Chapter 8

Neighbour selection and weighting in user-based collaborative filtering

User-based recommender systems suggest interesting items to a user relying on similar-minded people called neighbours. The selection and weighting of the input from these neighbours characterise different variants of the approach. Thus, for instance, while standard user-based collaborative filtering strategies select neighbours based on user similarities, trust-aware recommendation algorithms rely on other aspects indicative of user trustworthiness and reliability.

In this chapter we restate the user-based recommendation problem, generalising it in terms of performance prediction techniques. We investigate how to adopt this generalisation to define a unified framework where we conduct an objective analysis of the effectiveness (predictive power) of neighbour scoring functions. We evaluate our approach with several state-of-the-art and novel neighbour scoring functions on two publicly available datasets. The notion of performance takes here a different nuance from previous chapters. More precisely, we consider the notion of neighbour performance, for which we propose several measures and new predictors. In an empirical comparison involving four neighbour quality metrics and thirteen performance predictors, we find a strong predictive power for some of the predictors with respect to certain metrics. This result is then validated by checking the final performance of recommendation strategies where predictors are used for selecting and/or weighting user neighbours. As a result, we are able to anticipate which predictors will perform better in neighbour scoring powered versions of a user-based collaborative filtering algorithm.

In Sections 8.1 and 8.2 we present a unified formulation and the proposed framework for neighbour selection and weighting in user-based recommendation, and in Section 8.3 we describe how the different neighbour scoring functions proposed in the literature fit into the framework. Finally, in Section 8.4 we present an experimental evaluation of the framework, and in Section 8.5 we provide conclusions.

8.1 Problem statement

We focus on user-based collaborative filtering algorithms, one type of memory-based approaches that explicitly seek people – commonly called **neighbours** – having preferences (and/or other characteristics of interest) in common with the target user, and use such preferences to predict item ratings for the user. User-based algorithms are built on the principle that a particular user's rating records are not equally useful to all other users as input to provide them with item suggestions (Herlocker et al., 2002). Therefore, as stated in Chapter 2, central aspects to these algorithms are a) how to identify which neighbours form the best basis to generate item recommendations for the target user, and b) how to properly make use of the information provided by them. Once the target user's neighbours are selected, the more similar a neighbour is to the user, the more her preferences are taken into account as input to produce recommendations.

A common user-based recommendation approach consists of predicting the relevance of an item for the target user by a linear combination of her neighbours' ratings, which are weighted by the similarity between the target user and her neighbours, as presented in Equation (2.3). For the sake of clarity, and since we shall later elaborate from it, we reproduce here the above equation:

$$\tilde{r}(u,i) = \bar{r}(u) + C \sum_{v \in N_k(u,i)} sim(u,v)\big(r(v,i) - \bar{r}(v)\big) \tag{8.1}$$

User similarity has been the central criterion for neighbour selection in most of the user-based collaborative filtering approaches (Desrosiers and Karypis, 2011). Nonetheless, recently it has been suggested that additional factors could have a valuable role to play on this point. For instance, two users with a high similarity value may no longer be reliable predictors for each other at some point because of a divergence of tastes over time (O'Donovan and Smyth, 2005). Thus, in the context of user-based collaborative filtering, more complex methods have been proposed in order to effectively select and weight useful neighbours (O'Donovan and Smyth, 2005; Desrosiers and Karypis, 2011). In this context a particularly relevant dimension relates the above additional factors with the general concept of trust (trustworthiness, reputation) on a user's contribution to the computation of recommendations. Hence, a number of trust-aware recommender systems have been proposed in the last decade (Hwang and Chen, 2007; O'Donovan and Smyth, 2005; Golbeck, 2009).

Most of these systems focus on the improvement of accuracy metrics, such as the Mean Average Error, by defining different heuristic trust functions, which, in most cases, are applied either as additional weighting factors in the neighbourhood-based formulation, or as a component of the neighbour selection criteria. The way trust is measured is considerably diverse in the literature. In fact, the notion of trust

has embraced a wide scope of neighbour aspects, spanning from personal trust on the neighbour's faithfulness, to trust on her competence, confidence in the correctness of the input data, or the effectiveness of the recommendation resulting from the neighbour's data. More specifically, in trust-aware recommender systems, a trust model is defined and, typically, introduced into the Resnick's equation (Equation (8.1)) either as an additional weight or as a filter for the potential user's neighbours. Moreover, depending on the nature of their input, different types of trust-aware recommendation approaches can be distinguished: rating-based approaches, and social-based approaches (using a trust network).

One of the first works that proposed *rating-based trust metrics* between users is (O'Donovan and Smyth, 2005). In that work O'Donovan and Smyth propose to modify how the "recommendation partners" (neighbours) are weighted and selected in the user-based collaborative filtering formula. They argue that the trustworthiness of a particular neighbour should be taken into account in the computed recommendation score by looking at how reliable her past recommendations were. Trust values are computed by measuring the amount of correct recommendations in which a user has participated as a neighbour, and then they are used for weighting the influence (along with computing the similarity), and selecting the target user's neighbours. Weng et al. (2006) propose an asymmetric trust metric based on the expectation of other users' competence in providing recommendations to reduce the uncertainty in predicting new ratings. The metric is used in the standard collaborative filtering formula instead of the similarity value. Two additional metrics are defined in (Kwon et al., 2009) based on the similarity between the ratings of a neighbour and the ratings from the community. Finally, Hwang and Chen (2007) define two trust metrics (local and global) by averaging the prediction error of co-rated items between a user and a potential neighbour.

Social-based trust metrics make use of explicit trust networks of users, built upon friendship relations (Massa and Avesani, 2004; Massa and Bhattacharjee, 2004) and explicit trust scores between individuals in a system (Ma et al., 2009; Walter et al., 2009). These metrics and, to some extent, their inherent meanings, are different with respect to rating-based metrics. Nonetheless, Ziegler and Lausen (2004) conduct a thorough analysis that shows empirical correlations between trust and user similarities, suggesting that users tend to create social connections with people who have similar preferences. Once such a correlation is proved, techniques based on social-based trust can be applicable. Golbeck and Hendler (2006) propose a metric called TidalTrust to infer trust relationships by using recursive search. Inferred trust values are used for every user who has rated a particular item in order to select only those users with high trust values. Then, a weighted average between past ratings and inferred trust values provides the predicted ratings. Massa and Avesani (2007b) ex-

periment with local (MoleTrust) and global (PageRank) trust metrics, showing that trust-based recommenders are very valuable for cold start users.

The research presented here seeks to provide an algorithmic generalisation for a significant variety of notions, computational definitions, and roles of trust in neighbour selection. Specifically, we aim to provide a theoretical framework for neighbour selection and weighting in which trust metrics can be defined and evaluated in terms of improvements on a final recommender's performance. We cast the rating prediction task – typically based, as described above, on the aggregation of neighbour preferences – into a framework for dynamic combination of inputs, from a performance prediction perspective, borrowing from the methodology for this area in the Information Retrieval field. The application of this perspective is not trivial, and requires a definition of what the performance of a neighbour means in this context. Hence, restated the problem in these terms, we propose to adapt and exploit techniques and methodologies developed in Information Retrieval for predicting query performance; in our case the target user's neighbours are equivalent to the queries, and our goal is to predict which of these neighbours will perform better for the target user.

Furthermore, since our framework provides an objective measure of the neighbour scoring function efficiency, we would be able to obtain a better understanding of the whole recommendation process. For instance, if the results obtained when a particular function is introduced in a recommender are not consistent with the (already observed) objective performance measures, it would mean that the chosen strategy is not the most appropriate, suggesting to experiment with further strategies, providing such a function has already shown some predictive power.

Therefore, the main contribution of our framework is that it provides a formal setting for the evaluation of neighbour selection and weighting functions, while, at the same time, enables to discriminate whether recommendation performance improvements are achieved by the neighbour scoring functions, or by the way these functions are used in the recommendation computation. Besides, our framework provides an unification of state-of-the-art trust-based recommendation approaches, where trust metrics are casted as neighbour performance predictors. As a result, in this chapter, we shall propose four neighbour quality metrics and thirteen performance predictors, defined upon a specific neighbour (user-based), a neighbour and the current user (user-user), or a neighbour and the current item (user-item). We shall generalise the different strategies proposed in the literature to introduce trust into collaborative filtering. Moreover, thanks to the proposed formulation, we will define and evaluate new strategies.

8.2 A performance prediction framework for neighbour scoring

8.2.1 Unifying neighbour selection and weighting in user-based Recommender Systems

From the observation that most of the methods for neighbour selection and weighting are elaborated upon the standard Resnick's scheme (Equation (8.1)), we propose a unified formulation as follows. Let us suppose, for the sake of generality, that we have a neighbour scoring function $s(u, v, i)$ that may depend on the target user u, a neighbour v, and a target item i. This function outputs a higher value whenever the user, neighbour, item, or a combination of them, is more trustworthy (in the case of trust models), or is expected to perform better as a neighbour according to the information available in the system, such as other ratings and external information, like a social network. Using this function we generalise Equation (8.1) to:

$$\tilde{r}(u,i) = \bar{r}(u) + C \sum_{v \in f^{neigh}(u,i;k;s)} f^{agg}\big(s(u,v,i), sim(u,v)\big)\big(r(v,i) - \bar{r}(v)\big) \quad (8.2)$$

where the function f^{neigh} denotes the selection of the set of neighbours, and f^{agg} is an aggregation function combining the output of s and the user similarity into a single weight value. In this way, we integrate the neighbour scoring function s into the Resnick's formula in order to: a) select the neighbours to be considered, instead of or in addition to the most similar users (via function f^{neigh}), and b) provide a general weighting scheme by introducing an aggregation function f^{agg} between the actual neighbour score and the similarity between the target user and her neighbours. Note that it is not required that s is bounded, since a constant C would normalise the output rating value. The function s is thus a core component in the generalisation of the user-based collaborative filtering techniques. It may embody similarity in itself (in such case f^{agg} may just return its first input argument), but sim and f^{agg} are left to simplify the connection with the original similarity-only formulation, and to suit particular cases where s applies other principles distinct to similarity.

The aggregation function f^{agg} can take different definitions, some examples of which can be found in the literature. For instance, O'Donovan and Smyth (2005) initially propose to use the arithmetic mean of the neighbour score (x) and the similarity (y; henceforth denoted as f_1^{agg}), and end up using the harmonic mean (f_2^{agg}) because of its better robustness to large differences in the inputs. In (Bellogín and Castells, 2010), on the other hand, we use the product function (f_3^{agg}). Moreover, Hwang and Chen (2007) propose to directly use the neighbour score as the weight

given to neighbours, that is, they use the projection function $f_4^{agg}(x,y) = x$. Obviously, the original Resnick's formulation can be expressed as the symmetric projection function $f_0^{agg}(x,y) = y$.

The neighbourhood selection embodied in function f^{neigh} also generalises Resnick's approach – the latter corresponds to the particular case $f_0^{neigh}(u,i;k;s) = N_k(u,i)$, where the neighbour scoring function is ignored, and only similarity is used. The general form admits different instantiations. In (Golbeck and Hendler, 2006) only the users with the highest trust values are selected as neighbours. In (O'Donovan and Smyth, 2005), on the other hand, those users whose trust values exceed a certain threshold are taken into consideration. This threshold is empirically defined as the mean across all the obtained values for each pair of users. The latter strategy can be formulated as follows:

$$f_1^{neigh}(u,i;k;s) = \{v \in N_k(u,i): s(u,v,i) > \tau\};\ \tau = \frac{1}{|\{(u,v,i)\}|}\sum_{(u,v,i)} s(u,v,i)$$

There are, nonetheless, some considerations to take into account when using specific combinations of neighbour weighting and neighbour selection functions. First, if f_4^{agg} is used together with f_0^{neigh} – only considering the most similar users in the neighbourhood –, then less reliable users (with low f_4^{agg}) who are very similar to the current user would be penalised, and more reliable neighbours but less similar to the current user are ignored, since they do not belong to the neighbourhood. Second, when using f_0^{agg} together with f_1^{neigh}, neighbours are weighted by their similarities with the target user. These similarities, however, could be very low, and thus, non-similar but reliable neighbours would be penalised. Finally, if f_4^{agg} is used with f_1^{neigh}, the similarity weight will not be considered at any point in the recommendation process.

Some of these configurations may deserve further investigation, and are considered in Section 8.4, along with other combinations not listed here.

8.2.2 Neighbour selection and weighting as a performance prediction problem

Neighbour scoring and selection can be seen as a task of predicting the effectiveness of neighbours as input for collaborative recommendations. In this section we elaborate and adapt the performance prediction framework presented in Chapter 5 to the problem of neighbour selection and weighting.

The same as performance prediction in Information Retrieval, which has been used to optimise rank aggregation (Yom-Tov et al., 2005a), in our proposed framework each user's neighbour can be considered as a retrieval subsystem (or criterion)

whose output is combined to form a final system's output (the recommendations) to the user.

For user-based collaborative filtering algorithms, the estimation $\tilde{r}(u,i)$ of the preference of the target user u for a particular item i can be formulated as an aggregation function of the ratings of some other users $\hat{V}$:

$$\tilde{r}(u,i) \propto \text{aggr}_{v \in \hat{V}}\big(sim(u,v);\ r(v,i); \bar{r}(u); \bar{r}(v)\big) \tag{8.3}$$

where $\hat{V}$ denotes the selected neighbours for a particular user u according to function f^{neigh} (see Equation (8.2)). As observed in (Adomavicius and Tuzhilin, 2005), different aggregation functions can be defined, but the most typical one is the weighted average function presented in the previous section.

In the previous function the term $\tilde{r}(u,i)$ can be seen as a retrieval function that aggregates the outputs of several utility subfunctions $r(v,i) - \bar{r}(v)$, each corresponding to a recommendation obtained from a neighbour of the target user. The combination of utility values is defined as a linear combination (translated by $\bar{r}(u)$) of the neighbours' ratings, weighted by their similarity $sim(u,v)$ with the target user. Hence, the computation of utility values in user-based filtering is equivalent to a typical rank aggregation model of Information Retrieval, where the aggregated results may be enhanced by predicting the performance of the combined recommendation outputs. In fact, the similarity value can be seen as a prediction of how useful a neighbour's advice is expected to be for the target user, which has proved to be a quite effective approach. The question is whether other performance factors beyond user similarity can be considered in a way that further enhancements can be drawn, as research on user trust awareness has attempted to prove in the last years.

The Information Retrieval performance prediction view provides a methodological approach, which we propose to adapt to the neighbour selection problem. The approach provides a principled path to drive the formulation, development and evaluation of effective neighbour selection and weighting techniques, as we shall see. In the proposed view, the selection/weighting problem is expressed as an issue of neighbour performance, as an additional factor (besides user similarity) to automatically tune the neighbours' contribution to the recommendations, according to the expected goodness of their advice. As summarised in Section 5.1, there are three core concepts in the performance prediction problem as addressed in the Information Retrieval literature: performance predictor, retrieval quality assessment, and predictor quality assessment. Since we are dealing with the prediction of which users may perform better as neighbours, the above three concepts can respectively be translated into *neighbour performance predictor*, *neighbour quality*, and *neighbour predictor quality*. For the sake of simplicity, let us assume we can define a performance predictor as a function that receives as input a user profile u (in general, it could receive other users or items as well), the set of items $\mathcal{I}_u$ rated by that user, and the collection S of ratings and

items (or any other user preference and item description information) available in the system. Then, following the notation given used in Chapter 5, we define a neighbour performance prediction function as:

$$\hat{\mu}(u) \leftarrow \gamma(u, \mathcal{I}_u, S). \tag{8.4}$$

The function γ can be defined in different ways, for instance, by taking into account the rating distribution of each user, the number of ratings available in the system, and the (implicit or explicit) relations made by that user with the rest of the community. Essentially, the neighbour performance predictor is intended to estimate the true neighbour quality metric, denoted as $\mu(u)$, which is typically measured using groundtruth information about whether the neighbour's influence is positive. The application of this perspective is not trivial, and requires, in particular, a definition of what the performance of a neighbour means in this context – where no standard metric for neighbour performance is yet available in the literature.

Once the estimated neighbour performance prediction values $\hat{\mu}(u_n)$ are computed for all users, the quality of the prediction can be measured as presented in Section 5.4.2, that is, either by measuring the correlation between the estimations and the real values $\mu(u_n)$, or by using classification accuracy metrics such as the F-measure. Since in this case we are interested in providing a ranking of users, this relates more with the traditional query performance task, and not with query difficulty (see Section 5.4.1), where the latter metrics are used. In other words, the neighbour predictor quality metric is defined as the following correlation:

$$q(\gamma) = \text{corr}([\hat{\mu}(u_1), \cdots, \hat{\mu}(u_n)], [\mu(u_1), \cdots, \mu(u_n)]). \tag{8.5}$$

Similarly to the situation in Information Retrieval, this correlation provides an assessment of the prediction accuracy (Carmel and Yom-Tov, 2010); the higher its (absolute) value, the higher the predictive power of γ. Moreover, the sign of $q(\gamma)$ represents whether the two involved variables – neighbour prediction and neighbour quality – are directly or inversely correlated.

Besides validating any proposed predictor by checking the correlation between predicted outcomes and objective metrics, we may further test the effectiveness of the defined predictors by introducing and testing a dynamic variant of user-based collaborative filtering. In this variant, the weights of neighbours are dynamically adjusted based on their expected effectiveness, along with the decision of which users belong to each neighbourhood, as in the general formulation presented in Equation (8.2). We propose to define the neighbour scoring function $s(u, v, i)$ based on the values computed from each neighbour performance predictors.

Hence, the basic idea of the framework presented here is to formally treat the neighbour selection and weighting in memory-based recommendation as a performance prediction problem. The performance prediction framework provides a principle basis to analyse whether the predictors are capturing some valuable, measurable

characteristic known to be useful for prediction, independently from their latter use in a recommendation strategy. Furthermore, if a neighbour scoring function with strong predictive power is introduced into the recommendation process and the performance is not improved, then, new ways of introducing such predictor into the rating estimation should be tested (either for selection or weighting), since we have some confidence that this function captures interesting user's characteristics, valuable for recommendation.

8.3 Neighbour quality metrics and performance predictors

The performance prediction research methodology requires a means to compare the predicted performance with the observed performance. This comparison is typically conducted in terms of some one-dimensional functional values, where the performance is assessed by some specific metric and the prediction can be translated to a certain numeric value. This value quantifies the expected degree of effectiveness, providing, thus, a relative magnitude.

Whereas in the context of performance prediction in IR, standard metrics of system effectiveness in response to a query are used for this purpose, in the case of predicting the performance of a neighbour for recommendation we would require to use metrics that measure how effective a neighbour is. In this section we propose several neighbour quality metrics and performance predictors which we shall evaluate in Section 8.4.

8.3.1 Neighbour quality metrics

The purpose of effectiveness predictors in our framework is to assess how useful specific neighbour profiles are as a basis for predicting ratings for the target user. Each predictor has to be contrasted to a measure of how "good" the neighbour's contribution is to the global community of users in the system. In contrast with query performance prediction, where a well established array of metrics are used to quantify query performance, to the best of our knowledge, in the literature there is not an equivalent function for neighbours used in user-based collaborative filtering. We therefore need to introduce and propose some sound candidate metrics.

Ideally, in the proposed framework, a quality metric should take the same arguments as the predictor, and thus, if we have, for instance, a user-item predictor, we should also be able to define a quality metric that depends on users and items. In general, we shall focus on user-based predictors, but it would be possible to explore item-based alternatives. Furthermore, we shall consider metrics taking neighbours as single input, independently from which neighbourhood is involved (i.e., independ-

ently from the target user), and which item is recommended. At the end of this section, nonetheless, we shall introduce a neighbour quality metric suitable for the user-user scenario, where both the target user and neighbour are taken into account.

Now, we propose three different neighbour quality metrics. The first two metrics had a different intended use by their authors, but we found they could be useful to evaluate how good a user is as a neighbour. The third metric was proposed by us in (Bellogín and Castells, 2010), where the problem of neighbour performance was explicitly addressed.

Rafter et al. (2009) propose two metrics in order to examine whether the neighbours have any influence in the recommendation accuracy. Both metrics are based on the comparison between true ratings and a neighbour's estimation of the ratings, as a way to measure the direction of the neighbour estimation and the average absolute magnitude of the shift produced by this estimation. Thus, the larger the neighbour's influence, the better her performance, according to our definition of a "good" neighbour. In this context we use those metrics as follows:

$$\mu_1 = \mu(v) = \frac{1}{|T_v|}\sum_{i \in T_v} \frac{1}{|N_k^{-1}(v;i)|} \sum_{w \in N_k^{-1}(v;i)} |r(w,i) - r(v,i)|$$

$$\mu_2 = \mu(v) = \frac{1}{|T_u|}\sum_{i \in T_v} \frac{1}{|N_k^{-1}(u;i)|} \sum_{w \in N_k^{-1}(v;i)} \delta\big([sgn(r(w,i) - \bar{r}(v)) = sgn(r(v,i) - \bar{r}(v))]; 1\big)$$

where δ is a binary function whose output is 1 if its arguments are true, and 0 otherwise. Metric μ_1 represents the **absolute error deviation** of a particular user, and μ_2 is the **sign of error deviation**. Note that $N_k^{-1}(v;i)$ denotes an inverse neighbourhood, which represents those users for whom v is a neighbour, and T_v denotes the items rated by user v in the test set. We can observe how each of these metrics represents a different method to measure how accurate the user v is as a neighbour.

In (Bellogín and Castells, 2010) we proposed a metric named **neighbour goodness**, which is defined as the difference in performance of the recommender system when including vs. excluding the user (i.e., her ratings) from the dataset. For instance, based on the mean average error standard metric, neighbour goodness can be instantiated as:

$$\mu_3 = \mu(v) = \frac{1}{|R_{\mathcal{U}\setminus\{v\}}|} \sum_{w \in \mathcal{U}\setminus\{v\}} \left[CE_{\mathcal{U}\setminus\{v\}}(w) - CE_{\mathcal{U}}(w)\right]$$

$$CE_X(v) = \sum_{i \in \mathcal{I}, r(v,i) \neq \emptyset} |\tilde{r}_X(v,i) - r(v,i)|$$

where $\tilde{r}_X(v,i)$ represents the predicted rating computed using only the data in X. This metric quantifies how much a user affects (contributes to or detracts from) the

total amount of mean average error of the system, since it is computed in the same way as that metric, but leaving out the user of interest – in the first term, the user is completely omitted; in the second term, the user is only involved as a neighbour. In this way we measure how a user contributes to the rest of users, or put informally, how better or worse the "world" is in the sense of how well recommendations work with and without the user. Hence, if the error increases when the user is removed from the dataset, it is considered as a good neighbour.

Based on the same idea of the previous metric, we propose a user-user quality metric that measures how one particular user affects to the error of another user when acting as her neighbour:

$$\mu_4 = \mu(u, v) = CE_{U\setminus\{v\}}(u) - CE_U(u)$$

We call this metric **user-neighbour goodness**. It quantifies the difference in user u's error when neighbour v is not in the system against the error when such neighbour is present, that is, it measures how much each neighbour contributes to reduce the error of a particular user.

8.3.2 Neighbour performance predictors

Having formulated neighbour selection in memory-based recommendation as a task of neighbour effectiveness prediction, and having proposed effectiveness metrics to compare against, the core of an approach to this problem is the definition of effectiveness predictors. For this purpose, similarity functions and trust models such as those mentioned in Section 8.1 can be directly used, since in trust-aware recommendation, trust metrics aim at measuring how reliable a neighbour is when introduced in the recommendation process (O'Donovan and Smyth, 2005). Interestingly, some of them only depend on one user (**global trust metrics**), and others depend on a user and an item or another user (**local trust metrics**). Furthermore, other authors have proposed different indicators for selecting good neighbours, mainly based on the overlap between the user and her neighbour, without considering the concept of trust.

We thus distinguish three types of neighbour performance predictors: **user predictors** – equivalent to the global trust metrics –, **user-item predictors**, and **user-user predictors** – equivalent to the local trust metrics. Note that, although trust metrics could now be interpreted as neighbour performance predictors, the proposed performance prediction framework let us to provide an inherent value to these metrics (identified as performance predictors), independently from whether they improve a recommender's performance when used for selecting or weighting in the specific collaborative filtering algorithm. This is due to the fact that it is possible to empirically check the quality of the prediction by analysing their correlation with respect to the neighbour performance metric, prior to the integration in any collabora-

tive filtering method. Thus, each predictor would obtain an explicit score that represents its predictive power, related to our *a priori* confidence on whether such predictor is capturing the neighbour's reliability or trustworthiness.

In the following we propose an array of neighbour effectiveness prediction methods, by adapting and integrating trust functions from the literature into our framework, and we also propose novel prediction functions.

User Predictors

User predictors are performance predictors that only depend on the target neighbour. When that neighbour is predicted to perform well, her assigned weight in the user-based collaborative filtering formulation is high.

One of the first user trust metrics proposed in the literature is the **profile-level trust** (O'Donovan and Smyth, 2005), which is defined as the percentage of correct recommendations in which a user has participated as a neighbour. If we denote the set of recommendations in which a user has been involved as

$$\text{RecSet}(u) = \{(v, i) : u \in N_k(v; i)\},$$

then the predictor is defined as follows:

$$\gamma_1(u, v, i) = \gamma(v) = \frac{|\text{CorrectSet}(v)|}{|\text{RecSet}(v)|},$$

where the definition of correct recommendations depends on a threshold ϵ:

$$\text{CorrectSet}(u) = \{(c_k, i_k) \in \text{RecSet}(u) : \text{Correct}(i_k, u, c_k; 1)\}$$

$$\text{Correct}(i, u, v; \lambda) = \delta(|r(u, i) - r(v, i)| \leq \epsilon; \lambda),$$

$\delta(a; b)$ being a binary function like before whose output is a value b if the predicate a is true, and 0 otherwise. That is, the recommendations considered as correct are those in which the user was involved as a neighbour, and her ratings were close (up to a distance of ϵ) to the actual ratings.

A similar trust metric, called **expertise trust**, is presented in (Kwon et al., 2009), where the concept of 'correct recommendation' is also used. In that work Kwon and colleagues introduce a compensation value for situations in which few raters are available. Specifically, the correct recommendation function only outputs a value of 1 when there are enough raters for a particular item (more than 10 in the paper). Otherwise, an attenuation factor is introduced by dividing the number of raters by 10, in the same way as significance weighting is introduced in Pearson's correlation in (Herlocker et al., 2002). More formally, the predictor is defined as:

$$\gamma_2(u, v, i) = \gamma(v) = \frac{1}{\sum_{j \in I_v} \sum_{w \in \mathcal{U}_i} 1} \sum_{j \in \mathcal{I}_v} \sum_{w \in \mathcal{U}_i} \text{Correct}\big(j, v, w; \lambda(j)\big)$$

where $\lambda(j)$ is 1 when item j has more than 10 raters, and $\mathcal{U}_i$ denotes the users who rated item i. In the same paper the authors propose another trust metric called **trustworthiness**, which is equivalent to the absolute value of the similarity between the target user's ratings and the average ratings given by the community (denoted as $\bar{R}$). The authors introduce the significance weighting factor β as in (Herlocker et al., 2002), in a way that $\beta(v)$ is 1 when user v has more than 50 ratings; otherwise, β is computed as the user's ratings divided by 50. Once the β factor is computed, the predictor is defined as follows:

$$\gamma_3(u,v,i) = \gamma(v) = \beta(v) \times \left| \frac{\sum_{j \in \mathcal{I}_v} \big(r(v,j) - \bar{r}(v)\big)(\bar{r}(j) - \bar{R})}{\sqrt{\sum_{j \in \mathcal{I}_v} \big(r(v,j) - \bar{r}(v)\big)^2 \sum_{j \in \mathcal{I}_v} (\bar{r}(j) - \bar{R})^2}} \right|$$

Hwang and Chen (2007) present a global trust metric, which we call **global trust deviation**, defined as an average of local (user-to-user) trust deviations. This metric makes use of the predicted rating for a user–item pair by using only one user as neighbour:

$$\tilde{r}(u,i) \sim \tilde{r}(u,i;v) = \bar{r}(u) + \big(r(v,i) - \bar{r}(v)\big)$$

where user v is the considered neighbour. The predictor is then computed by averaging the prediction error of co-rated items between each user, and normalising the error according to the rating range R_r (e.g. in a typical 1 to 5 rating scale, $R_r = 4$):

$$\gamma_4(u,v,i) = \gamma(v) = \frac{1}{|N_k(v)|} \sum_{w \in N(v)} \left(\frac{1}{|\mathcal{I}_v \cap \mathcal{I}_w|} \sum_{j \in \mathcal{I}_v \cap \mathcal{I}_w} \left[1 - \frac{|\tilde{r}(v,j;w) - r(v,j)|}{R_r} \right] \right).$$

Finally, a performance predictor inspired by the clarity score defined for query performance (Cronen-Townsend et al., 2002) was proposed in (Bellogín and Castells, 2010), considering its adaptation to predict neighbour performance in collaborative filtering. In the same way query clarity captures the lack of ambiguity in a query, **user clarity** is expected to capture the lack of ambiguity in a user's preferences. Thus, the amount of uncertainty involved in a user's profile is assumed to be a good predictor of her performance; and the larger the following value, the lower the uncertainty and the higher the expected performance:

$$\gamma_5(u,v,i) = \gamma(v) = KLD(v \,\|\mathcal{U} \setminus \{u\}) = \sum_{w \in \mathcal{U} \setminus \{v\}} p(w|v)\, log_2 \frac{p(w|v)}{p(w)}$$

The probabilistic models defined in that work are based on smoothing estimations and conditional probabilities over users and items. Specifically, a uniform distribution is assumed for users and items, whereas the user-user probability is defined by an expansion through items as follows:

$$p(v|u) = \sum_{i \in \mathcal{I}_u} p(v|i)p(i|u).$$

Conditional probabilities are linearly smoothed with the user's probabilities and the maximum likelihood estimators, which finally depend on the rating given by the user towards an item; i.e., $p_{ml}(i|u) \propto r(u,i)$.

It is interesting to note that this predictor (and the probability model in which is grounded) does not correspond with any of the adaptations of the clarity score proposed in Chapter 6, since relations between users are not considered in any of the rating-based probability models presented.

In addition to the integration of the above methods in the role of neighbour effectiveness predictors in our framework, we propose two novel predictors based on well known quantities measured over the probability models of (Bellogín and Castells, 2010): the entropy and the mutual information. Entropy, as an information-theoretic magnitude, measures the uncertainty associated with a probability distribution (Cover and Thomas, 1991). Borrowing the definition of user entropy from Chapter 6, we hypothesise that the uncertainty in the system's knowledge about a user's preferences may be a relevant signal in the effectiveness of a user as a potential neighbour, which could be captured by the entropy of the item distribution as follows:

$$\gamma_7(u,v,i) = \gamma(v) = -H(\mathcal{I}_v) = \sum_{j \in \mathcal{I}_v} p(j|v)\, log_2\, p(j|v).$$

Note that uncertainty, measured in this way, can be due to the system's knowledge about the user's tastes, or may come from the user herself (e.g. some users may have strong preferences, while others may be more undecided), and both causes may similarly affect the neighbour effectiveness. In either case the predictor can be interpreted as the lack of ambiguity in a user profile.

The second information-theoretic magnitude we propose to use over the probability models presented above is the mutual information. To be precise, the mutual information is a quantity computed between two random variables that measure the mutual dependence of the variables, or, in other terms, the reduction in uncertainty about one variable provided some knowledge about the other (Cover and Thomas, 1991). Here, we propose to adapt this concept, and compute the **mutual information** between the neighbour and the rest of the community in order to assess the uncertainty involved in the neighbour's preferences. For this purpose, instead of computing the mutual information over all the events in the sample space for both variables (users), we fix one of them (for the current neighbour), and move along the other dimension:

$$\gamma_6(u,v,i) = \gamma(v) = MI(v; \mathcal{U} \setminus \{u\}) = \sum_{w \in \mathcal{U} \setminus \{u\}} p(w|v)\, log_2 \frac{p(w|v)}{p(v)p(w)}.$$

User-Item Predictors

User-item predictors consist of performance predictors that depend on a user-item pair. More specifically, they are defined upon the active neighbour and the target item. This type of predictor is more difficult to apply because of its higher vulnerability to data sparsity. In a bi-dimensional user-item input space less observations can be associated to each input data point, whereby the confidence on the predictor outcome is lower, as it can be biased to outliers or unusual users or items.

A local trust metric based on the target user and item is proposed in (O'Donovan and Smyth, 2005). This metric is called **item-level trust**, and aims to discriminate reliable neighbours depending on the current item, since the same user may be more trustworthy for predicting ratings for certain items than for others. The formulation of this predictor can be seen as a particularisation of γ_1, but constraining the recommendation set only to the pairs in which the current item is involved:

$$\gamma_8(u,v,i) = \gamma(v,i) = \frac{|\{(c_k,i_k) \in \text{CorrectSet}(v): i_k = i\}|}{|\{(c_k,i_k) \in \text{RecSet}(v): i_k = i\}|}.$$

User-User Predictors

The user-user predictors take as inputs two users: the active user and the current neighbour. User-user predictors based on local trust metrics have been studied further than user-item predictors in the literature, since the former are able to represent how much a user can be trusted by another, and let for different interpretations of the relation between users. These metrics have been often researched in the scope of social networks, and the users' explicit links in this context (Ziegler and Lausen, 2004; Massa and Avesani, 2007a), along with several trust metrics based on ratings, as we shall show below. In this way, although social-based metrics could be smoothly integrated in our framework, here we focus on a complementary view on trust where predictors are defined based on ratings. We leave other type of predictors as future work.

A first simple neighbour reliability criterion one may consider is the amount of common experience with the target user, that is, the amount of information upon which the two users can be compared. If we define "user experience" as the set of items the user has interacted with, we may define a predictor embodying this principle as:

$$\gamma_9(u,v,i) = \gamma(u,v) = |\mathcal{I}_u \cap \mathcal{I}_v|.$$

We shall refer to this predictor as **user overlap**. This predictor will serve as a basis for subsequent predictors, since most of them will depend on the items rated by both users. For instance, it has a clear use in assessing the reliability of the inter-user similarity assessments, which has been applied in the literature under a more practi-

cal, ad-hoc manner. Specifically, Herlocker et al. (2002) proposed the introduction of a weight on the similarity function, where the latter is devalued when it has been based on a small number of co-rated items. We may formulate **Herlocker's significance weighting** predictor as follows:

$$\gamma_{10}(u,v,i) = \gamma(u,v) = \frac{|\mathcal{I}_u \cap \mathcal{I}_v|}{n_H} \text{ if } |\mathcal{I}_u \cap \mathcal{I}_v| < n_H; 1 \text{ otherwise,}$$

where n_H is the minimum number of co-rated items that two users should have in common in order to avoid similarity penalisation. A value of $n_H = 50$ was proved empirically to work effectively.

A variation of the previous scheme was proposed in (McLaughlin and Herlocker, 2004), to which we shall refer as **McLaughlin's significance weighting**:

$$\gamma_{11}(u,v,i) = \gamma(u,v) = \frac{\max(|\mathcal{I}_u \cap \mathcal{I}_v|, n_{Mc})}{n_{Mc}}.$$

This predictor is aimed to be equivalent to the Herlocker's significance weighting (γ_{10}) formulation when $n_{Mc} = n_H$. However, we note that γ_{10} and γ_{11} represent different concepts, and are not fully equivalent. For instance, as noted in (Ma et al., 2007), γ_{11} may return values larger than 1 when $|\mathcal{I}_u \cap \mathcal{I}_v| > n_{Mc}$, while γ_{10}, by definition, always returns a value in the $(0,1]$ interval.

Alternatively, the following variant can be drawn from (Ma et al., 2007), which is just a more compact reformulation of γ_{10}:

$$\gamma_{12}(u,v,i) = \gamma(u,v) = \frac{\min(|\mathcal{I}_u \cap \mathcal{I}_v|, n_M)}{n_M}.$$

A more elaborated predictor was proposed in (Weng et al., 2006). The rationale behind such predictor is to consider two situations depending whether or not user u takes into account the recommendation made by neighbour v. In this sense trustworthiness is defined as the reduction in the proportion of incorrect predictions of going from the latter situation to the former. The definition of this predictor, denoted as **user's trustworthiness**, is the following:

$$\gamma_{13}(u,v,i) = \gamma(u,v) = \frac{1}{|R|^2 - \sum_x n(u,v;x,\cdot)^2}\left[|R| \sum_x \sum_y \frac{n(u,v;x,y)^2}{n(u,v;\cdot,y)} - \sum_x n(u,v;x,\cdot)^2\right]$$

In this formulation $|R|$ represents the number of allowed rating values in the system (e.g. in a 1 to 5 rating scale, $|R| = 5$), the function $n(u,v;x,y)$ represents the number of co-rated items on which v's ratings have the value y while u's ratings are x, that is, $n(u,v;x,y) = |\{(u,\cdot,x)\} \cap \{(v,\cdot,y)\}|$ when each rating tuple is represented as (a,b,c), given a user a, an item b, and a rating value c. In the same way, $n(u,v;x,\cdot) = \sum_y n(u,v;x,y)$ represents all the co-rated items between u and v

rated with any rating value by user v, and, analogously, $n(u,v;\cdot,y)=\sum_x n(u,v;x,y)$. In this case, the assumed hypothesis is that trust is one's expectation of other's competence in reducing its uncertainty in predicting new ratings.

Finally, a user-user predictor can be defined based on the global trust deviation predictor defined above (γ_4). In fact, Hwang and Chen (2007) define **trust deviation** by ignoring the average along users as follows:

$$\gamma_{14}(u,v,i)=\gamma(u,v)=\frac{1}{|\mathcal{I}_u\cap\mathcal{I}_v|}\sum_{j\in\mathcal{I}_u\cap\mathcal{I}_v}\left[1-\frac{|\tilde{r}(u,j;v)-r(u,j)|}{R_r}\right]$$

This predictor identifies effective neighbours mainly based on how many trustworthy (understood as "accurate") recommendations a user has received from another.

8.4 Experimental results

In this section we report experiments in which the proposed neighbour effectiveness prediction framework is tested. First, we check the existing correlations between the user-based predictors defined in Section 8.3.2 and the neighbour performance metrics proposed in Section 8.3.1, as a direct test of their predictive power. For the user-item predictors we cannot analyse their correlation because we have no neighbour performance metric depending on both the target user and an item available.

Moreover, we test the usefulness of the predictors to enhance the final performance of memory-based algorithms, by using the predictors' values in the selection and weighting of neighbours, that is, by taking the predictors as the scoring function in Equation (8.2).

Our experiments were conducted on two versions of the MovieLens dataset, namely the 100K and 1M versions, described in Section 3.4.1 and Appendix A.1. For the user-based collaborative filtering method, we used Pearson's correlation as the similarity measure between users, and a varying neighbourhood size (k), which is a parameter with respect to which the results were examined.

8.4.1 Correlation analysis

We analyse the correlation between neighbour quality metrics and neighbour performance predictors in terms of the Pearson and Spearman's correlation metrics. Correlation provides a measure of the predictive power of the neighbour effectiveness prediction approaches: the higher the (absolute) correlation value, the better the predictor estimates the positive neighbour effect on the recommendation accuracy. The sign of the correlation coefficient represents whether the two involved variables – neighbour quality metric and neighbour performance predictor – are directly or inversely correlated.

	Absolute error deviation μ_1 (-)	Neighbour goodness μ_3 (+)	Sign of error μ_2 (+)
Clarity	-0.21	+0.17	+0.14
Entropy	-0.18	+0.18	+0.12
Expertise	-0.62	+0.03	+0.25
Global Trust Deviation	-0.35	-0.01	+0.08
Mutual Information	-0.20	+0.17	+0.12
Profile Level Trust	+0.62	-0.04*	-0.24
Trustworthiness	-0.21	+0.03	+0.20

Table 8.1. Pearson's correlation between the proposed neighbour quality metrics and neighbour performance predictors in the MovieLens 100K dataset. Next to the metric name, an indication about the sign of the metric – direct(+) or inverse(-) – is included. Not significant values for a p-value of 0.05 are denoted with an asterisk (*).

	Absolute error deviation μ_1 (-)	Neighbour goodness μ_3 (+)	Sign of error μ_2 (+)
Clarity	-0.30	+0.16	+0.21
Entropy	-0.22	+0.17	+0.15
Expertise	-0.65	+0.02	+0.30
Global trust deviation	-0.38	-0.03	+0.11
Mutual Information	-0.25	+0.16	+0.17
Profile Level Trust	+0.65	-0.02	-0.30
Trustworthiness	-0.24	+0.03	+0.25

Table 8.2. Spearman's correlation between quality metrics and performance predictors in the MovieLens 100K dataset.

Table 8.1 and Table 8.2 show the correlation values obtained on the MovieLens 100K dataset for the user-based predictors. We associate a sign to each quality metric indicating whether the metric is direct (denoted as '+') or inverse (denoted with '-'), according to the expected sign of the correlation with the predictor, i.e., a metric is direct if the higher its value, the better the true neighbour performance. We can observe that the Spearman's correlation values are consistent, but slightly higher than Pearson's, thus evidencing a non-linear relationship between the quality metrics and the performance predictors.

The absolute error deviation (μ_1) metric presents higher values when the neighbour's prediction is less accurate, being thus an inverse neighbour metric. The other two metrics, sign of error (μ_2) and neighbour goodness (μ_3), are, by definition, direct neighbour metrics, since the former indicates how many times a recommendation from the neighbour has been made in the right direction, whereas the latter represents the change in error between excluding a particular user in the neighbourhood or including her, and thus, the larger this error, the "better" neighbour this user.

	Absolute error deviation μ_1 (-)	Neighbour goodness μ_3 (+)	Sign of error μ_2 (+)
Clarity	-0.14	+0.40	+0.02
Entropy	-0.07	+0.39	-0.08
Expertise	-0.95	-0.06	+0.70
Global Trust Deviation	-0.55	-0.24	+0.36
Mutual Information	-0.17	+0.30	+0.13
Profile Level Trust	+0.83	+0.04	-0.55
Trustworthiness	-0.27	+0.03	+0.36

Table 8.3. Pearson's correlation between quality metrics and performance predictors in the MovieLens 1M dataset. All the values are significant for a p-value of 0.05.

	Absolute error deviation μ_1 (-)	Neighbour goodness μ_3 (+)	Sign of error μ_2 (+)
Clarity	-0.16	+0.35	+0.04
Entropy	-0.03	+0.37	-0.10
Expertise	-0.94	-0.09	+0.69
Global trust deviation	-0.54	-0.25	+0.39
Mutual information	-0.16	+0.31	+0.04
Profile level trust	+0.94	+0.09	-0.69
Trustworthiness	-0.25	+0.02	+0.37

Table 8.4. Spearman's correlation between quality metrics and predictors in the MovieLens 1M dataset.

We can observe in Table 8.1 that, except for some of the predictors that obtain very low absolute values (< 0.10), the four quality metrics are consistent with each other. This consistency is evidenced by the way the predictors correlate with the different metrics: some of the predictors obtain the correct correlations in every situation, that is, positive correlation with direct metrics and negative correlation with the inverse metric (like the clarity predictor), while other predictors obtain opposite values for all the metrics, that is, positive correlations with the inverse metric and negative correlations with direct metrics (such as the profile level trust predictor).

Also in Table 8.1 and Table 8.2 we see that each metric captures a different notion of neighbour quality because they show different correlation values with respect to the predictors. In this way, although consistent correlation results are obtained for direct and inverse metrics, each of them is actually detecting a different nuance of how a neighbour should behave in order to perform well.

Table 8.3 and Table 8.4 show the correlation values obtained on the Movie-Lens 1M dataset. We can observe that the trend in correlation is very similar to the behavior observed on the 100K dataset, and thus, similar conclusions can be drawn from it. There are, however, some changes in the absolute values of the correlation scores for some combinations of performance predictor and quality metric. For instance,

the clarity predictor and the neighbour goodness metric obtain larger values in this dataset, while the correlation between entropy and absolute error deviation is smaller.

It is important to note that the number of points used to compute the correlation values is different in the two datasets; there are less than 1,000 points in MovieLens 100K (with 943 users), and more than 6,000 points in MovieLens 1M dataset. This difference affects the significance of the correlation results, as already described in Section 5.4.2, where we observed how the confidence test for a Pearson's (and Spearman's) correlation depends on the size of the sample, and thus, the significance of a correlation value may change for different sample sizes.

In our experiments, for MovieLens 100K, the correlations are significant for a p-value of 0.05 when $r > 0.05$, and in the 1M dataset when $r > 0.02$. Hence, in Table 8.1, there is only one non-significant correlation value (denoted with an asterisk), whereas in Table 8.3, all the results are statistically significant.

Analysing in more detail the reported results for both datasets, we observe that the profile level trust predictor consistently obtains direct correlation values with inverse metrics, whereas inverse correlation values are obtained with direct metrics. This predictor seems to give higher scores to neighbours with larger deviations in their accuracy error, which would result on bad performance prediction because these values are not in the same direction than the performance metrics. The expertise and global trust deviation predictor obtain strong inverse correlations with the absolute error deviation metric, although their correlations with respect to the neighbour goodness metric are negligible, especially for the first predictor, in both datasets. At the other end of the spectrum, the clarity, entropy, and mutual information predictors obtain strong correlation values with the neighbour goodness, and moderate correlations with the rest of metrics, which make these predictors good candidates for successful neighbour performance predictors. Finally, the trustworthiness predictor obtains a significant amount of correlation with respect to the absolute error deviation and sign of error metrics, although its correlation with respect to the neighbour goodness is very low. This predictor thus seems to be useful on estimating how accurate the neighbour may be in terms of the error in a user basis, but probably not as a global metric.

Table 8.5 shows the correlations obtained for user-user neighbour predictors and the proposed user-neighbour clarity metric. Due to the high dimensionality of the vectors involved in this computation, we have considered only those users that have at least one item in common. Despite this fact, correlations are almost negligible, except for the McLaughlin's significance weighting predictor and the Spearman's coefficient, which evidences a non-linear relation between this predictor and the metric. In the next section we shall show that this function is one of the best performing predictors among the evaluated neighbour scoring functions. This result confirms the usefulness of the proposed neighbour performance metric since it is able to discrimi-

	Movielens 100K		Movielens 1M	
	Pearson	Spearman	Pearson	Spearman
Herlocker	0.02	0.03	0.01	0.02
McLaughlin	0.01	0.12	0.01	0.11
Trust Deviation	0.01	0.01	0.01	0.01
User Overlap	0.02	0.03	0.02	0.02
User's Trustworthiness	-0.02	-0.02	-0.01	-0.01

Table 8.5. Correlation between the user-neighbour goodness and user-user predictors in the two datasets evaluated.

nate which neighbour performance predictors are able to capture interesting properties between the user and her neighbours.

In summary, we have observed that most of the performance predictors agree with respect to the different performance metrics, and in general, the correlations computed between neighbour quality metrics and neighbour performance predictors are statistically significant.

8.4.2 Performance analysis

The results reported in the previous section show that some of the studied predictors have the ability to capture neighbour performance, and because of that we hypothesise that they could be used to improve the accuracy of a recommendation model. This hypothesis, nonetheless, has to be checked since the metric against which we measure the neighbour goodness is not the same as the final recommendation performance metric we aim to optimise. With the experiments we report next we aim to confirm the usefulness of the proposed predictors, the validity of the proposed metrics as useful references to assess the power of the predictive methods, and the usefulness of the overall framework as a unified approach to enhance neighbourhood-based collaborative filtering.

In order to achieve this we test the integration of the neighbour predictors into a neighbour selection and weighting scheme for user-based collaborative filtering, as described in Section 8.2.1. Besides testing the effectiveness of the predictors, this experiment provides for observing to what extent the correlations obtained in the previous section correspond with improvements in the final performance of those predictors.

We provide recommendation accuracy and precision results on the MovieLens 1M dataset. Those obtained on the MovieLens 100K dataset are not reported here since they had similar trends. Figure 8.1 and Figure 8.2 show the Root Mean Square Error (RMSE) of the Resnick's collaborative filtering adaptation proposed in Equation (8.2) when used for different neighbour selection and weighting approaches. The curves at the top of the figures represent the values obtained when neighbour per-

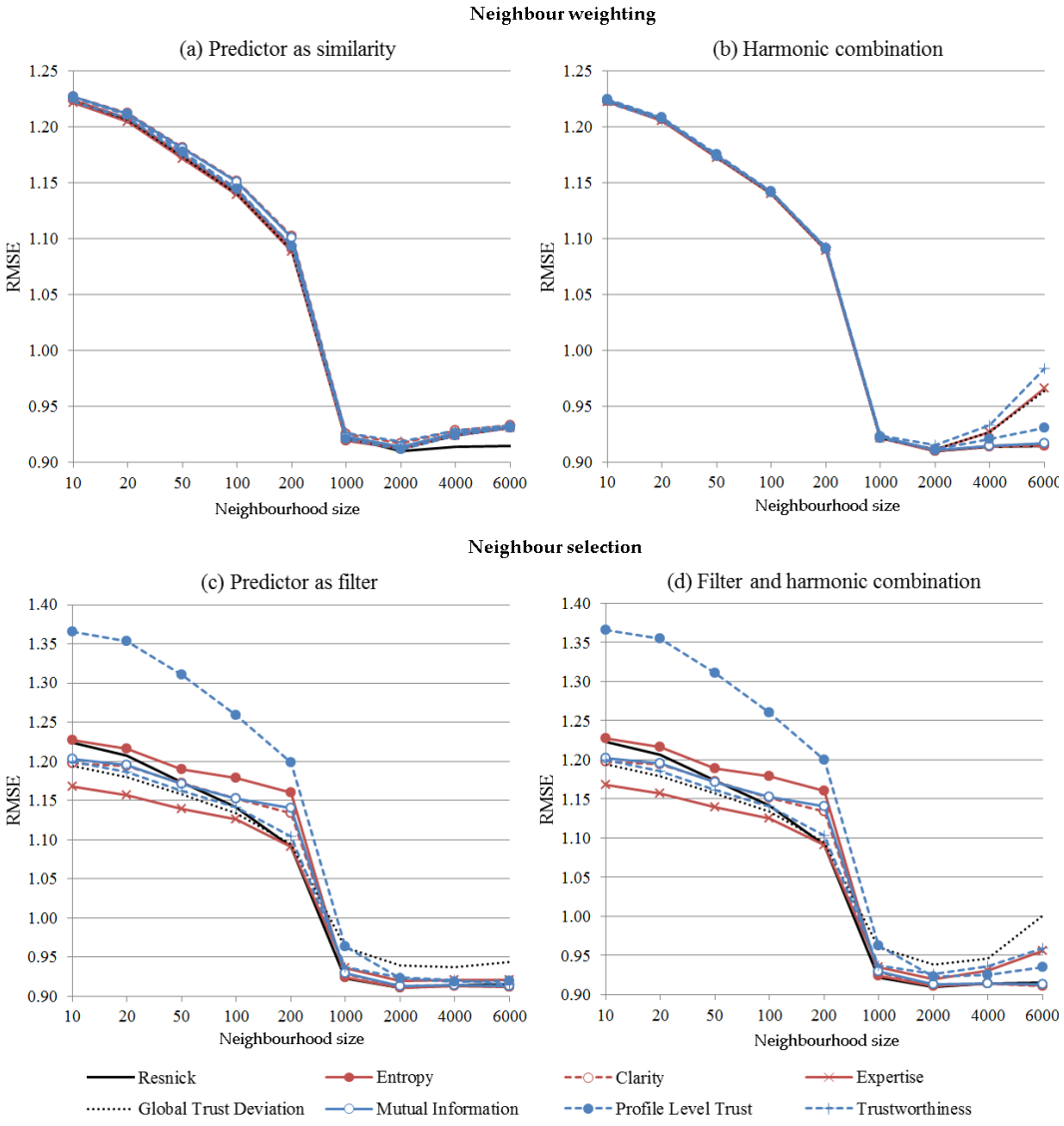

Figure 8.1. Performance comparison for user-based predictors and different neighbourhood sizes.

formance predictors are used for neighbour weighting, that is, when the standard neighbour selection strategy is used ($f^{neigh} = f_0^{neigh}$ in Equation (8.2)). Note that since the lines represent errors, the lower these values, the better the performance. Besides, Figure 8.3 presents the results found with the precision at 10 (P@10) ranking metric of a subset of the proposed methods, where in this case the higher the values, the better the performance.

A different aggregation function is used in each approach, depending on whether the harmonic mean between the predictor score and the similarity value (function $f^{agg} = f_2^{agg}$, on the right), or the projection function ($f^{agg} = f_4^{agg}$, on the left) are used, in the latter case in order to ignore the similarity. The curves at the bottom

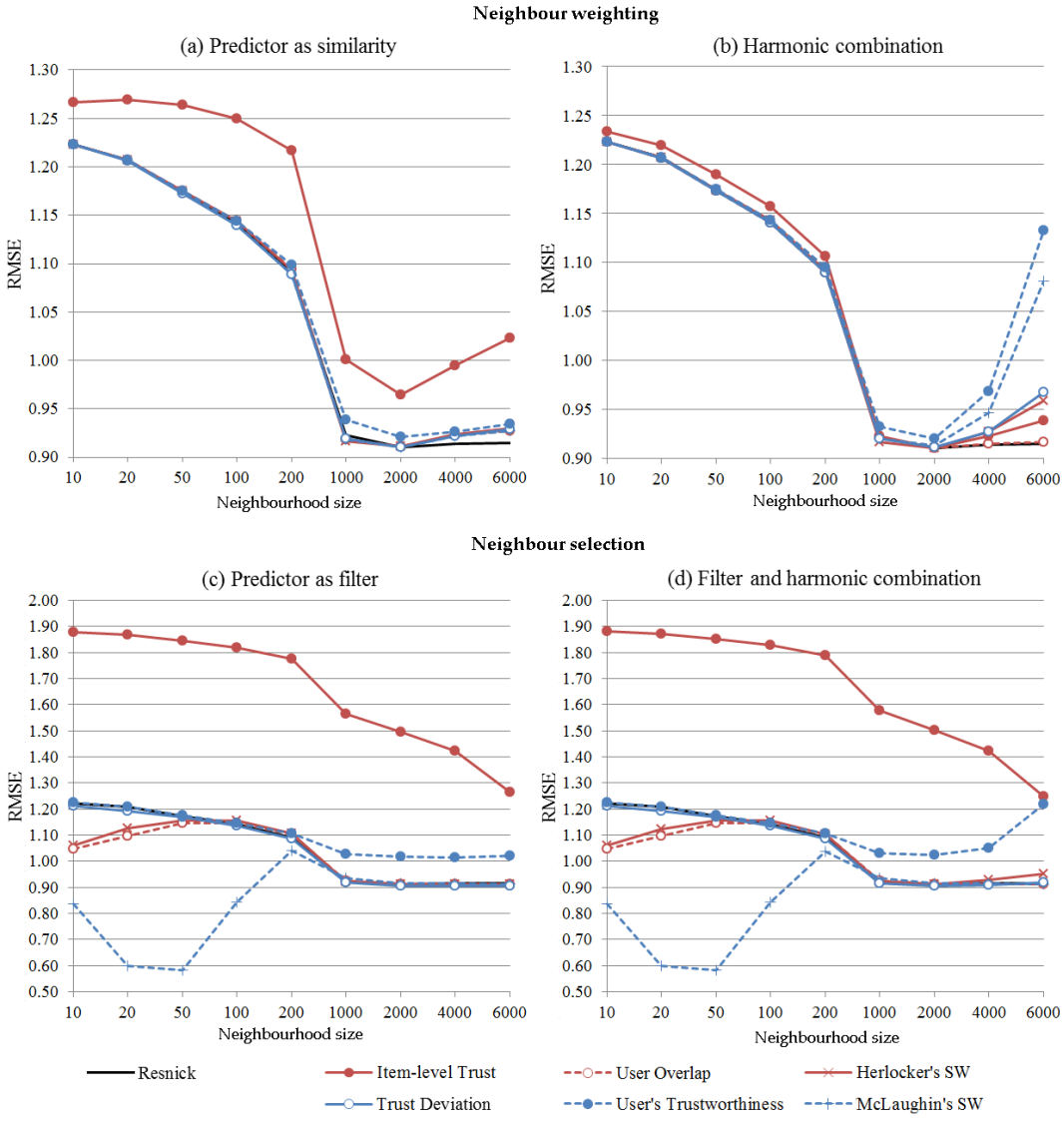

Figure 8.2. Performance comparison using user-item and user-user predictors for different neighbourhood sizes.

of the figures show the neighbour selection approach ($f^{neigh} = f_1^{neigh}$ in Equation (8.2)) along with the same neighbour weighting functions described above (i.e., f_2^{agg} on the right and f_4^{agg} on the left). The rest of the aggregation functions, such as average (f_1^{agg}) and product (f_3^{agg}), were also evaluated for neighbour selection and weighting, but provided results equivalent to those of the harmonic mean. For this reason, they have been omitted in the figures to avoid cluttering them. We believe this equivalence may be due to the normalisation factor included in the collaborative filtering formulation, since it would cancel out the weights obtained by the harmonic, average, and product functions in the same way.

	RMSE
Resnick	1.174
Clarity	1.181
Entropy	1.175
Expertise	1.171
Global Trust Deviation	1.173
Mutual Information	1.180
Profile Level Trust	1.177
Trustworthiness	1.175

	RMSE
Resnick	1.174
Herlocker	1.175
Item-level Trust	1.264
McLaughlin	1.174
Trust Deviation	1.173
User Overlap	1.175
User's Trustworthiness	1.175

Table 8.6. Detail of the accuracy of baseline vs. recommendation using neighbour weighting; here, performance predictors are used as similarity scores (50 neighbours).

	RMSE
Resnick	1.174
Clarity	1.172
Entropy	1.189
Expertise	1.139
Global Trust Deviation	1.158
Mutual Information	1.171
Profile Level Trust	1.310
Trustworthiness	1.162

	RMSE
Resnick	1.174
Herlocker	1.156
Item-level Trust	1.843
McLaughlin	0.581
Trust Deviation	1.168
User Overlap	1.146
User's Trustworthiness	1.174

Table 8.7. Detail of the accuracy of baseline vs recommendation using neighbour selection; here, performance predictors are used for filtering (50 neighbours).

Figure 8.1 shows the accuracy results when only user-based neighbour predictors are evaluated. We observe that, independently from the neighbourhood size, using performance predictors as similarity scores does not lead to large differences with respect to the baseline. These results are compatible with those presented in (Weng et al., 2006), where the improvement in RMSE is not very high (ΔMAE < 0.05 in that work). For the sake of clarity, in Table 8.6 and Table 8.7 we show the error values for a horizontal cut of the left curves; specifically, when the neighbourhood size is 50. We can observe that some predictors do improve Resnick's accuracy. Regarding the use of the harmonic mean as aggregation function (curves on the right), similar results are obtained except for very large neighbourhood sizes, for which some of the performance predictors produce worse results than the baseline, probably due to the amount of noise created by considering too many neighbours.

The curves at the bottom of the figures represent the accuracy results for neighbour selection strategies. In this case some of the predictors lead to worse performance than the baseline, particularly the profile level trust (γ_1). This situation is consistent with the correlations observed in the previous section, since this predictor obtained inverse correlations with the different metrics, i.e., direct correlation values

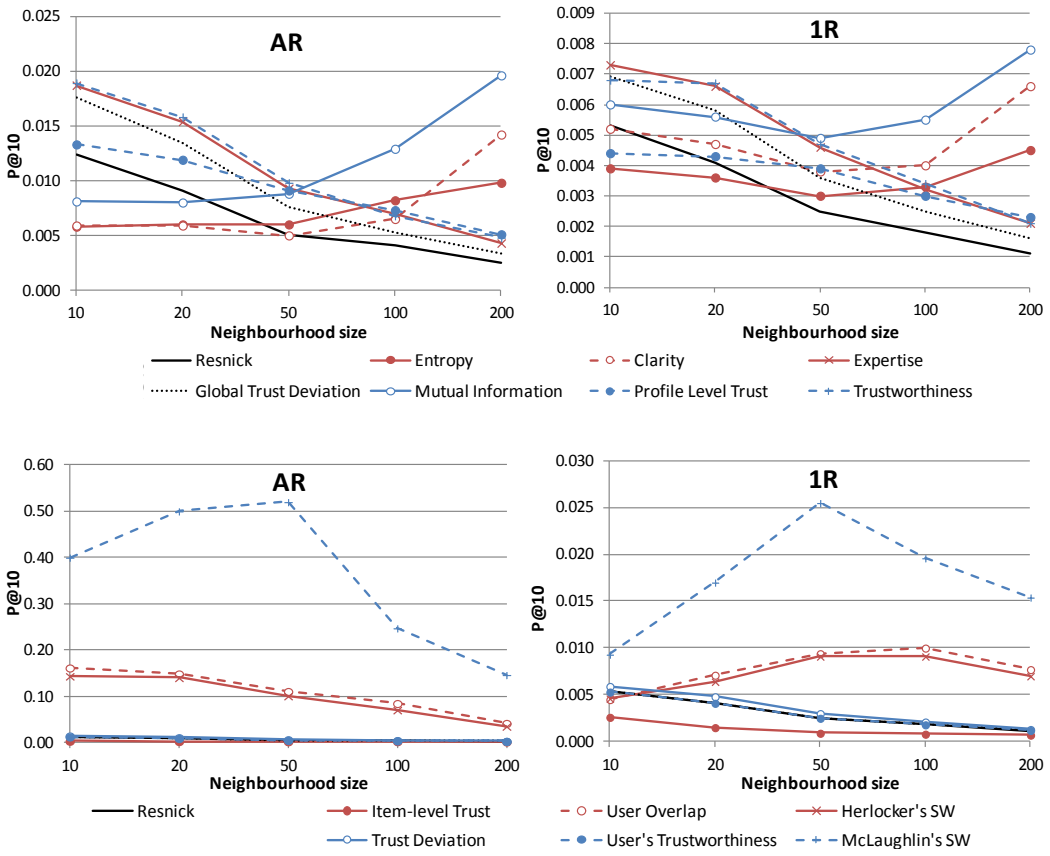

Figure 8.3. Performance comparison using ranking-based metrics for both user and user-user neighbour predictors using the AR and 1R evaluation methodologies.

with inverse metrics, and inverse values with direct metrics. Moreover, as predicted by the correlation analysis, trustworthiness (γ_3), mutual information (γ_6), and clarity (γ_5) result in some of the best performing recommenders (with strong correlations), as shown in the figures and in Table 8.7, along with expertise (γ_2) and global trust deviation (γ_4), which obtained more moderated correlation values.

In Figure 8.2 we can see how user-item and user-user neighbour predictors affect the performance of collaborative filtering recommenders. The curves in the top show that most of the predictors obtain a similar performance to that of the baseline, except for the item-level trust (γ_8), the performance of which is much worse than Resnick's. Table 8.6 shows the specific error values for these recommenders. It is interesting to note that the performance of this predictor is drastically improved when using the harmonic mean as the aggregation function (shown on the right side of the figure). Similarly to user-based neighbour predictors (Figure 8.1), some of the user-item and user-user predictors decrease their accuracy with large neighbourhoods; in this case, user's trustworthiness (γ_{13}) and McLaughlin's significance weighting (γ_{12}) are the more representative examples.

A different conclusion results when neighbour selection is analysed (curves at the bottom). Two of the predictors are characterised by a much better (McLaughlin's significance weighting, γ_{12}) or worse (item-level trust, γ_8) final performance, independently from the weighting aggregation function. Table 8.7 shows the specific error values obtained for each of these predictors. It is interesting how the McLaughlin's predictor, despite its inability to boost good neighbours (see top figures), seems to be very useful for neighbour selection. This effect, nonetheless, is attenuated when the neighbourhood increases, since in that situation, selection methods have to deal with too many users in each neighbourhood. We believe the reason why this predictor is very good for neighbour selection is because it gives higher scores to those neighbours that have more items in common with the target user, and thus the confidence in the computation of the similarity values between the neighbour and the target user is higher. It is worth noting that, to the best of our knowledge, this function has never been used for neighbour selection, since its original motivation was to penalise the similarity value whenever it has been based on a small number of co-rated items. However, by plugging this function into our framework, and measuring its predictive power for user-neighbour performance, a novel application naturally emerges and provides very good results.

Finally, in Figure 8.3 we can observe that a similar trend is found with P@10 for both user-based predictors (top curves), and user-item and user-user predictors (bottom curves). In the figure we only present the results of the neighbour selection and weighting approaches for less than 200 neighbours, since the results of the rest of the approaches and neighbourhoods are very similar. It is worth noting that the two methodologies evaluated – AR and 1R – agree on the order of the best and worst performing dynamic approaches, although as already observed in the previous chapter, the absolute performance values obtained with each methodology may be very different – e.g. the maximum P@10 value with 1R is 0.1, which is reached by several recommendation methods with the AR methodology. More interestingly, these results show consistency between the performance of some dynamic approaches using error- and ranking-based metrics, since the best and worst predictors according to RMSE and P@10 are the same; McLaughlin's significance weighting and item-level trust, respectively. Moreover, the entropy and clarity user-based predictors show worse performance in small neighbourhoods, but outperform the baseline significantly in larger neighbourhoods, something different to what we observed in the previous experiment with error-based metrics.

In summary, we have been able to validate both the proposed user-user neighbour performance metrics, and the different evaluated user-user neighbour performance predictors. We have obtained positive results when this type of predictors has been introduced and compared against the baseline in the different aggregation strategies and configurations, and these results are consistent with the correlations

obtained between the predictors and the performance metrics. In particular, McLaughlin's significance weighting obtains an improvement up to 55% in both accuracy (i.e., error decrease) and precision (i.e., precision improvement) when this predictor is used to select the neighbours which will further contribute to the rating prediction. Besides, the (Spearman's) correlation for this predictor is positive and strong, in contrast to the values obtained for the rest of user-user predictors, which did not improve the accuracy of the baseline. In this context, a possible drawback of the conducted analysis is that we have not been able to define neighbour performance metrics based on user-item pairs, and thus the user-item neighbour performance predictors are out of the scope of the developed correlation analysis. Nevertheless, the obtained results showed that the only user-item neighbour performance predictor defined here – the item-level trust – is not able to outperform the baseline recommender. We believe this fact, which is in contradiction with what was reported in (O'Donovan and Smyth, 2005), may be caused by the different variables taking place in our evaluation, such as the dataset (MovieLens 1M instead of MovieLens 100K), the neighbourhood size (not specified in the original paper), and the several aggregation functions and combinations used across our experiments.

8.4.3 Discussion

The reported experiment results provide empiric evidence of the usefulness of the proposed framework, and the specific proposed predictors, as an effective approach to enhance the accuracy of memory-based collaborative filtering. As described in the preceding sections, the methodology comprises two steps, one in which the predictive power of neighbour predictors is assessed, and one in which the predictors are introduced in the collaborative filtering scheme to enhance the effectiveness of the latter. Our experiments confirm a strong correlation for some of the predictors – both user predictors and user-user predictors –, and this has been found to correspond with final accuracy enhancements in the recommendation strategy: the predictors that obtain strong direct correlations with the performance metrics are the best performing dynamic strategies; the profile level trust predictor, which obtains inverse correlation values with respect to the neighbour performance metrics, is the worst performing dynamic strategy.

In light of these results, it could be further investigated whether the actual correlation values between neighbour performance predictors and neighbour performance metrics could be used to infer how each predictor should be incorporated into a memory-based collaborative filtering method as a neighbour scoring function, since there is no obvious link between the ranking of the best performing scoring functions and the strength of their corresponding correlations. As a starting point, only the sign of the correlation could be considered, using either the raw neighbour predictor score (for positive correlations) or its inverse (for negative values). Then, this

rationale could be further elaborated and evaluated in order to check whether the performance improvements are consistent.

Research on finding functions with strong correlation power with respect to neighbour performance metrics could be an interesting area by itself, since it could have different final applications. We have experimented here with variations in neighbour selection and weighting for user-based collaborative filtering, but those predictors (functions) could also be used, for instance, for active learning (Elahi, 2011), or for providing more meaningful explanations (Marx et al., 2010), depending or based on the predicted performance of a particular user's neighbours.

8.5 Conclusions

We have shown in this chapter that performance prediction does not only serve to aggregate entire recommender systems, but also to aggregate subcomponents of recommender algorithms – in this case, neighbour related terms in collaborative filtering. We propose a theoretical framework for neighbour selection and weighting in user-based recommender systems, which is based on a performance prediction approach drawn from the query performance methodology of the Information Retrieval field. By viewing the neighbourhood-based collaborative filtering rating prediction task as a case of dynamic output aggregation, our approach places user-based collaborative filtering in a more general frame, linking to the principles underlying the formation of ensemble recommenders, and rank aggregation in Information Retrieval. By doing so, it is possible to draw concepts and techniques from these areas, and vice versa. Our study thus provides a comparison of different state-of-the-art rating-based trust metrics and other neighbour scoring techniques, interpreted as neighbour performance predictors, and evaluated under this new angle. The framework lets an objective analysis of the predictive power of several neighbour scoring functions, integrating different notions of neighbour performance into a unified view. Thus, the proposed methodology discriminates which neighbour scoring functions are more effective in predicting the goodness of a neighbour, and thus identifies which weighting functions are more effective in a user-based collaborative filtering algorithm.

Drawing from different state-of-the-art neighbour scoring functions – cast as user, user-user, and user-item neighbour performance predictors –, we have reported several experiments in order to, first, check the predictive power of these functions, and second, validate them by comparing the final performance of neighbour-scoring powered memory-based strategies with that of the standard collaborative filtering algorithm. We also evaluate different ways to introduce these functions in the rating prediction formulation, namely for neighbour weighting, neighbour selection, and combinations thereof. In this context, methods where neighbour scoring functions

were integrated outperform the baseline for different values of neighbourhood size and predictor type.

We have also proposed several neighbour performance metrics that capture different notions of neighbour quality. The evaluated performance predictors show consistent correlations with respect to these metrics, and some of them present particularly strong correlations. Interestingly, a correspondence is confirmed between the correlation analysis and the final performance results, in the sense that the correlation values obtained between neighbour performance predictors and neighbour performance metrics anticipate which predictors will perform better when introduced in a memory-based collaborative filtering algorithm.

This research opens up the possibility to several research lines for the integration of other types of predictors and trust metrics into our framework. For instance, performance predictors defined upon social data, such as those defined in Chapter 6 based on user's trust network, could be smoothly integrated into our framework and analysed in the future. Furthermore, alternative neighbour performance metrics may be defined to check the predictive power of user-user and user-item predictors. These metrics may help better understand which characteristics of the neighbour performance such predictors are capturing, although based on a smaller amount of information since in rating-based systems users only rate items once. In particular, our framework would allow for different interpretations of the user's performance, by modelling different neighbour performance metrics, which may be oriented to accuracy (using error metrics as in this chapter), ranking precision, or even alternative metrics such as diversity, coverage and serendipity (Shani and Gunawardana, 2011). Additionally, other predictors based on item information could be defined similar to those proposed in (Weng et al., 2006; Ma et al., 2007), and easily incorporated into our framework using item-based algorithms instead of user-based.