the system based on how useful were the previous recommendations made by user $u$ concerning item $i$ for other users.

**Remark.** The fuzzy relations introduced above are used to represent ill-defined relations attached to vague predicates like preference, similarity, ressemblance and qualification. Hence, the gradations used for membership values must be seen as levels of intensity but not as levels of plausibility. An interesting alternative use of fuzzy sets might be to consider the $P, Q, R, S$ as ill-known relations reflecting the imperfection of the information about preference and similarities. In such a case it might be interesting to re-examine the problem in the setting of possibility theory [24]. This is left for further investigation.

## 2.2 The content-based, non-collaborative approach

There is a large body of recommander systems that are solely based on the analysis of content to provide recommendation. These approaches encompassing the ones used in information retrieval are the ones that have been

[2] the term “expected” has a different meaning here than in ”expected value”. In particular, it does not mean that we are dealing with uncertainty.

used in the past for various kinds of items. The basic idea is to rely on a fuzzy *similarity* between items $S$ and a fuzzy *preference* relation $P$. Following this idea already present in [39], we simply define relation $\hat{P}$ as the composition of fuzzy relations $S$ and $P$:

$$\hat{P} = S \cdot P \tag{1}$$

This can be explained by the following simple principle:

*Item $i$ is recommended to user $u$ if there exists another item $j$ similar to $i$ which is known as attractive for $u$*

Following this principle, we ought to recommend an item $i$ to a user $u$ if and only if $i$ was not graded by $u$ and $iSjPu$ for some $j$. This can be formalized by the following logical condition:

$$\hat{P}(i,u) \;\equiv\; [\neg(i \in G(u)) \;\wedge\; (\exists j \in I \setminus \{i\}, (S(i,j) \wedge P(j,u)))] \tag{2}$$

**Example.** In the case depicted on Figure 1, relations $S$ is represented by a double dotted arrow and $\hat{P}$ by a dotted line. We assume here that $z_{2b}$ is a very positive grade (close to $z^*$). We have $dS.P2$ due to the $SP-$Path: $(d,b),(b,2)$. Considering this path, $d$ ought to be recommended to user 2 because the later expressed a positive opinion on $b$, an item similar to $d$.

Equation (2) defines the preference $\hat{P}(i,u)$ as a combination of the preference indices $P(j,u)$ for all items $j$ which are similar to $i$. In this respect, it can be seen as a case-based decision rule close to those proposed in [20, 21].

One might object that the presence of a single item $j$ in the neighborhood of $i$ is not a sufficient reason to justify the recommendation. Moreover, the close neighborhood of $j$ may contain another item $k$ which is negatively graded by $u$. For this reason, it seems preferable to consider negative as well as positive arguments in the definition of $\hat{P}$. For this reason, we have to detect the presence of positive and negative arguments, represented by the two following conditions, for $\sigma \in \{-,+\}$:

$$\hat{P}^{\sigma}(i,u) \;\equiv\; [\neg(i \in G(u)) \;\wedge\; (\exists j \in I \setminus \{i\}, (S(i,j) \wedge P^{\sigma}(j,u)))] \tag{3}$$

The condition $\hat{P}^{+}(i,u)$ reflects the presence of arguments supporting the recommendation $\hat{P}(i,u)$ whereas the condition $\hat{P}^{-}(i,u)$ reflects the presence of arguments against this recommendation.

Let us assume temporarily that relations $P^+$ and $P^-$ are crisp and that conditions introduced in (3) are interpreted in the classical boolean logic. These two conditions being independent, we obtain four basic different situations (as argued in [2], [28] and [44]). This four situations which are a consequence of the bi-polarity of the preference scale are represented by the following table:

| $\hat{P}^+(i,u)$ | $\hat{P}^-(i,u)$ | $\hat{P}(i,u)$ |
|---|---|---|
| 1 | 0 | true |
| 0 | 0 | unknown |
| 1 | 1 | contradictory |
| 0 | 1 | false |

At this level of description, the first situation (true) is the only one where the recommendation of item $i$ to individual $u$ seems fully justified. In practice, one can expect to observe an artificially large number of cases in which the situation *contradictory* or *unknown* holds, making the recommendation principle inefficient.

In order to refine this first recommandation principle and produce a more discriminating decision procedure, it seems interesting to balance the strength of pros and cons in the assessment of predicate $\hat{P}(i,u)$. In this respect, it is much more interesting to interpret equation (3) in a multivalued logic. To this end, we need to consider relations $P^+$ and $P^-$ as fuzzy relations. In this case, the classical numerical counterpart of equation (3) is given by:

$$\hat{p}^\sigma(i,u) = \bigvee_{j \in I \setminus \{i\}} T(s(i,j), p^\sigma(j,u)) \tag{4}$$

where $\hat{p}^\sigma$ is the membership function characterizing $P^\sigma$, $T$ is a t-norm and $\bigvee$ is the associated t-conorm (for more details see [36]). The choice of the t-norm depends on the application. For instance, choosing the $\max$ t-conorm amounts to focusing only on the more significant positive neighbor and the most significant negative neighbor. Choosing a non-idempotent t-norm allows reinforcement effects when several positively (or negatively) graded objects are around $i$ (see also [29, 39]).

For technical reasons, it is important to restrict the analysis of positive and negative arguments to a limited neighborhood of $i$. It can significantly reduce the complexity of the search of $SP-$paths. Moreover, it prevents the possible drowning of information due to over-reinforcement effects on a too large set of examples. Hence, we define $N_k(i)$ as the set of the $k$ most similar items from $i$ (*i.e.* objects $j \in I \setminus \{i\}$ having the $k$ greatest values $s(i,j)$). This yields the following:

$$\hat{p}^{\sigma}(i,u) = \bigvee_{j \in N_k(i)} T(s(i,j), p^{\sigma}(j,u)) \tag{5}$$

This new equation defining $\hat{P}^{+}$ (resp. $\hat{P}^{-}$) must be seen as a fuzzy k-Nearest Neighbor rule (see [33, 3, 6, 22]) deciding whether an item $i$ belongs to the class of candidates for recommandation (resp. exclusion). The fuzzy similarity relation $S$ plays the role of a distance weighting the importance given to the different (positive or negative) neighbors.

A first selection of relevant items for users $u$ can be obtained according to the following definition :

$$\hat{P}(u) = \{i \in I \setminus \{u\} : \hat{p}^{+}(i,u) \geq \gamma \text{ and } \hat{p}^{-}(i,u) < \delta\}$$

where $\gamma$ and $\delta$ are acceptation and veto thresholds chosen in the unit interval.

If we want to assess the expected relevance of these pre-selected items, and possibly to rank the different items from the most relevant to the worst, we can define an overall confidence level attached to the recommendation of $i$ to user $u$ by balancing the positive and negative arguments represented by $\hat{P}^{+}(i,u)$ and $\hat{P}^{-}(i,u)$. Clearly, from a logical point of view, $\hat{P}$ must be linked to $\hat{P}^{+}$ and $\hat{P}^{-}$ by the following equation:

$$\hat{P}(i,u) \equiv (\hat{P}^{+}(i,u) \wedge \neg \hat{P}^{-}(i,u)) \tag{6}$$

The numerical counterpart of this condition is of the form:

$$\hat{p}(i,u) = T(\hat{p}^{+}(i,u), 1 - \hat{p}^{-}(i,u))$$

where $T(x,y)$ is a t-norm (e.g. $\min(x,y)$, $x.y$ , $\max(x+y-1,0)$).

Moreover, one can interpret the quantity $\min\{\hat{p}^{+}(i,u), \hat{p}^{-}(i,u)\}$ as the level of conflict between positive and negative arguments. Such an indicator may be useful to explain a recommendation. It can help to produce moderating arguments in the justification of the recommendation.

This procedure is clearly more discriminating that the previous one based on (3) and classical logic. However, our experiments have shown that such a procedure progressively looses its discriminating power as the number of graded items increases. This can be explained very simply. On the one hand, due to equation (5), the existence of an item $j$ in the close neighborhood of $i$ having received a very good grade (i.e. such that $p^+(i,u)$ is close to 1) is sufficient to rise $\hat{p}^+(i,u)$ close to 1. One the other hand, the existence of an item $k$ in the close neighborhood of $i$ having received a very bad grade (i.e. such that $p^-(i,u)$ is close to 1) is sufficient to rise $\hat{p}^-(i,u)$ close to 1. As soon as such $j$ and $k$ co-exist in the neighborhood of $i$, the positive and negative arguments neutralise each other, thus forbidding the recommendation of $i$. The situation remains the same when multiple positive neighbors like $j$ are present around $i$, even if $k$ remains the single negative counterexample.

In some applications, this particular feature may be seen as a drawback. One should prefer to "count" positive and negative arguments within the neighborhood and to compare the importance of the coalitions. In such a case, equation $\hat{p}(i,u)$ can be replaced by:

$$\hat{p}(i,u) = \frac{\sum_{j \in N_k(i)} s(i,j)\, p(j,u)}{\sum_{j \in N_k(i)} s(i,j)} \tag{7}$$

where $p(j,u)$ stands for $p^+(j,u) - p^-(j,u)$, or possibly another quasi-linear mean. In some cases, it might be interesting to discount $\hat{p}(i,u)$ when $N_k(i)$ too small ($N_k(i)$ contains at most $k$ elements but might also be empty). For example, one can bound $\hat{p}(i,u)$ to a certain threshold whose value possibly increases with the size of $N_k(i)$. The numerical rule given in (7) defines $\hat{p}(i,u)$ as the aggregate of positive and negative arguments evaluated independently and represented by $p^+$ and $p^-$. It this respect, it can be compared to the inference model used in the MYCIN system [43]. Such an additive rule is also close to those proposed in case-based decision theory [27], the main difference being that we restrict the set of possible influent "cases" to a subset of $k$-neighbors.

## 2.3 The individual-oriented, collaborative approach

Collaborative filtering methods recommend an item $i$ to an user $u$ based on aggregated user ratings of this item by like-minded users. The characterization of like-minded users (the $R$ relation) relies on the similarity of their preferences $P$ and possibly the usefulness of their recommendation ($Q$ relation). Let us mention that a more general view (taken by Bilsus and Pazzani [4]) is to consider that not only like-minded users are useful but any user whose ratings are somehow correlated to the active user. For the sake of simplicity and efficiency, we only consider here recommendation principles based on like-minded users. The collaborative approach simply defines relation $\hat{P}_c$ as the composition of fuzzy relations $P$ and $R$:

$$\hat{P}_c = P \cdot R \tag{8}$$

This can be explained by the following simple principle:

*Item $i$ is recommended to user $u$ if it is declared as attractive by another user $v$ whose value system is similar to $u$*

Following this principle, we ought to recommend an item $i$ to user $u$ if and only if $i$ was not graded by $u$ and $iPvRu$ for some $v$ different from $u$. This can be formalized by the following logical condition:

$$\hat{P}_c(i,u) \ \equiv \ [\neg(i \in G(u)) \ \wedge \ (\exists v \in U \setminus \{u\}, (P(i,v) \wedge R(v,u)))]$$

**Example.** In the case depicted on Figure 1, relation $R$ is represented by double arrows and the collaborative $\hat{P}_c$ by dashed lines. Assuming that $z_{2b}$ is a very positive grade (close to $z^*$), we have $bP.R1$ due to the $PR-$Path: $(b,2),(2,1)$. Considering this path, $b$ ought to be recommended to user 1 because it is well graded by user 2 which is close to him. Similarly, assuming that $z_{3c}$ is a very negative grade (close to $z_*$), $c$ seems inadvisable to 2 due to the $PR-$Path: $(c,3),(3,2)$.

Following the approach used in the non-collaborative case, we have to distinguish between positive and negative arguments in the collaborative filtering. They are represented by the two following conditions, for $\sigma \in \{-,+\}$:

$$\hat{P}_c^\sigma(i,u) \ \equiv \ [\neg(i \in G(u)) \ \wedge \ (\exists v \in U \setminus \{v\}, (P^\sigma(i,v) \wedge R(v,u)))] \tag{9}$$

The classical numerical counterpart of equation (9) is therefore:

$$\hat{p}_c^\sigma(i,u) = \bigvee_{v \in U \setminus \{u\}} T(p^\sigma(i,v), r(v,u)) \tag{10}$$

Similarly, following what was done in the non-collaborative case, we define $N_k(u)$ as the set of the $k$ more ressemblant users around $u$ (*i.e.* users $v \in U \setminus \{u\}$ having the $k$ greatest values $r(u,v)$). Then, we restrict the previous definition to the $k$-nearest neighbors by setting:

$$\hat{p}_c^\sigma(i,u) = \bigvee_{v \in N_k(u)} T(\hat{p}^\sigma(i,v), r(v,u)) \tag{11}$$

When the information about the usefulness of collaborative recommendations is available (under the form of the $Q$ relation), equations (9) and (11) can be sophisticated and rewritten as follows:

$$\hat{P}_c^{\sigma}(i,u) \equiv [\neg(i \in G(u)) \wedge (\exists v \in U \setminus \{v\}, (Q(i,v) \wedge P^{\sigma}(i,v) \wedge R(v,u)))]$$

and therefore we have:

$$\hat{p}_c^{\sigma}(i,u) = \bigvee_{j \in N_k(u)} T(q(i,v), \hat{p}^{\sigma}(i,v), r(v,u)) \tag{12}$$

Hence, a first collaborative selection of relevant items for users $u$ is given by:

$$\hat{P}_c(u) = \{i \in I \setminus \{u\} : \hat{p}_c^{+}(i,u) \geq \gamma \text{ and } \hat{p}_c^{-}(i,u) < \delta\}$$

where $\gamma$ and $\delta$ are acceptation and veto thresholds chosen in the unit interval.

In order to assess the expected relevance of these pre-selected items using a collaborative principle, and to rank them by decreasing order of relevance, we define the confidence level attached to a recommendation of $i$ for user $u$ by the following equation:

$$\hat{P}_c(i,u) \equiv (\hat{P}_c^{+}(i,u) \wedge \neg \hat{P}_c^{-}(i,u)) \tag{13}$$

The numerical counterpart of this condition is of the form:

$$\hat{p}_c(i,u) = T(\hat{p}_c^{+}(i,u), 1 - \hat{p}_c^{-}(i,u))$$

where $T$ is a t-norm. If one prefers to "count" positive and negative arguments within the neighborhood and to compare the importance of the two conflicting coalitions, the previous equation can be replaced by:

$$\hat{p}_c(i,u) = \frac{\sum_{v \in N_k(u)} r(v,u)\, p(i,v)}{\sum_{v \in N_k(u)} r(v,u)} \tag{14}$$

where $p(i,u)$ stands for $p^{+}(i,u) - p^{-}(i,u)$. In some cases, it might be interesting to discount $\hat{p}_c(i,u)$ when the size of $N_k(u)$ is too small.

## 2.4 Integrating content-based and collaborative filtering

The two families of approaches presented in the previous section have each known limitations when used independently of one another in the context of information filtering. An obvious limitation of purely content-based filtering arises when items are not sufficiently well described to allow the definition of a useful fuzzy similarity $S$ to be computed. In the worst case, each item is solely described by a title and the sole similarity that can be drawn between the items is based on their title as strings. A second limitation that affects purely content-based filtering arises when the set of preferences $p(i,u)$ expressed by users $u$ for different items $i$ is low on average [3]. It is therefore difficult to make significant inferences on such small figures or to build standard profiling techniques. A third limitation of such an approach is on the expected preference relation $\hat{P}$. A purely content-based approach has the tendency to inhibit the system from recommending items $i$ that are "far" from whose already preferred. By construction, nearest-neighbour approaches do not suggest items that are not in the vicinity of existing preferred items.

On the other hand, several limitations of purely collaborative filtering that could only be partially predicted in theory were recently empirically confirmed. In particular, different metrics have been proposed to analyse the accuracy of collaborative filtering algorithms [10, 34]. A reference dataset used for this kind of study is the *EachMovie* dataset. This is an explicit voting dataset(users explicitely rate movies between 0 to 5) using example from a collaborative filtering site deployed by Digital Equipment from 1995 to 1996 [4]. The relationship between the accuracy of the prediction and the number of users in the data set for data extracted from the EachMovie database shed some light on this limitation. Between 2 and 100 users, the inaccuracy decreases linearly at a vey sharp rate. Then from 100 to 600 this rate is less sharp [13]. These results are not surprising as a pure collaborative learning relies on the fact that various users have rated *similar* items. It is clear that the less are the users the smaller are the chance that two users will have rated similar items. Ultimately, with a single user, collaborative filtering does not make any sense at all. The approach we advocate with others (see [11]) is to address the issues mentioned above to integrate both pure collaborative-filtering with content-based techniques. In the next section we describe the rationales for such an integration.

[3] As an example, the average number of movies rated in filmconseil is around 14 for a database that counts more than twenty thousands movies

[4] See http://www.research.digital.com/SRC/EachMovie/ for further information

# 3 RECOMMENDER SYSTEMS INTEGRATING PREFERENCE-BASED SEARCH AND MACHINE LEARNING

## 3.1 Rationales

This section describes several conditions on a CDS and argues why they are desirable. An architecture to adhere to these conditions is then presented. The way to adapt existing algorithms to support such architecture is finally given.

We argue that there are several conditions that are desirable when designing a recommender system or a collaborative decision support system. Such system ought to

- **scale up to users and items** in both response time and space requirements. A naive approach to build a CDS is to compute and store the whole similarity matrices that represent the $S$ similarity relations and the $R$ relations. Both the number of items $N_I$ and the number of users $N_U$ may be large (several thousands). Considering that a naive approach requires to store one matrix of size ${N_I}^2$ and one of size ${N_U}^2$, specific representation must be developed to enable a smooth scaling in both the number of items and users. This aspect is fundamental for an industrialization of the solution and is almost never detailed in collaborative filtering algorithms because of patents pending on recommender systems.

- **Adapt the integration** to the state of the system. The drawbacks of each method (content and collaborative based) taken individually are strongly related to the number of users and the availability of information to describe items. A CDS should therefore dynamically adapt its strategy. This aspect is important to maximize the chances that the system will provides adequate recommendations.

- **Explain its recommendations**. Although this aspect is not mandatory and many recommander systems do not include this dimension, we are interested in systems that are not only black-boxes but are able to somehow explain to a user the reason for a particular recommendation. This aspect is important for the user to build confidence in the system recommendation and better understand bad recommendations when they occur.

- **Motivate ratings of items**. In facts, one major problem of recommander systems is to have users express their preferences. To increase the number of rated items, one approach consists in forcing users to rate pre-definied items. A major drawback of this approach is that users may easily get tired of it [5]. A second approach consists in using user navigation (this is called *implicit* rating [10]) to automatically rate items. The drawback of this second method lies in its highly speculative approach of ranking. We support a third approach that consists in *motivating* the user to express its preferences that we call *active rating*.

## 3.2 The construction of fuzzy relations $S$ and $R$

We briefly present the basic ideas used to construct the similarity relation over items and the resemblance relation over attributes. We will address computational problems linked to the storage and update of these relations in the next subsection.

### 3.2.1 Multiattribute comparison of items

Relation $S$ over a set of items is defined from decription vectors of type $(d_1(i), \ldots, d_n(i))$. For each attribute $X_j, j = 1, \ldots, n$, we construct a one-dimensional similarity relation $S_j$ defined on $I \times I$. Due to a lack of space a comprehensive presentation of the various techniques that can be used for the construction of $S_j$ cannot be detailed here. The construction depends on the nature of the attribute. As an illustration, consider the 2 following simple cases:

- Attributes valued on an ordered numerical scale:

$$S_j(i,k) = \delta(|d_j(i) - d_j(k)|)$$

  where $\delta$ is a non-increasing function valued into $[0, 1]$ and such that $\delta(0) = 1$. The determination of the $\delta$ function is performed using

[5] In Firefly [46] for example, a new user needs to choose to ratea minimum number of movies amongst predefined sets. If ever you have not seen many movies you end up scanning dozens of screens before finding one you can actually rate.

classical methods based on the definition of both indifference and preference thresholds for each attribute (see [38, 36, 37]).

- Attributes valued on a nominal scale:

$$S_j(i,k) = \begin{cases} 1 & \text{if } d_j(i) = d_j(k) \\ 0 & \text{otherwise.} \end{cases}$$

Then the overall fuzzy similarity over items is defined on $I \times I$ by:

$$S(i,k) = \psi(S_1(i,k), \ldots, S_n(i,k))$$

where $\psi$ is a weighted compromise operator, *e.g.* a weighted quasi-linear mean :

$$\psi(x_1, \ldots, x_n) = \phi^{-1}\left(\sum_{i=1}^{n} \omega_i \phi(x_i)\right)$$

where $\phi$ is a continuous strictly monotonic function on the unit interval and $w_j, j = 1, \ldots, n$ are factors weighting the relative importance of each attribute. Such a construction leads to a fully compensatory measure of similarity between items. If necessary, this construction can be sophisticated in such a way that attributes have a veto. This yields a non-compensatory aggregation procedure making items totally non-similar as soon as they are strongly different on a given significant attribute (for more details see [36, 38]).

### 3.2.2 Resemblance and influence between users

Following the approach adopted for the comparison of items, the resemblance between individuals could also be based on a multiattribute profile representing the individual. Notice however that in most cases, the collaborative decision support system, to be successful, must allow the users to keep anonymity. Thus, the only available information to be used in the production of relevant recommendations is the set of grades given to items by each user. They reflect the value system of each user and allow the preference profiles of users to be compared. For every pair of users $(u, v)$, the comparison of their preference profiles can be made by investigating the grades of the items both graded by $u$ and $v$, in other words items belonging to $G(u,v) = G(u) \cap G(v)$. More precisely, within this set of items, the preference profile of each user

$u \in U$ is characterized by membership functions $p^+(., u)$ and $p^-(., u)$ characterizing the fuzzy set of items he likes and the set of items he does not like. Such fonctions are useful to evaluate the resemblance $r(u, v)$ and the extent to which user $u$ can influence an individual $v$. For the sake of illustration, we define two fuzzy influence relations over individuals.

- *positive influence:* we say that $u$ has a positive influence over $v$ when $I^+(u, v)$ holds: $I^+(u, v) \equiv [\forall\, i \in G(u, v),\ \ ((P^+(i, u) \Rightarrow P^+(i, v)) \wedge (P^-(i, u) \Rightarrow P^-(i, v))]$
- *symmetric influence :* we say that $u$ has a symmetric influence over $v$ when $I^=(u, v)$ holds: $I^=(u, v) \equiv [\forall\, i \in G(u, v),\ \ ((P^+(i, u) \Leftrightarrow P^+(i, v)) \wedge (P^-(i, u) \Leftrightarrow P^-(i, v))]$

Because $P^+$ and $P^-$ are fuzzy relations, the above logical equations must be interpreted in a [0, 1]-valued logic, using t-norms, fuzzy implication and equivalence operators (see [26, 7]). Both relations $I^=$ and $I^+$ can be used as the $R$ relation. A frequent alternative choice is to directly define $R$ as a decreasing function of the distance between vectors $(z_{iu}, i \in G(u, v))$ and $(z_{iv}, i \in G(u, v))$ (see e.g. [10]).

### 3.3 Hybrid Filtering to improve Recommendations

The general recommendation system we are proposing can be seen as an hybrid filtering process aggregating the outputs of the content-oriented filtering process with those obtained by collaborative filtering. The final rating $r_{iu}$ assessed for user $u$ for each pre-selected item $i$ is defined by:

$$r_{iu} = (1 - \beta)\hat{p}(i, u) + \beta\hat{p}_c(i, u) \tag{15}$$

where $\beta \in [0, 1]$ represents the weight of the other users in the recommendation. We set $\beta = 0$ at the beginning of the decision support process, and then $\beta$ is increased as the number of users involved in the system increases. Using these ratings, items can be ordered by decreasing order of relevance for a given user $u$ and the pairs $(i, r_{iu})$ can be presented to user $u$.

One major feature of our approach is that the computation of $r_{iu}$ results from the numerical interpretation of simple logical conditions reflecting very natural principles. Indeed, each computation step can be explained by a simple sentence following the SP and PR paths used to produce a content-based and/or a collaborative advice. Coming back to the initial example pictured on Figure 1, user 2 would get very easily explanations of type:

"*You should like $d$ because it is very similar to $b$ but you should not like $c$ because individual 3 which is close to you dislikes $d$*"

The argumentation is even more convincing when several positive or negative examples can be mentioned. To go further, it is also possible to explain to any user $u$, on request, the reasons why an item is assessed as similar as another, which are the users having the more influence on his recommendations, and why the system considers them as influent. This might be an invitation to provide the system with additional preference information, with the hope to get more informed recommendations in return.

As a complement, the user might be allowed to exclude some people considered as influent (on him) from his neighborhood and to include some other which have not yet been detected. Another useful option which can be simply implemented using our basic algorithms is partisan filtering. The idea of partisan filtering is to restrict a priori the population of possible advisors to a given subset of individuals (e.g. representing a given opinion pool) and to constrain the collaborative algorithm to search possible neighbors within this subset.

The major difficulty in developing such a system is to preserve a high level of interaction between users and the recommander system as the number of users and items increases. The interaction results here from the alternance of dialog stages (aiming at collecting preference information) with computational stages (aiming at producing recommendations). To get a successful system, the computational efficiency of filtering algorithms is one of the key issues (the other being to produce userfriendly interfaces). This point is particularly crucial for collaborative filtering methods because the resemblance $R$ between users must be revised each time a new grade is entered by some user. Moreover, even if the periodic addition of new items does not modify the known part of the similarity matrix $S$, it may have an impact on the definition of neighborhoods of type $N_k(i)$ and therefore on the final ratings of type $r_{iu}$. For this reason, we devoted a particular attention to the control of the computational times. This control is achieved by the following parameters :

- $k$ : the size of neighborhoods of type $N_k(i), i \in I$
- $k'$ : the size of neighborhoods of type $N_{k'}(u), u \in U$

- $\alpha_S$ : a threshold used to cut all SP-paths containing a too weak similarity $(s(i,j) < \alpha_S)$
- $\alpha_R$ : a threshold used to cut all PR-paths containing a too weak ressemblance $(r(i,u) < \alpha_R)$
- $\alpha_P$ : a threshold used to cut all SP-paths or PR-paths containing a too weak preference $(p(i,u) < \alpha_P)$

Moreover, since we restrict the analysis to neighborhoods of size $k$, only a border of size $k$ must be stored in matrices $S$ and $P$. Thus, $S$ must be organized in such a way that the $i^{th}$ column contains only the elements $j$ of $N_k(i)$ sorted by decreasing order of similarity $s(i,j)$. Similarly, $R$ must be organized in such a way that the $u^{th}$ row contains only the elements $v$ of $N_k(u)$ sorted by decreasing order of resemblance $r(u,v)$. The sorting of these rows and columns facilitates both access to similarity indices for computations and update of these indices when necessary.

## 3.4 Machine Learning to improve recommendation

The traditional task studied in machine learning is to classify as accurately as possible unseen instances based on a set of pre-classified examples [18]. In the context of collaborative filtering, learning to *order* items is a priori more desirable. In such a context, a meaningful task could be to learn a function that ranks all the items like they would be ranked by the user. This task has yet only received a very little attention in the Machine Learning community. Recently Cohen, Shapire and Singer have proposed a two-stage approach to learn to order items [14].

However such an approach does not apply when the number of items ranked by the user is an order of magnitude smaller than the items of the database. There are, for example, about 100.000 movies in the international movie database (IMDB) and an average user is not likely to rank more than one hundred movies which represent less than $0.1\%$ of the dataset. In other words, unless a user is willing to rate thousands of items, such learning task has very little chance to produce any useful result to the recommandation system. In fact, the lazy learning approach that focuses on neighborhood of previously ranked items is known to be more adequate in such context.

We advocate the use of Machine Learning to support the fourth rationale (see Section 3.1), i.e. to motivate the active user to express his preferences.

To do so, we suggest a two-stage approach. In a first stage the system builds an intelligible representation of the user's taste. In a second stage the system makes the active user evaluate whether this acquired knowledge is accurate or not. This approach lies inbetween *active learning* and *learning apprentice* approaches. An active learning problem is one "where the learner has the ability or needs to influence or select its own training data" [15]. In our case, the system always needs more preferences (i.e. data) and will influence the user to be motivated to do so. This approach also pertain to learning apprentices systems where the user interactively teaches the system [31]. Informally, our goal is to have the system build on the fly, a classifier based on the latest positive $P^+$ and negative preferences $P^-$ and to submit it to the active user. Indeed, like in learning apprentice systems, the idea is that when a user disagrees with some of the induced rules made by the system, the user will be more willing to provide the system with counter-examples.

The learning task we focus on is therefore to learn a classifier that based on already ranked items is able to classify any item into the category "rather to be liked" ($P^+$) or "rather to be disliked" ($P^-$). To illustrate the principle of such a process in a movie recommendation context, let us consider a user $u$ that has given high rates to movies A and B both directed by Orson Welles and a lower rate to a movie C directed by Brian de Palma. The system may come up with a classifier that says that the "Director" is critical in the attribution of rate of user $u$ and if the movie is from Brian de Palma, the movie is rather disliked. The user may find this model of himself wrong and react by giving a counter-example that contradicts this rule (e.g. a movie D from B. De Palma liked by $u$) and allows the system to get more ranks and more discriminant preferences about the user.

In a machine learning context what makes the specificity of this task is:

- The number of items rated per user has a low average and a high standard deviation: for each user the confidence in the results will be dramatically different

- Many users rate a very limited number of items (from one to a few hundreds), which requires a special treatment (bayesian approach are often not applicable). The system has to propose an hypothesis even with two ranked items.

- The results of the classifier ought to be comprehensible and concise so that the user can react

- Predictive accuracy is less important than speed since a model ought to be generated on the fly after each newly rated item.

## 3.5 A learning algorithm for active rating

To learn from examples of "rather liked" and "rather disliked" items, given the constraint presented above, top-down algorithms for decision tree learning were chosen. The main reason is that they provide comprehensive models (decision trees) and enjoy good performances. They are employed in the widely used software packages C4.5 and CART. In these algorithms, the function used to select the best attribute to split a given node (also called the splitting function) plays a key role in the performance of the decision tree algorithms. Intuitively, the role of this function is to value each potential split (e.g. Actor="Daroussin", Director="Spielberg", ProductionDate > 1990, etc.) to decide which node will be split and which function will be used to label the split ("rather liked" $P^+$ or "rather disliked" $P^-$).

Two of the most famous splitting function are the *Gini criterion* used in CART: $G(q) = 4q(1-q)$ and the *entropy function* used in C4.5 $G(q) = H(q) = -q\log(q) - (1-q)\log(1-q)$. In these functions, $q$ stands for the proportion of positive or negative examples (e.g. proportion of "rather liked" or "rather disliked" movies). Recently, Kearns and Mansour have shown that if these popular and empirically successful functions were so efficient resulted from the fact that CART and C4.5 were *boosting* algorithms [32]. They proposed a new criterion that we shall call the boosting criterion hereafter: $B(q) = 2\sqrt{q(1-q)}$.

Kearns and Mansour [32] have shown that the boosting criterion is the only one to enjoy strong concavity around zero. This strong concavity is linked to the a priori estimate accuracy of the tree generated using a splitting criteria. Based on this boosting criterion we have proposed a simple algorithm for learning a decision tree that classifies any item into the "rather like" or "rather dislike" class. To take into account the fuzzy nature of the preference relation we use a meta-learning approach similar in principle to that used in stratification to deal with skewed distribution [12] or in METACOST to make a machine learning algorithm cost-sensitive [19]. The intuitive idea is to automatically replicate particular examples (up-sampling) depending on their classification cost. In our *Amplification* algorithm (see Figure 2), the farther the preference from the neutral grade $z_u^0$), the greater the cost of misclassification of its associated example. To model this cost, given an active user $u$, for each item $i$ for which $u$ has given a preference a *number* $Rep_{i,u}$ of identical examples (a pair made of its item $i$ and its label $P^+$ or $P^-$) are produced. $Rep_{i,u}$ corresponds to the classification cost mentioned above. The learning algorithms based on this process is given in pseudo code in Figure 2.

**LearnDecisionTree($u$,$I$)**
% Learn a decision tree for user $u$ given
% his/her list of graded items $I$
**1. Compute** on the fly the neutral grade $z_u^0$
% This value correspond to the mean of the grades
**2. For each** graded item $i$ by the user $u$
**2.1** **Assign** label($i$, $u$)=sign(p($i$, $u$)-$z_u^0$)
% Each item $i$ above average is labeled positive and negative otherwise
% This labeled item becomes an "example" of the concept to learn
**2.2** **Create** $Rep_{i,u}$= mod(p($i$, $u$)-$z_u^0$,$redfactor$) examples of label($i$, $u$)
% To take into account the quantitative value of the graded item $i$,
% each example is replicated depending of its distance to the neutral
% grade. The effect of such replication is to drastically increase
% the misclassification cost of strongly preferred items.
% The $redfactor$ is a parameter that is initialized to 1,
% it increases with the total number of preferences of user $u$.
**3. Build** *decision tree* DTree($u$,$I$) using the boosting criterion $B(q)$
% The total number of reformulated examples may be approximated by
% the number of graded items times the standard deviation of their values
**4. Return** DTree($u$,$I$)
% Given the limited number of graded items, the tree is not pruned.
% Each leave of the tree is pure in the sense that it contains
% only positive or negative examples (see Figure 9).

Figure 2: The proposed Decision Tree Algorithm called *Amplification*

Let us give a brief example of the replication process. Here is a set of three items (here movies) for which a preference has been given:

| Title | Type | Origin | Rank | $Rep_{i,u}$ |
|---|---|---|---|---|
| UN AIR DE FAMILLE | Comédie | français | 82 | 21 |
| TALONS AIGUILLES | Comédie | espagnol | 64 | 3 |
| DANSE AVEC LES LOUPS | Western | américain | 39 | 22 |

Given that the neutral grade $z_u^0$ is 61.7 the two movies of the above tabular will be labeled positive and the last one negative. There will be a total of forty six examples after replication. The decision tree learning algorithm will be therefore biased to give less importance to the second movie which is slightly above the neutral grade. Based on the decision tree learned, an active learning algorithm is presented in Figure 3.

**ActiveLearning($u$,$I$)**
1. **Compute** on the fly DTree($u$,$I$)= **LearnDecisionTree($u$,$I$)**
    % Learn a decision tree for user $u$ given its list of graded items $I$
2. **Display** DTree($u$,$I$) to the user
    % Each leaf of the tree may be selected by the user when
    % s/he finds it erroneously representing her/his appreciations.
2.1 **Get** the decision tree node $N$ selected by the user.
    % This node $N$ corresponds to the one the user disagree with
2.2 **Build** the SQL query $Q$ corresponding to the tree node $N$
    % This query $Q$ corresponds to the Boolean expression defining the node $N$
2.3 **Perform** a dataset query $Q$ on the entire database of items
    and **build** the answered subset $S$
2.4 **Remove** from $S$ items already graded by the user
2.5 **Rank** the items $S$ using collaborative filtering so that
    the items most subject to contradict the label of $N$ are first
    % Using the recommendation algorithm allows to filter the items
    % so as to ease the interaction
2.6. **Display** the list of graded items of $S$
2.7. **Request** from the user an item that most contradicts label of $N$
3. **Return** to 1

Figure 3: The proposed Active Learning Algorithm

The decision tree learnt from the rates given by the user may be seen as a summary (a generalization) of its rates[6]. As such it may not only be used to make the user actively rate new items to contradict an erroneous rules learnt. It may also be used to predict whether an item will be liked or not by the user and explain why.

[6]In order to evaluate the predictive accuracy of the decision tree learned by the system, a leave-one out validation is available upon user request.

## 3.6 Related Work

As mentioned in the previous section, in the context of collaborative decision system, learning to *order* items is a priori more desirable. Recently Cohen, Shapire and Singer have proposed a two-stage approach to learn to order things [14]. In a first stage, their algorithm learns an approximation of a binary preference function between pairs of items. In a second stage they compute an ordered list that heuristically attempts to minimize disagreement with the learnt preference function [14]. In effect, minimizing the disagreement with a preference function has been proved to be NP-complete. The difference with our approach is that learning is used in their case as a substitute of traditional collaborative filtering.

Bilsus and Pazzani have recently proposed a framework for reformulating the problem of collaborative filtering as a standard machine learning algorithm [4]. The ranked values may be considered as unordered multiple values and a standard multi-class classification algorithm may be used. An even stronger assumption is to define one threshold in the ranked things and distinguish between things that are liked and others that are disliked.

Claypool et al. [13] has proposed a reinforcement learning approach to learn, for each user, the parameter $\beta_u$ used to aggregate content-based and collaborative filtering algorithm (see Equation (15)). The basic idea of his algorithm is to update the balance between content-based and collaborative filtering whenever a new preference $p(i,u)$ on a item $i$ is given to the system. The update is meant to minimize the prediction error $r_{iu} - p(i,u)$ on item $i$. To do so the system compares $p(i,u)$ with $\hat{p}(i,u)$ and $\hat{p}_c(i,u)$. The parameter $\beta_u$ is modified accordingly. A limitation of such an algorithm lies in the fact that it depends on the order of presentation of the ranked items. Moreover it performs a greedy optimization of the weighted prediction function. An approach to avoid this *catastrophic forgetting* problem would require several learning passes (in the same way neural networks are trained). However, this latter solution would be at the expense of efficiency.

# 4 Designing a system for movie- recommendations: Film Conseil

## 4.1 Introduction to Recommender Systems for Movies

To experiment our approach to collaborative decision making, we have developed a movie-recommendation system called "Film Conseil", accessible from the web [7]. There exists today several commercial products that propose collaborative filtering for movies or music: MovieLens [8], firefly [9], Movie Critic [10], etc. The movie-recommendation task has one main practical advantage: it is relatively easy to attract users to give holistic judgments about movies and thus their preferences.

The originality of our system is manyfold:

1. the hybrid nature of the recommendation that may be controlled by the user,
2. the explanation given by the system on its recommendation (be it content-based, collaborative or hybrid),
3. the active learning motivation for ranking movies,
4. the systems' ability to provide the user with his/her best and worst friends.

The following subsections present the main features of the "Film Conseil" system.

## 4.2 The Representation of users and movies in "Film Conseil"

In the system "Film Conseil", the items considered are movies. Each movie $m$ of the database that counts about 15000 different movies is described by a tuple of attribute values. The attributes used by the system are the ID of the movie (a key to index images), the title, the genre (western, comedy,

[7] http://www-poleia.lip6.fr/∼fconseil

[8] http://movielens.unm.edu

[9] http://www.firefly.com

[10] http://www.moviecritic.com

drama, ...), the origin (USA, India, England, France, ...), the duration, the year produced, the film maker and the first four main actors. Within the "Film conseil" website, each user is represented by his preference profile, *i.e.* the set of grades he has assigned to movies. The only information the user is supposed to provide is his/her pseudonyme. The fuzzy similarity between movies and the fuzzy influence relations linking users are computed periodically and stored in an independent database. Then, "Film conseil" uses the different filtering approaches introduced in the previous section so as to provide the user with an ordered list of recommendations. A level of confidence in the recommendation is associated to each recommended movie and is precisely used to rank them.

### 4.3 "Film Conseil" Sessions

To provide the reader with an illustrative example that one may easily reproduce we have considered a user that has expressed his preferences regarding only 8 movies. Among them, we have movies like "Le gout des autres", "Cuisine et Dépendances" and "On connaît la chanson" which have been well graded. From such an information, the various filtering process and learning algorithms are combined to provide the user with relevant recommendations. The general principle of the filmconseil recommender system is intuitively pictured on Figure (4).

The main stages of a typical session and the associated screens are presented in the appendix. One lesson we learnt from the industrial application of the "Film Conseil" system [30] is the gap between what we believe is important as researchers and what is perceived as crucial by end-users. The capacity of the system to provide each user with the list of its best friends (closest neighbors according to $R$) is a good illustration. The algorithms we develop to identify the best and worst friends of an active user are straightforward and at first, we did not consider it as a major functionality. The effective use of the system has demonstrated that in terms of interaction this feature was one of the most appealing. Indeed, we observed that many users immediately wanted to know their best friends after having graded new movies.

## 5 CONCLUSION AND PERSPECTIVES

Recommender systems address the general problem of recommending items to an active user relying on the implicit sharing of preferences and experi-

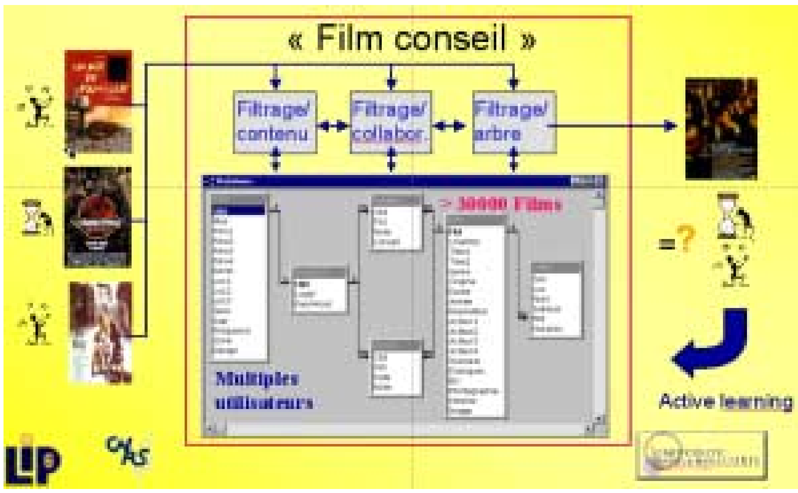


Figure 4: Intuitive principle of the filmconseil approach

ence with other users [41]. Both content-based (solely based on the active user preferences) and pure collaborative filtering (solely based on the other user preferences) approaches have proved to be inadequate in various real deployments [10, 13].

In this paper we propose an integration of this two recommendation principles as well as the use of active learning to stimulate the users to express their preferences. The first recommendation principle "content based" relies on fuzzy filtering methods that only use the active user fuzzy preference relations. The second one relies on fuzzy influence relations between users allowing an implicit cooperation between users. Both principles are weighted then summed to produce the final recommandation of items to the active user. The respective weights may be either fixed, chosen interactively by the user or dynamically computed from the global parameters of the system such as the number of users and the number of items. We advocate to weight more favorably content-based approach when the number of users is low and to increase the weight of collaborative thereafter.

To summarize, the first contribution of this paper is an approach to integrate two types of recommendation that play a complementary role to propose the most relevant recommendation. The second contribution is an active learning algorithm meant to motivate the user to grade items. This problem of motivation is often overlooked by researchers because its evaluation is hard to assess. Nevertheless the problem remains and the solution we proposed does stimulate. The third contribution is related to the explanation power given by the system.

As mentioned by Bilsus [4], they are many ways to evaluate the quality of a recommandation system. This paper does not intend to give an empirical evaluation of algorithms taken independently as it is done for several collaborative filtering algorithms. A partial evaluation of our architecture was done by the users of the industrial version of filmconseil freely available on the web.

One of the current limitation of our architecture concerns its ability to *explicitly* take into account feedback from users regarding recommendations that have been made to them. Suppose for example that a movie recommended to a user is later graded as irrelevant. Although the method, by construction, guarantees that the system will not make such a recommendation again, it means that the procedure *per se* ought to be adapted. For this reason we investigate the use of machine learning which among the other users are good advisors and bad advisors. In this case the feedback provided by the user is used to learn a degree of confidence in other users advises. In this respect, we currently investigate the use of multi-instance learning algorithms [45].

Finally, in order to increase the potential applications of our works, the development of tools able to extract the content of semi-structured documents is necessary. This is particularly true concerning textual documents which form a major part of the information available on the web. Actually, several tools concerning the semantic access to textual documents, the processing of open requests and those concerning the aggregation of experts are currently under development in our laboratory ("Web-conseil" project, [17, 16]). For the future, we are considering integrating these two techniques so as to provide our recommender system with new features.

## ACKNOWLEDGEMENTS

The authors express their deep thanks to Nicolas Bredeche that helped us designing and deploy the "Film Conseil" recommender system, and to Florence D'Alché-Buc and Patrick Gallinari for their fruitful cooperation in the

LIP6 project Web-conseil. We finally wish to thank Patrick Bosc and Henri Prade for their constructive remarks on a preliminary version of this paper.

## REFERENCES


[1] N. Belkin et W. Croft. Information filtering and information retrieval: Two sides of the same coin, 1992.

[2] N. D. Belnap. A useful four-valued logic. In G. Epstein et J. Dumm, diteurs, *Modern uses of multiple valued logics*, pages 8–37. D. Reidel, 1977.

[3] M. Béreau et B. Dubuisson. A fuzzy extended $k$-nearest neighbor rule. *Fuzzy Sets and Systems*, 44:17–32, 1991.

[4] D. Bilsus et M.J. Pazzani. Learning collaborative information filters. In *Fifteenth International Conference in Machine Learning*, pages 46–54, Bari, Italy, 1998. Morgan Kaufmann.

[5] P. Bosc, P. B. Buckles, F. E. Petry et O. Pivert. Fuzzy information systems. In J. C. Bezdek, D. Dubois et H. Prade, diteurs, *Fuzzy Sets in Approximate Reasoning and Information Systems*, Handbook of fuzzy sets series, pages 403–468. Kluwer Academic Publishers, 1999.

[6] P. Bosc et O. Pivert. Some approaches for relational databases flexible querying. *Journal of Intelligent Information Systems*, 3(4):323–354, 1992.

[7] B. Bouchon. *La logique Floue et ses Applications*. Addison Wesley, 1995.

[8] B. Bouchon-Meunier, D. Dubois, L. Godo et H. Prade. Fuzzy sets and possibility theory in approximate and plausible reasoning. In J. C. Bezdek, D. Dubois et H. Prade, diteurs, *Fuzzy Sets in Approximate REasoning and Information Systems*, Handbook of fuzzy sets series, pages 15–190. Kluwer Academic Publishers, 1999.

[9] B. Bouchon-Meunier et C. Marsala. Learning fuzzy decision rules. In J. C. Bezdek, D. Dubois et H. Prade, diteurs, *Fuzzy Sets in Approximate REasoning and Information Systems*, Handbook of fuzzy sets series, pages 199–304. Kluwer Academic Publishers, 1999.

[10] J. Breese, D. Heckerman et C. Kadie. Empirical analysisof predictive algorithms for collaborative filtering. In *Fourteenth Conference of Uncertainty in AI*, pages 43–52, Madison, WI, 1998. Morgan Kaufmann.

[11] R. Burke. Integrating knowledge-based and collaborative-filtering recommender systems. In *Workshop on AI and Electronic Commerce. AAAI 99*, Orlando, Florida Claypool, 1999.

[12] Ph. Chan et S. Stolfo. Toward scalable learning with non-uniform class and cost distribution: A case in credit card fraud detection. In *International Conference on Knowledge Discovery in Databases*, pages 164–173. AAAI Press, 1998.

[13] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes et M. Sartin. Combining content-based and collaborative filters in an online newspaper, 1999.

[14] W. W. Cohen, R. E. Shapire et Y.Singer. Learning to order things. In G. Tesauro, D. Touretzky et J. Alspector, diteurs, *Journal of Artificial Intelligence Research*, volume 10, pages 243–270. 1999.

[15] D. Cohn, Z. Ghahramani et M. Jordan. Active learning with statistical methods. In G. Tesauro, D. Touretzky et J. Alspector, diteurs, *Advances in Neural Information Processing Systems*, volume 7. MIT Press, Cambridge, 1995.

[16] F. d'Alché Buc et P. Perny. Agrégation de systèmes d'apprentissage. Communication aux Journées GDR-PRC ISIS-I3 "Fusion et Décision", Février 2000.

[17] F. d'Alché Buc, P. Perny, P. Gallinari et J.D. Zucker. Projet "WEB CONSEIL": Modélisation des préférences et recherche d'information pour le conseil aux utilisateurs dans un contexte distribué. Technical report.

[18] T. Dietterich. Inductive learning from preclassified training examples. In *Readings in Machine Learning*, pages 45–56. Morgan Kaufmann, 1990.

[19] P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Knowledge Discovery and Data Mining*, 1999.

[20] D. Dubois, D. Le Berre, H. Prade et H. Zapico. Making decision in a qualitative setting: From decision under uncertainty to case-based decision. In Ted Senator et Bruce Buchanan, diteurs, *Proceedings of the Fourteenth National Conference on Artificial Intelligence and the Ninth Innovative Applications of Artificial Intelligence Conference*, pages 588–593, Menlo Park, California, 1998. AAAI Press.

[21] D. Dubois, L. Godo, H. Prade et H. Zapico. On the possibilistic decision model : from decision under uncertainty to case-based decision. *Journal of Uncertainty, Fuzziness and Knowledge based decision*, 7(6):631670, 1999.

[22] D. Dubois, E. Hullermeier et H. Prade. Flexible control of case-based prediction in the framework of possibility theory. In E. Banzieri et L. Portinale, diteurs, *Proceedings of EWCBR 2000*, pages 61–73. Springer-Verlag, 2000.

[23] D. Dubois et H. Prade. *Fuzzy sets and systems : theory and application*. Academic Press, New-York, 1980.

[24] D. Dubois et H. Prade. *Théorie des possibilités : application à la représentation des connaissances en informatique*. Masson, Paris, 1985.

[25] D. Dubois et H. Prade. *Grands paradigmes du traitement intelligent et interactif de l'information*, dans Actes du PRC $I^3$. 1998.

[26] J. Fodor et M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision support*. Kluwer Academic Publishers, 1994.

[27] I. Gilboa et D. Schmeidler. Case-based decision theory. *The Quarterly Journal of Economics*, 110:605–639, 1995.

[28] M. Ginsberg. Multivalued logics: a uniform approach to reasoning in artificial intelligence. *Computational Intelligence*, 4(3):265316, 1988.

[29] L. Henriet et P. Perny. Méthodes multicritres non-compensatoires pour la classification floue d'objets. *Proceedings of LFA'96*, pages 9–15, December 4-5 1996.

[30] http://www-poleia.lip6.fr/$\sim$fconseil.

[31] J. Jourdan, L. Dent, J. McDermott, T. Mitchell et D. Zabowski. Interfaces that learn: A learning apprentice for calendar management, 1991.

[32] M. Kearns et Y. Mansour. On the boosting ability of top-down decision tree algorithms. In *STOC*, 1996.

[33] J.M. Keller, M.R. Gray et J.A. Givens. A fuzzy $k-$nn neighbor algorithm. *IEEE, Trans. Syst. Man. and Cyber.*, 15(4):580–585, 1985.

[34] B.P. Mathews et A. Diamantopoulos. Towards a taxonomy of forecast error measures. volume 13, pages 409–416. 1994.

[35] M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, 1983.

[36] P. Perny. Multicriteria filtering methods based on concordance and non-discordance principles. *Annals of Operations Research*, 80:137–165, 1998.

[37] P. Perny et M. Roubens. *Preference Modelling*, dans R. Slowinski (ed), D. Dubois et H. Prade (series eds), "Handbook of Fuzzy Sets and Possibility Theory, Operations Research and Statistics", chapitre 1, pages 3–30. Kluwer Academic Publishers, 1998.

[38] P. Perny et B. Roy. The use of fuzzy outranking relations in preference modelling. *Fuzzy Sets and Systems*, 49:33–53, 1992.

[39] P. Perny et J.D. Zucker. Collaborative filtering methods based on fuzzy preference relations. *Proceedings of EUROFUSE-SIC'99*, pages 279–285, May 25-28 1999.

[40] P. Resnick, N. Iacovou, M. Suchak, Bergstrom et J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994*, pages 175–186. Stanford Research Institute, 1994.

[41] P. Resnick et H. Varian. Recommender systems, guest editors special section: recommendation systems. volume 40, pages 56–58. 1997.

[42] U. Shardanand et P. Maes. Social information filtering: Algorithms for automating *word of mouth*. In *Human Factors in Computing Systems Proceedings of CHI95*, pages 210–217, 1995.

[43] E. H. Shortliffe et B. G. Buchanan. A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23:351–379, 1975.

[44] A Tsoukiàs et Ph. Vincke. A new axiomatic foundation of partial comparability. *Theory and Decision*, 19:19–114, 1995.

[45] J. Wang et J-D. Zucker. Solving multiple-instance problem: a lazy learning approach. In Pat Langley, diteur, *International Conference on Machine Learning*, Stanford, USA, 2000. Morgan Kaufmann Publishers.

[46] www.firefly.com.

## APPENDIX

This appendix presents some typical stages of a standard "Film-conseil" session illustrated by their associated screen-dump.

- THE NOTATION STAGE: For any movie chosen by the user, the descriptive information concerning this movie is given on the screen and the user is invited to enter a grade on a bipolar scale using a slider. This scale is continuous but some typical marks are represented along the scale, in an intuitive way so has to help him to locate some reference levels like "very bad", "bad", "neutral", "good", "very good" and to select the appropriate grade (see Figure 5).
- THE RECOMMENDATION STAGE: The user is invited to select a recommendation mode ("pure content-based", "pure collaborative", "hybrid") and then the top list of most promising selected movies appears on the screen, ranked by decreasing order of relevance (see Figures 6 and 7 ) .
- THE EXPLANATION STAGE: By clicking on any selected movie, the author is provided with first level explanations. In the case of content-oriented filtering as well as in the case of pure collaborative filtering, the user can obtain two levels of explanation. The first level of explanation is based on the detection of $SP-$paths in the relational structure we have defined in the paper. More precisely, in order to explain why an item $i$ is proposed to an user $u$, the list of the $k$-nearest neighbors of $i$ within the set $\{j \in G(u) : iSjPu\}$ is given. Upon request, the user has the possibility to get a more detailed explanation. At this level, the precise features in common between $i$ and $j$ are listed, and this for each item $j$ considered as closed to $i$ in the first level explanation.

  In the case of pure collaborative filtering, there exists also two levels of explanation. The first one is based on the detection of $PR-$paths in the relational structure we have defined in the paper. In order to explain why an item $i$ is proposed to user $u$ by collaboration, the list of the $k$-nearest neighbors of $u$ within the set $\{v \in I : iPvRu\}$ is given. Upon request, the user has the possibility to get a more detailed information, explaining why a user $v$ mentioned in the first-level explanation is seen as close to him.

  In the example given on Figure 8, the user has required an hybrid filtering and therefore received a twofold explanation based on content-based and collaborative arguments.

- THE ACTIVE LEARNING STAGE: At any time, after having at least graded two movies, the active user may visualize the profile that the system has inferred from her/his grades. The profile is a decision tree where each leaf defines a prediction whether a movie will be rather liked or disliked by the active user. This tree is learnt following the amplification algorithm described on Figure [**?**]. Based on this tree, a page is dynamically built using the active learning algorithm sketched on Figure [**?**]. Figure 9 presents a tree learnt from the same active user as above. This decision tree indicates that the presence of the actor called “Jean-Pierre Bacri” is a good predictor of the attractiveness of a movie for the active user. Another leaf states that when this actor is not in the cast, the ”origin” of the movie is in turn a good predictor. Whenever the user disagree with a rule that was inferred by the system, it may click on the hyperlink ”Pas d’accord” (i.e. Disagree in French) and a list of movies that have not been yet graded and might contradict the rule are proposed to the user.

  The (expert) user may also request a leave-one out cross-validation of the learning algorithm of the system on its grades. The small window on the bottom of Figure 9 indicates that the reliability of the model is 75%. In other words, the average error by learning a model from all the movies but one and testing the model on this one is 25%.

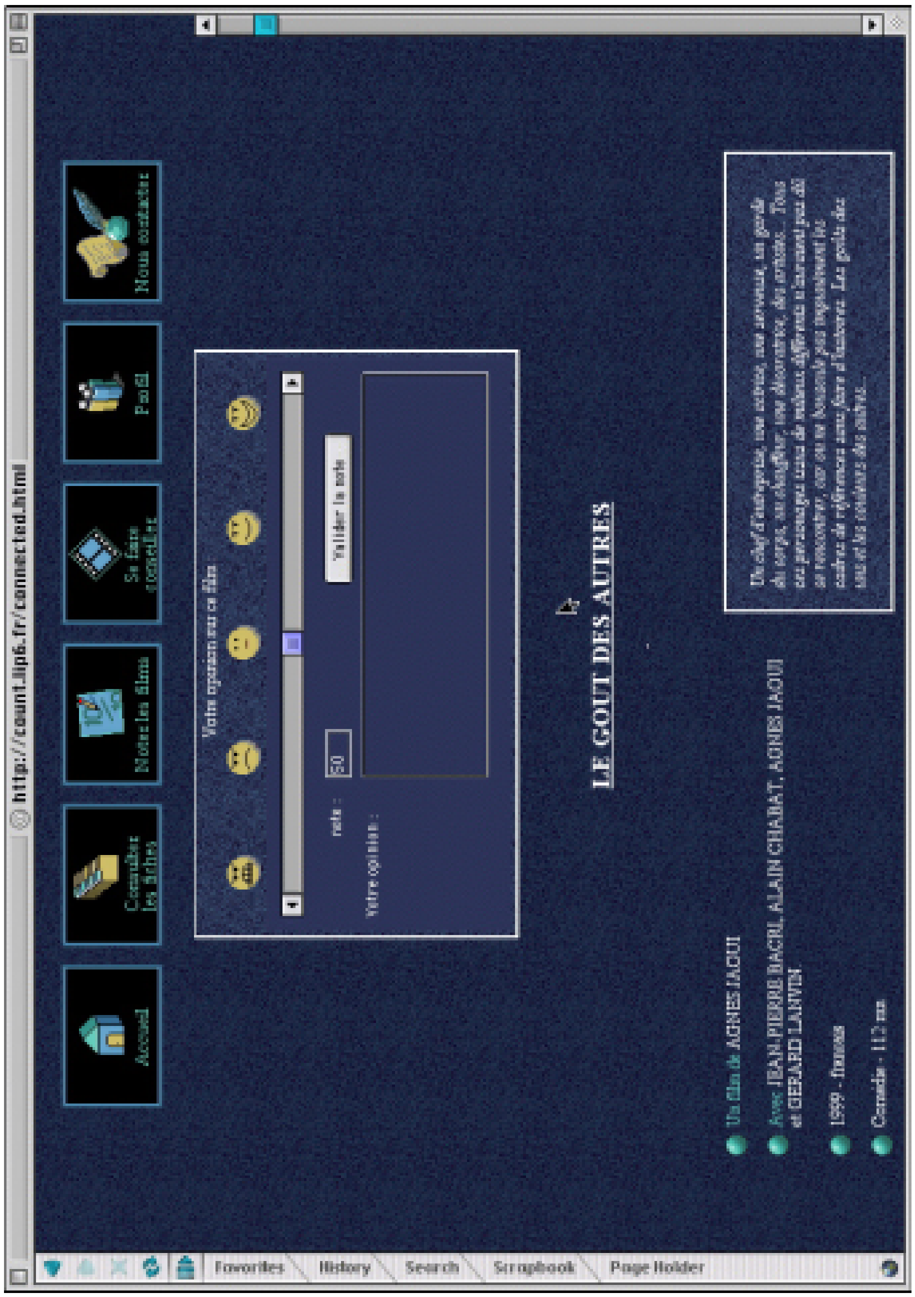


Figure 5: The notation screen


Figure 6: The selection of the filtering-mode

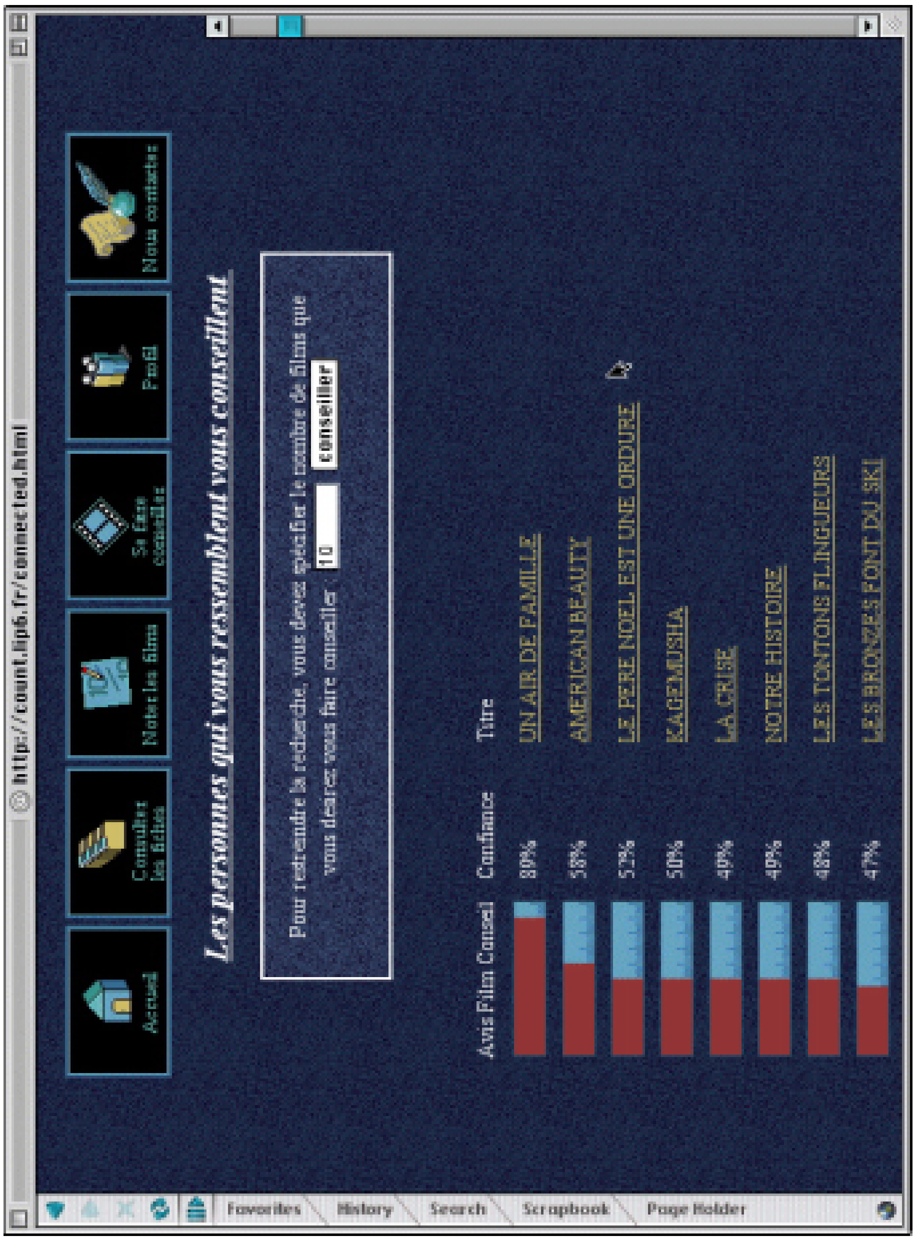


Figure 7: The selection obtained by collaborative filtering

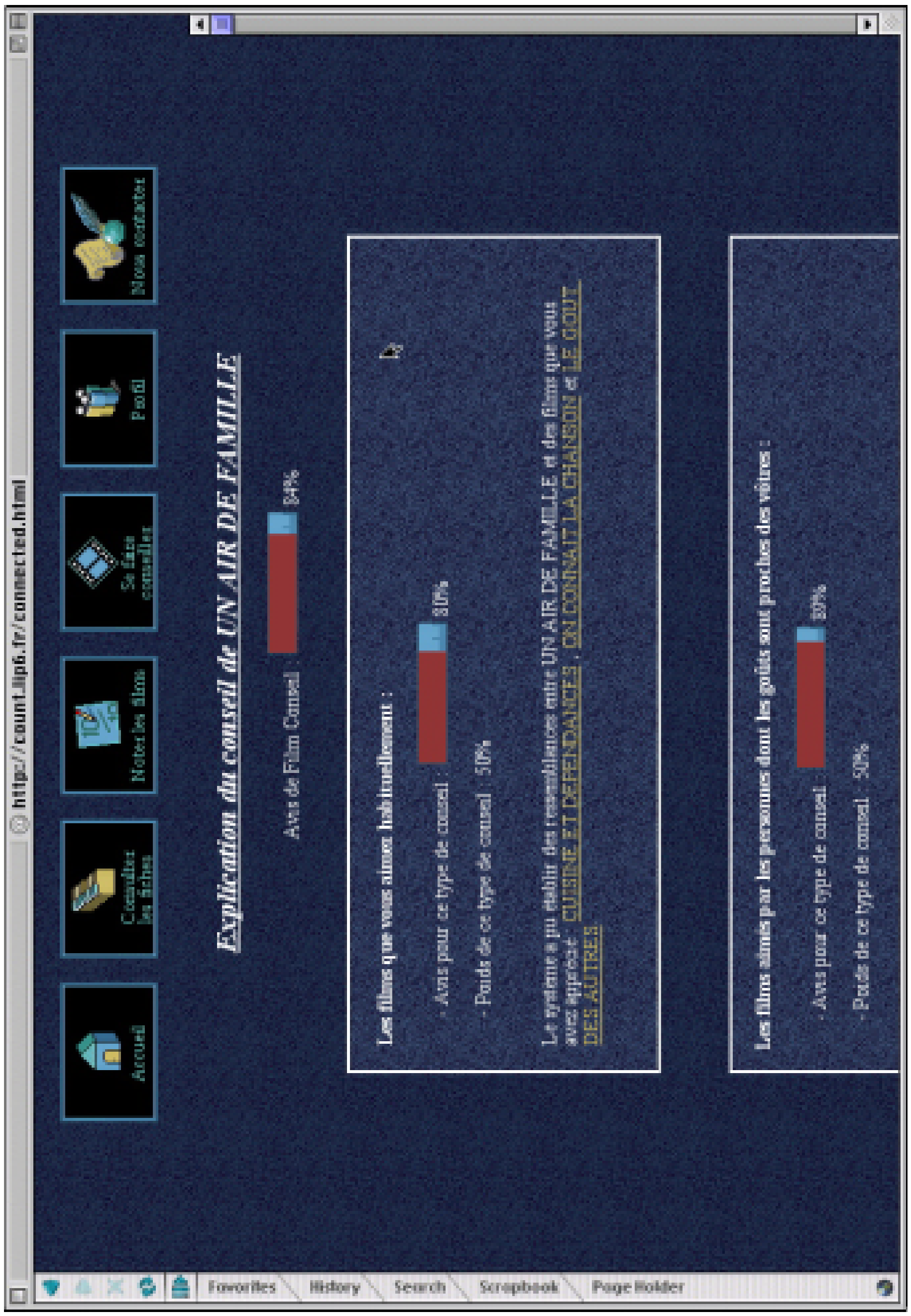


Figure 8: First-level explanations

Figure 9: Preference Profile and active learning