Chapter 15
Collaborative Filtering Based on Choosing a Different Number of Neighbors for Each User

Antonio Hernando, Jesús Bobadilla, and Francisco Serradilla

15.1 Introduction

We present here a new technique for making predictions on recommender systems based on collaborative filtering. The underlying idea is based on selecting a different number of neighbors for each user, instead of, as it is usually made, selecting always a constant number k of neighbors. In this way, we have improved significantly the accuracy of the recommender systems.

Recommender Systems are programs able to make recommendations to users about a set of articles or services they might be interested in. Such programs have become very popular due to the fast increase of Web 2.0 [11, 12, 15] and the explosion of available information on the Internet. Although Recommender Systems cover a wide variety of possible applications [3, 4, 16, 19, 21], Movie recommendation websites are probably the best-known example for common users and therefore they have been subject to significant research [2, 14].

Recommender Systems are based on a filtering technique trying to reduce the amount of information available to the user. So far, collaborative filtering is the most commonly used and studied technology [1, 5, 9] and thus judgment on the quality of a recommender system depends significantly on its Collaborative Filtering procedures [9]. The different methods on which Collaborative Filtering is based are typically classified as follows:

- Memory-based methods [13, 18, 20] use similarity metrics and act directly on the matrix that contains the ratings of all users who have expressed their preferences on the collaborative service; these metrics mathematically express a distance between two users on behalf of their respective ratings.
- Model-based methods [1] use the matrix with the users' ratings to create a model on which the sets of similar users will be established. Among the most widely used models of this kind we have: Bayesian classifiers [6], neural networks [10] and fuzzy systems [22].

A. Hernando (✉)
Depto. de Sistemas Inteligentes Aplicados, Universidad Politécnica de Madrid, Spain
e-mail: ahernando@eui.upm.es

B. Furht (ed.), *Handbook of Social Network Technologies and Applications*,
DOI 10.1007/978-1-4419-7142-5_15, © Springer Science+Business Media, LLC 2010

Generally speaking, commercial Recommender Systems use memory-based methods [8], whilst model-based methods are usually associated with research Recommender Systems. Regardless of the approach used in the Collaborative Filtering stage, the technical purpose generally pursued is to minimize the prediction errors, by making the accuracy [7,8,17] of the Recommender Systems as high as possible. This accuracy is usually measured by the mean absolute error (MAE) [1,9].

In this paper, we will focus, among the Memory-based methods, those which rely on the user-based nearest neighborhood algorithm [1,5]: The K most similar users to one given (active) user are selected on behalf of the coincidence ratio between their votes as registered within the database. In this paper, a variant of this algorithm is presented. This is based on choosing, for each different user, not a constant, but a variable number of neighbors. As we will see, our algorithm improves significantly the accuracy as compared to the typical user-based nearest neighborhood algorithm with a constant number of neighbor users.

In Sects. 15.2 and 15.3, we formalize some concepts on recommender systems and the memory-based methods of collaborative filtering. In Sects. 15.4 and 15.5, we present new techniques based on the idea of choosing a variable number of neighbors for each user. In Sect. 15.6, we discuss how our algorithm improves significantly the K-nearest neighborhood algorithm. Finally, in Sect. 15.7, we set our conclusions.

15.2 Recommender Systems

We will consider a recommender system based on a database consisting of a set of m users, $U = \{1,\ldots,m\}$, and a set of n items, $I = \{1,\ldots,n\}$ (in the case of a movie recommender system, U would stand for the database users registered in the system and I would refer to the different movies in the database).

Users rate those items they know with a discrete range of possible values $\{min, ..., max\}$, associating higher values to their favorite items. Typically, this range of values is $\{1,\ldots,5\}$ or $\{1,\ldots,10\}$.

Given a user $x \in U$ and a item $i \in I$, the expression $v(x,i)$, will represent the value with which the user x has rated the item i. Obviously, users may have not rated every item in I. We will use the symbol $\bullet$ to represent that a user has not made any rating concerning an item i. In this way, the possible values in the expression $v(x,i)$ is the set $V = \{min,\ldots,max\} \cup \{\bullet\}$.

In order to offer reliable suggestions, recommender systems try to make accurate predictions about how a user x would rate an item, i, which has not been rated yet by the user x (that is to say, that $v(x,i) = \bullet$). Given a user $x \in U$ and an item $i \in I$, we will use the expression $v^*(x,i)$, to denote the system's estimation of the value with which the user x is expected to rate the item i.

Different methods have been used so far in order to achieve good estimations on the users' preferences. The quality of these techniques is typically checked in an empirical way, by measuring two features of the recommender system:

- The error made in the predictions
- The amount of predictions that the system can make.

Regarding error made in the estimation, different measures have been proposed. The most used one is probably the MAE [9] (see Definition 15.1) which conveys the mean of the absolute difference between the real values rated by the users, $v(x,i)$, and the estimated values $v^*(x,i)$. As may be seen in Definition 15.1, in the hypothetical case that the recommender system cannot provide any estimation, then we would consider that $MAE = 0$.

Definition 15.1 (MAE). Let $J = \{(x,i)|x \in U\ i \in I\ v(x,i) \neq \bullet\ v^*(x,i) \neq \bullet\}$

$$MAE = \begin{cases} \frac{1}{|J|}\sum_{(x,i)\in J}|v(x,i) - v^*(x,i)| & \text{if } J \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

In order to quantify the amount of predictions that the recommender system can make, it is often used the coverage of a recommender system (see Definition 15.2), being defined as the percentage of predictions actually made by the system over the total amount of every possible prediction within the system. In the hypothetical case that all of the users had rated every item in the database, we would consider that the coverage is 1.

Definition 15.2 (Coverage). Let $A = \{(x,i)|x \in U,\ i \in I,\ v(x,i) = \bullet\}$
Let $B = \{(x,i)|x \in U,\ i \in I,\ v(x,i) = \bullet,\ v^*(x,i) \neq \bullet\}$

$$coverage = \begin{cases} \frac{|B|}{|A|} & \text{if } |A| \neq \emptyset \\ 1 & \text{otherwise} \end{cases}$$

15.3 Memory-Based Methods of Collaborative Filtering

In this section, we will focus on how recommender systems perform a prediction about the value with which the user $x \in U$ would rate the item $i \in I$, $v^*(x,i)$. These methods are based on the following idea: if we find a user $y \in U$ who has rated very similarly to $x \in U$, then, we can conclude that the user x's tastes are akin to those of the user y. Consequently, given an item $i \in I$ which the user x has not rated yet while the user y already has, we could infer that the user x would probably rate the item i with a similar value to the one given by the user y.

Thus, methods of this kind search, for each user, $x \in U$, a subset of k users, $y_1 \in U, \ldots, y_k \in U$ (called 'neighbors') who have rated very similarly to the user x. In order to predict the value with which the user x would rate an item $i \in I$, the recommender system examines first the values with which the neighbors $y_1, ..., y_k$ have rated the item i, and then, uses these values to make the prediction $v^*(x,i)$. Consequently, two main issues must be considered in order to make predictions:

- Evaluating how similar two users are in order to select, for each user x, a set of users, $y_1, ..., y_k$ (called 'neighbors') with similar tastes to the user x.
- Given a user x, and an item i, estimating the value with which the user x would rate the item i, $v^*(x,i)$, by considering the values with which the neighbors of x have rated this item i, $v(y_1,i)$,...,$v(y_k,i)$.

As far as the first issue is concerned, there are several possible measures for quantifying how similar the ratings between two different users are [18]. The similarity measure between two users $x, y \in U$ is defined on those items who have been rated by both x and y. That is to say, we define the set $C(x,y)$, of the common items between $x, y \in U$ as follows:

Definition 15.3 (Common Items). Given $x, y \in U$, we define $C(x,y)$ as the following subset of I:

$$C(x,y) = \{i \in I | v(x,i) \neq \bullet,\ v(y,i) \neq \bullet\}$$

The Mean Square Difference, *MSD*, may be regarded as the simplest similarity measure:

$$MSD(x,y) = \frac{1}{|C(x,y)|} \sum_{i \in C(x,y)} (v(x,i) - v(y,i))^2$$

As may be seen, *MSD* is based on a known metric distance[1] and $MSD(x,y) \geq 0$. When $MSD(x,y) = 0$, then we have that x and y have assigned exactly the same values to those items which both users have rated. Besides, the lower $MSD(x,y)$ is, the more similar the users x and y are.

The cosine between two vectors, cos, or the correlation coefficient, $\rho(x,y)$, are the similarity measures most used:

- The cosine similarity:

$$\cos(x,y) = \frac{\sum_{i \in C(x,y)} v(x,i)v(y,i)}{\sqrt{\sum_{i \in C(x,y)} v(x,i)^2} \cdot \sqrt{\sum_{i \in C(x,y)} v(y,i)^2}}$$

- The correlation coefficient or Pearson similarity:

$$\rho(x,y) = \frac{\sum_{i \in C(x,y)} (v(x,i) - \bar{x})(v(y,i) - \bar{y})}{\sqrt{\sum_{i \in C(x,y)} (v(x,i) - \bar{x})^2} \cdot \sqrt{\sum_{i \in C(x,y)} (v(y,i) - \bar{y})^2}}$$

where $\bar{x} = \frac{1}{|C(x,y)|} \sum_{i \in C(x,y)} v(x,i)$ and $\bar{y} = \frac{1}{|C(x,y)|} \sum_{i \in C(x,y)} v(y,i)$

[1] Indeed, $\sqrt{MSD}$ fulfills the definition of distance given in metric spaces when $\forall x \in U\ \forall i \in I\ v(x,i) \neq \bullet$.

Unlike $MSD(x, y)$, the measures $\cos(x, y)$ and $\rho(x, y)$ do not fulfill the conditions related to distance in metric spaces.[2] Indeed, both measures lie within the range $[-1, 1]$ and, when the higher $\cos(x, y)$ or $\rho(x, y)$ are, the more similar the users x and y are.

Once a similarity measure has been chosen, the recommender system selects, for each user x, a subset of the k users most similar to it, $N(x) = \{y_1, ..., y_k\}$, and then, these are used to predict how the user x will rate an item i: $v^*(x, i)$.

As for this late issue, the simplest way to determine $v^*(x, i)$ consists of calculating the mean of the values rated by the k users $N(x) = \{y_1, ..., y_k\}$. That is to say:

$$v^*(x, i) = \frac{1}{|B(x, i)|} \sum_{y \in B(x,i)} v(y, i) \tag{15.1}$$

where $B(x, i) = \{y \in N(x) | v(y, i) \neq \bullet\}$

The estimation $v^*(x, i)$ can be improved significantly by weighting more the values from the users who are more similar to x over those given by the users who are not so similar. If we consider a measure, *sim*, like ρ or cos, we could perform this easily in the following way:

$$v^*(x, i) = \frac{\sum_{y \in N(x)\ v(y,i) \neq \bullet} sim(x, y) \cdot v(y, i)}{\sum_{y \in N(x)\ v(y,i) \neq \bullet} sim(x, y)} \tag{15.2}$$

As may be seen, the Expression 15.2 works perfectly when using similarity measures like ρ or cos, since they give the higher values to those users who are more similar to a given one. However, the Expression 15.2 does not work when dealing with a function like *MSD*, since it gives lower values to those items who are more similar (indeed, when two users, x, y have rated exactly with the same values, the value $MSD(x, y) = 0$). Consequently, when using *MSD*, the estimation is calculated using the Expression 15.1.

Once we have selected the similarity measure and the estimation expression, the recommender system based on collaborative filtering has been just designed. Both the evaluation of the recommender system in relation to MAE (see Definition 15.1) and the *coverage* (see Definition 15.2) depend strongly on the constant k, the number of neighbors for each user. Indeed, the optimal value associated to this constant depends on the recommender system and is often hard to find out.

15.4 Choosing Variable Number of Neighbors for Each User

As we have described in the previous section, typical recommender systems based on collaborative filtering select, for each user x, the set of the k most similar users (neighbors), and then use these neighbors to make predictions about how the user x will rate the different items.

[2] In metric spaces, the distance $d(x, y)$ must fulfill that $d(x, x) = 0$. However, as may be seen, $\rho(x, x) = cos(x, x) = 1$.

In this section, we discuss a new technique to select the neighbors and calculate the estimations. This is based on choosing a variable number of neighbors (instead of choosing always k neighbors) for each user. This idea is inspired by the fact that, as it usually happens, a certain user x may have much more than k highly similar neighbors, while another one, y, may have much less than k. When this happens, in order to make predictions on x, we are in fact discarding a certain number of users which are highly similar to x, and in the same way, while making predictions on y, we would be including some users which are not similar enough to y, but are merely necessary to complete the fixed number of k neighbors.

In order to avoid this drawback, with our technique, a variable amount of neighbors is associated to each user in the following way. First we need to define a function, $d : U \times U \longrightarrow \mathbb{R}^+$, where $d(x, y)$ measures the inadequacy of user y's rates in order to predict the ratings of x. A user y will be considered as a neighbor of x when $d(x, y)$ lies under a constant value α (see Definition 15.6). This function d, like ρ and cos, is based on linear regression.

Next it must be dealt with obtaining this function d. We will consider that a user $y \in U$ is completely suitable to predict the ratings of user $x \in U$, if there is a value $b(x, y)$ such that for every item common to both users x and y, (that is to say, $\forall i \in C(x, y)$) the following holds:

$$v(x,i) = b(x,y) \cdot v(y,i) \tag{15.3}$$

In case the user $y \in U$ is not completely suitable to predict the user $x \in U$, there will be an error in the previous expression, and this expression will turn to be:

$$v(x,i) = b(x,y) \cdot v(y,i) + e(x,y,i) \tag{15.4}$$

Given $b(x, y)$, we evaluate the general error in the statement as follows:

$$\frac{1}{|C(x,y)|} \sum_{i \in C(x,y)} e(x,y,i)^2 \tag{15.5}$$

In order to define the unsuitability degree, d, of a user y to predict the user x, we will consider the value $b(x, y) \in \mathbb{R}$ such that the Expression 15.5 is minimum. As is commonplace in mathematics (using linear regression), this happens when $b(x, y)$ takes the value described in Definition 15.4.

Definition 15.4. We define $b : U \times U \longrightarrow \mathbb{R}$ as follows:

$$b(x,y) = \begin{cases} \dfrac{\sum_{i \in C(x,y)} v(x,i) \cdot v(y,i)}{\sum_{i \in C(x,y)} v(y,i)^2} & \text{if } C(x,y) \neq \emptyset \\ \infty & \text{otherwise} \end{cases}$$

Factor $b(x, y)$ is used in order to consider the case that two users employ different scales when rating items, even though they have similar tastes. That is to say, we will consider that users x and y have a similar taste when $\forall i \in C(x, y) v(x, i)$ tends to be very close to $b(x, y) \cdot y(i)$ (where $b(x, y)$ is a constant associated to both users x and y).

The function $d(x, y)$ is the minimum possible value in Expression 15.5 above.

$$d(x, y) = \min \frac{1}{|C(x, y)|} \sum_{i \in C(x,y)} e(x, y, i)^2 \tag{15.6}$$

It is not too hard to prove that this expression is completely equivalent to the one given in Definition 15.5. In this expression we have considered that d is infinite when there are no items common to both users x and y (that is to say, $C(x, y) = \emptyset$).

Definition 15.5. We define $d : U \times U \longrightarrow \mathbb{R}^+$ as follows:

$$d(x, y) = \begin{cases} \frac{1}{|C(x,y)|} \cdot \left(\sum_{i \in C(x,y)} v(x, i)^2 - \frac{(\sum_{i \in C(x,y)} v(x,i) \cdot v(y,i))^2}{\sum_{i \in C(x,y)} v(y,i)^2} \right) & \text{if } C(x, y) \neq \emptyset \\ \infty & \text{otherwise} \end{cases}$$

Besides, by Proposition 15.1, if we know the value $d(x, y)$, we could bound the value of the mean absolute error between the users x and the prediction made by user y, that is to say:

$$\frac{1}{|C(x, y)|} \sum_{i \in C(x,y)} |e(x, y, i)| = \frac{1}{|C(x, y)|} \sum_{i \in C(x,y)} |v(x, i) - b(x, y) \cdot v(y, i)|$$

Proposition 15.1. *Let $\gamma > 0$ be a positive real number. We have that if $d(x, y) \leq \gamma^2$, then:*

$$\frac{1}{|C(x, y)|} \sum_{i \in C(x,y)} |e(x, y, i)| \leq \gamma$$

Proof.

According to Expression 15.6, we have that:

$$d(x, y) = \frac{1}{|C(x, y)|} \sum_{i \in C(x,y)} e(x, y, i)^2$$

The following expression

$$\frac{1}{|C(x, y)|} \sum_{i \in C(x,y)} |e(x, y, i)|$$

reaches the maximum value for the users $x, y \in U$ fulfilling that:

$$d(x, y) = \frac{1}{|C(x, y)|} \sum_{i \in C(x,y)} e(x, y, i)^2 \leq \gamma^2$$

when

$$\forall i \in C(x,y)\ |e(x,y,i)| = \gamma$$

Consequently, we have that:

$$\frac{1}{|C(x,y)|}\sum_{i\in C(x,y)}|e(x,y,i)| \leq \frac{1}{|C(x,y)|}\sum_{i\in C(x,y)}\gamma = \gamma$$

□

The following proposition is immediately proven by taking into account the above one.

Proposition 15.2. *Let* γ *such that* $0 \leq \gamma \leq 1$.
Let $y \in N(x)$ *such that* $d(x,y) \leq \gamma^2 \cdot (max - min)^2$.
We have that:

$$\frac{1}{|C(x,y)|}\sum_{i\in C(x,y)}|e(x,y,i)| \leq \gamma \cdot (max - min)$$

Once we have defined the function d, we can state that a user y is a neighbor of x if the value of $d(x,y)$ keeps under constant α.

Definition 15.6 (Neighborhood). Given $x \in U$, we define $N(x)$ as the following set of users:

$$N(x) = \{y \in U | d(x,y) \leq \alpha\}$$

Although the parameter α may be defined arbitrarily and may depend on the specific recommender system in use, we suggest employing the following number (note that the possible values with which a user can rate an item are $\{min,\ldots,max\}$):

$$\alpha = \frac{(max - min)^2}{100}$$

When α takes this number, we can be sure, by Proposition 15.2, that for every neighbor, $y \in N(x)$, the mean absolute error between the users x and y is below the 10% of the difference $max - min$, that is to say:

$$\frac{1}{|C(x,y)|}\sum_{i\in C(x,y)}|e(x,y,i)| \leq 0.1 \cdot (max - min)$$

Once we have selected the neighbors of a user, $x \in U$, we can make an estimate on with which value the user x would rate an item i, by taking into consideration the neighbors of x.

Given an item $i \in I$ and a neighbor, $y \in N(x)$, who has rated the item i, we can estimate as $b(x,y) \cdot v(y,i)$ the value with which the user x would rate the item i. In the same way as in Expression 15.1, we take into account all the neighbors who have rated the item i to make the estimation $v^*(x,i)$ (see Definition 15.7).

In this way, we make an average of all the estimations arisen from the neighbors of x who have rated the item i. In case there are no neighbors which have rated the item i, we would say that $v^*(x, i) = \bullet$ (that is to say, we cannot estimate the value with which the user x would rate the item i).

Definition 15.7 (Estimation). We define the function $v^* : U \times U \longrightarrow \mathbb{R} \cup \{\bullet\}$ such that $\forall x \in U$ and $\forall i \in I$:

$$v^*(x,i) = \begin{cases} \dfrac{1}{|B(x,i)|} \sum_{y \in B(x,i)} b(x,y) \cdot v(y,i) & \text{if } B(x,i) \neq \emptyset \\ \bullet & \text{otherwise} \end{cases}$$

where $B(x,i) = \{y \in N(x) | v(y,i) \neq \bullet\}$

15.4.1 Example

Next, we will consider an example in order to illustrate what we have stated above.

Let us consider four users $x, y, z, t \in U$ and the items $i_1, i_2, i_3, i_4, i_5 \in I$. Let us consider that $V = \{1, 2, 3, 4, 5\}$. Just consider the following ratings made by the users:

$$\begin{array}{lllll} v(x,i_1) = 1 & v(x,i_2) = \bullet & v(x,i_3) = 2 & v(x,i_4) = 3 & v(x,i_5) = 5 \\ v(y,i_1) = \bullet & v(y,i_2) = 2 & v(y,i_3) = 3 & v(y,i_4) = 3 & v(y,i_5) = 5 \\ v(z,i_1) = 5 & v(z,i_2) = \bullet & v(z,i_3) = \bullet & v(z,i_4) = 3 & v(z,i_5) = 1 \\ v(t,i_1) = 2 & v(t,i_2) = 4 & v(t,i_3) = 4 & v(t,i_4) = 5 & v(t,i_5) = \bullet \end{array}$$

We calculate the value α:

$$\alpha = \frac{(5-1)^2}{100} = 0.16$$

In order to make a recommendation to user x, we need to calculate the neighbors of this user x. Consequently, we calculate previously the following:

$$\begin{aligned} C(x,y) &= \{i_3, i_4, i_5\} \quad |C(x,y)| = 3 \\ b(x,y) &= \frac{2{\cdot}3 + 3{\cdot}3 + 5{\cdot}5}{3^2 + 3^2 + 5^2} = 0.93 \;\; d(x,y) \\ &= \frac{1}{3}\left(2^2 + 3^2 + 5^2 - \frac{(2{\cdot}3 + 3{\cdot}3 + 5{\cdot}5)^2}{3^2 + 3^2 + 5^2}\right) = 0.26 > \alpha \end{aligned}$$

$$\begin{aligned} C(x,z) &= \{i_1, i_4, i_5\} \quad |C(x,z)| = 3 \\ b(x,z) &= \frac{1{\cdot}5 + 3{\cdot}3 + 5{\cdot}1}{5^2 + 3^2 + 1^2} = 0.66 \;\; d(x,z) \\ &= \frac{1}{3}\left(1^2 + 3^2 + 5^2 - \frac{(2{\cdot}3 + 3{\cdot}3 + 5{\cdot}1)^2}{5^2 + 3^2 + 1^2}\right) = 3.07 > \alpha \end{aligned}$$

$$C(x,t) = \{i_1, i_3, i_4\} \quad |C(x,t)| = 3$$

$$b(x,t) = \frac{1{\cdot}2 + 2{\cdot}4 + 3{\cdot}5}{2^2 + 4^2 + 5^2} = 0.55 \;\; d(x,t)$$

$$= \frac{1}{3}\left(1^2 + 2^2 + 3^2 - \frac{(1{\cdot}2 + 2{\cdot}4 + 3{\cdot}5)^2}{2^2 + 4^2 + 5^2}\right) = 0.04 \leq \alpha$$

As a result, there is only one neighbor of x, namely, t. That is to say,

$$N(x) = \{t\}$$

Now, we can estimate how user x would rate the item i_2 in the following way:

$$v^*(x, i_2) = \frac{1}{1}(b(x,t) \cdot v(t, i_2)) = 0.55 \cdot 4 = 2.2$$

15.5 The Coverage Improvement

As will be seen in Sect. 15.6, when evaluating the technique described in the previous section, we can see that (when α is low), the MAE level is extraordinarily good, but the coverage is very low. That is to say, the recommender system makes real good but few predictions.

In this section, we deal with a way to get a better coverage, while preserving to the greater possible extent the quality of the system's predictions.

First of all, we will study why the recommender system makes so few predictions. The main reason lies in the fact that since we only select as neighbors those users who are very suitable so as to predict the rating of a user x, the resulting set of neighbors is usually very small, and consequently it often happens that $B(x,i) = \emptyset$ for many $x \in U$ and $i \in I$.

In order to correct this, we propose to get a bigger set of the neighbors $N(x)$ of a user x, taking into account also some users, y, which, although might not be considered as very suitable to predict the ratings of x, were indeed useful for making predictions on some items which cannot be predicted by the set of neighbors $N(x)$ alone. This new enlarged set of neighbors of a user x will be called $N^*(x)$.

For each user, $x \in U$, and each item, $i \in I$, we will consider the neighbor $w(x,i)$, who is the user most suitable to predict the ratings of x among the users who have rated the item i (see Definition 15.8).

Definition 15.8. We define a function $w : U \times I \longrightarrow U$ fulfilling that

$$w(x,i) = y \Leftrightarrow d(x,y) = \min_{\substack{z \in U \\ v(z,i) \neq \bullet}} d(x,z)$$

For each user, $x \in U$, we include as neighbors every user $w(x,i)$ where $i \in I$ (see Definition 15.9). Consequently, in this new set of neighbors, not only we include very suitable neighbors to predict the ratings of x, but also those which let us get a closer prediction on how the user x would rate the items.

Definition 15.9 (New Neighborhood). Given $x \in U$, we define $N^*(x)$ as the following set of users:

$$N^*(x) = N(x) \cup \{w(x,i) | i \in I\}$$

where $N(x) = \{y \in U | d(x,y) \leq \alpha\}$

According to this definition, the coverage of the recommender system would be the highest possible, since we are considering for each user $x \in U$ and each item $i \in I$, the best neighbor of x who has rated the item i. Indeed, an item $i \in I$ cannot be predicted on behalf of the user x if and only if this item i has not been rated by any users.

Once we have defined the neighbors of each user, we will focus on how to make estimations. Unlike Definition 15.6, the new definition of neighbors of $x \in U$ in Definition 15.9 involves the possibility of including users who are not very suitable so as to predict the ratings of x. In order to have good levels of MAE, we need to weight (unlike the estimation proposed in Definition 15.7) the inadequacy levels of each neighbor of x, in such way that those users more suitable to predict the ratings of x will have more importance. Although this is the underlying idea already implied in Expression 15.2 in Sect. 15.3, we cannot use this expression since, unlike the similarity measures like ρ or cos, the lesser $d(x,y)$ is, the higher will be the adequacy level of user y in order to predict the ratings of x.

In order to weight the unsuitability degrees of each neighbor, we need to design a function f_α (see Definition 15.10):

Definition 15.10. Let $\alpha > 0$
We define the function $f_\alpha : \mathbb{R}^+ \longrightarrow [0,1]$ as follows:

$$f_\alpha(x) = \frac{\alpha^4}{x^4 + \alpha^4}$$

This function fulfills the following properties:

Proposition 15.3. *Let $\alpha > 0$. The previous function f_α has the following properties:*

i) $f_\alpha(0) = 1$ $f_\alpha(\alpha) = 1/2$
ii) *If* $0 \leq x_1 < x_2$*, then* $0 < f_\alpha(x_2) < f_\alpha(x_1) \leq 1$
iii) *If* $x \leq 0.5 \cdot \alpha$*, then* $f_\alpha(\alpha) > 0.9$
iv) *If* $x \geq 2 \cdot \alpha$*, then* $f_\alpha(\alpha) < 0.1$

The function f_α is suitable to weigh the neighbors of x when making estimations. As may be seen in Definition 15.11, we weigh, by means of f_α, the inadequacy degree of the neighbors of x.

Definition 15.11 (New Estimation). Let $x \in U$.

We define the function $v^* : U \times U \longrightarrow \mathbb{R} \cup \{\bullet\}$ such that $\forall x \in U$ and $\forall i \in I$, the following holds:

$$v^*(x,i) = \begin{cases} \dfrac{\sum_{y \in N^*(x)\ v(y,i) \neq \bullet} f_\alpha(d(x,y)) \cdot b(x,y) \cdot v(y,i)}{\sum_{y \in N^*(x)\ v(y,i) \neq \bullet} f_\alpha(d(x,y))} & \text{if } \exists y \in N^*(x)\ v(y,i) \neq \bullet \\ \bullet & \text{otherwise} \end{cases}$$

According to the properties of f_α, we give much more importance to those neighbors $y \in N(x)$ with $d(x,y) \leq \alpha$, than to those fulfilling $d(y,x) \geq \alpha$. Consequently, when there are neighbors $y \in U$ who have rated an item $i \in I$ (that is to say, $v(y,i) \neq \bullet$) and with unsuitability degree lower than α (that is to say, $d(x,y) \leq \alpha$), the estimation calculated in Definition 15.11 is very close to that which ensues from Definition 15.7 in the previous section. Besides, as we have said above, we are considering the highest possible value of coverage.

15.6 Evaluation of Our Techniques

In this section, we will analyze the value of coverage and MAE of our techniques in relation to the k-neighborhoods based on the metrics, *MSD*, ρ (‘correlation’) and cos. We have used the “MovieLens” database [23], which has been a reference for many years in research carried out in the area of Collaborative Filtering. The database contains 943 users, 1,682 items and 100,000 ratings, with a minimum of 20 items rated per user. The items represent motion pictures and the rating values range from 1 to 5.

In relation to the techniques presented here, we calculate the constant α as we suggested in a previous section:

$$\alpha = \frac{(5-1)^2}{100} = 0.16$$

In relation to the techniques presented in Sects. 15.4 and 15.5, we have obtained the values of MAE and coverage described in Table 15.1. As may be seen, although the MAE of this second technique is not so good in as the first technique, the coverage level has been increased significantly.

Figure 15.1 illustrates the MAE and coverage values of the algorithm of the k nearest neighbors for different values of k (15, 30, 60, 90, 120, 150, 180, 210 and 240), covering from 1.6 to 25% of the total number of users.

Table 15.1 Evaluation of the techniques presented in the paper

Technique	MAE	Coverage
Technique 1 (Sect. 15.4)	0.099	35.99%
Technique 2 (Sect. 15.5)	0.504	100%

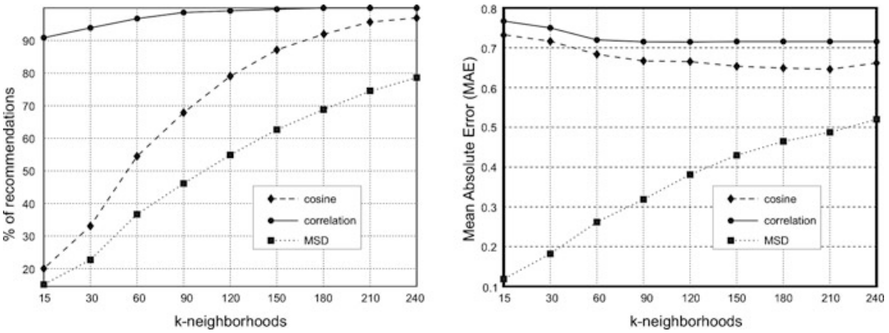

Fig. 15.1 MAE and Coverage for the k nearest neighbors algorithm based on the metrics MSD, ρ and cos

Although *MSD* provides the best results in relation to MAE, the coverage level keeps very low (until the value of k is high). Instead, the similarity measure ρ ("correlation") keeps good levels of coverage and MAE (this is an example which helps understand why correlation is often used instead of *MSD*).

As may be seen, the values the algorithm k-nearest neighbors provides for any of the three similarity measures and for any value in the constant k are consistently worse than those obtained by the technique presented in Sect. 15.5. Besides, our technique presents the advantage that it does not need to find the value for any parameter, unlike the classic algorithm, whose results depend significantly on the parameter k (and consequently, it is always necessary to find out the optimal value for this parameter beforehand).

15.7 Conclusions

In this paper, we have presented a new technique for making predictions on recommender systems based on collaborative filtering. The underlying idea is based on selecting a variable number of neighbors for each user on behalf of the number of high similar users (neighbors) in the database (instead of, as it is usually made, selecting always a constant number k of neighbors). Thanks to this new technique, we can get a significant improvement of both the value MAE and the coverage.

References

1. G. Adomavicius and A. Tuzhilin, Toward the Next Generation of Recommender Systems: a survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Enginnering, Vol. 17, No 6, 2005, pp. 734–749
2. N. Antonopoulus and J. Salter, Cinema screen recommender agent: combining collaborative and content-based filtering, IEEE Intelligent Systems, 2006, pp. 35–41

3. R. Baraglia and F. Silvestri, An Online Recommender System for Large Web Sites, Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, 2004, pp. 199–205
4. J. Bobadilla, F. Serradilla and A. Hernando, Collaborative Filtering adapted to Recommender Systems of e-learning, Knowledge Based Systems, Vol. 22, 2009, pp. 261–265
5. J.S Breese, D. Heckerman and C. Kadie, Empirical Analysis of Predictive Algorithms for Collaborative Filtering, Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 43–52, 1998
6. S.B. Cho, J.H. Hong and M.H. Park, Location-Based Recommendation System Using Bayesian Users Preference Model in Mobile Devices, Lecture Notes on Computer Science, 4611, pp. 1130–1139, 2007
7. I. Fuyuki, T.K. Quan and H. Shinichi, Improving Accuracy of Recommender Systems by Clustering Items Based on Stability of User Similarity, Proceedings of the IEEE International Conference on Intelligent Agents, Web Technologies and Internet Commerce, pp. 61–61, 2006
8. G.M. Giaglis and G. Lekakos, Improving the Prediction Accuracy of Recommendation Algorithms: Approaches Anchored on Human Factors, Interacting with Computers, Vol. 18, No. 3, 2006, pp. 410–431
9. J.L. Herlocker, J.A. Konstan, J.T. Riedl and L.G. Terveen, Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems, Vol. 22, No. 1, 2004, pp. 5–53
10. H. Ingoo, J.O. Kyong and H.R. Tae, The collaborative filtering recommendation based on SOM cluster-indexing CBR, Expert Systems with Applications, Vol. 25, 2003, 413–423
11. T. Janner and C. Schroth, Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services, IT Pro, 2007, pp. 36–41
12. M. Knights, Web 2.0, IET Communications Engineer, Vol. 5, No. 1, 2007, pp. 30–35
13. F. Kong, X. Sun and S. Ye, A Comparison of Several Algorithms for Collaborative Filtering in Startup Stage, Proceedings of the IEEE networking, sensing and control, 2005, pp. 25–28
14. J.A. Konstan, B.N. Miller and J. Riedl, PocketLens: toward a personal recommender system, ACM Transactions on Information Systems, Vol. 22, No. 3, 2004, pp. 437–476
15. K.J. Lin, Building Web 2.0, Computer, Vol. 40, No. 5, 2007, pp. 101–102
16. F. Loll and N. Pinkwart, Using Collaborative Filtering Algorithms as eLearning Tools, 42nd Hawaii International Conference on System Sciences HICSS '09, 2009, pp. 1–10
17. Y. Manolopoulus, A. Nanopoulus, A.N. Papadopoulus and P. Symeonidis, Collaborative recommender systems: combining effectiveness and efficiency, Expert Systems with Applications, Vol. 34, No. 4, 2008, pp. 2995–3013
18. J.L. Sanchez, F. Serradilla, E. Martinez and J. Bobadilla, Choice of Metrics used in Collaborative Filtering and their Impact on Recommender Systems, Proceedings of the IEEE International Conference on Digital Ecosystems and Technologies (DEST'08), 2008, pp. 432–436
19. S. Staab, H. Werthner, F. Ricci, A. Zipf, U. Gretzel, D.R. Fesenmaier, C. Paris and C. Knoblock, Intelligent Systems for Tourism, Intelligent Systems, Vol. 17, No. 6, 2002, pp. 53–64
20. P. Symeonidis, A. Nanopoulos and Y. Manolopoulos, Providing Justifications in Recommender Systems, IEEE Transactions on Systems, Man and Cybernetics, Part A, Vol. 38, No. 6, 2008, pp. 1262–1272
21. K. Wei, J. Huang and S. Fu, A survey of e-commerce recommender systems, Proceedings of the International Conference on Service Systems and Service Management, 2007, pp. 1–5
22. R.R. Yager, Fuzzy Logic Methods in Recommender Systems, Fuzzy Sets and Systems, Vol. 136, No.2, 2003, pp. 133–149
23. http://www.movielens.org